PLOS COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

Optimal noise level for coding with tightly balanced networks of spiking neurons in the presence of transmission delays

Jonathan Timcheck[1]*, **Jonathan Kadmon**[2¤], **Kwabena Boahen**[3], **Surya Ganguli**[2]

1 Department of Physics, Stanford University, Stanford, California, United States of America, **2** Department of Applied Physics, Stanford University, Stanford, California, United States of America, **3** Department of Bioengineering, Stanford University, Stanford, California, United States of America

¤Current address: Edmond and Lily Safra Center for Brain Sciences, The Hebrew University, Jerusalem, Israel
* timcheck@alumni.stanford.edu

Citation: Timcheck J, Kadmon J, Boahen K, Ganguli S (2022) Optimal noise level for coding with tightly balanced networks of spiking neurons in the presence of transmission delays. PLoS Comput Biol 18(10): e1010593. https://doi.org/10.1371/journal.pcbi.1010593

Editor: Gunnar Blohm, Queen's University, CANADA

Received: March 13, 2022

Accepted: September 21, 2022

Published: October 17, 2022

Data Availability Statement: All relevant code and data are within the manuscript and its Supporting information files.

Funding: S.G. thanks Nippon Telegraph and Telephone (NTT) Research (https://ntt-research.com/), the Simons Foundation (https://www.simonsfoundation.org/), the James S. McDonnell Foundation (https://www.jsmf.org/), and an NSF Career Award for funding (https://www.nsf.gov/). J. K. thanks the Swartz Foundation (http://www.theswartzfoundation.org/) for funding. J.T. thanks

Abstract

Neural circuits consist of many noisy, slow components, with individual neurons subject to ion channel noise, axonal propagation delays, and unreliable and slow synaptic transmission. This raises a fundamental question: how can reliable computation emerge from such unreliable components? A classic strategy is to simply average over a population of N weakly-coupled neurons to achieve errors that scale as $1/\sqrt{N}$. But more interestingly, recent work has introduced networks of leaky integrate-and-fire (LIF) neurons that achieve coding errors that scale *superclassically* as $1/N$ by combining the principles of predictive coding and fast and tight inhibitory-excitatory balance. However, spike transmission delays preclude such fast inhibition, and computational studies have observed that such delays can cause pathological synchronization that in turn destroys superclassical coding performance. Intriguingly, it has also been observed in simulations that noise can actually *improve* coding performance, and that there exists some optimal level of noise that minimizes coding error. However, we lack a quantitative theory that describes this fascinating interplay between delays, noise and neural coding performance in spiking networks. In this work, we elucidate the mechanisms underpinning this beneficial role of noise by deriving *analytical* expressions for coding error as a function of spike propagation delay and noise levels in predictive coding tight-balance networks of LIF neurons. Furthermore, we compute the minimal coding error and the associated optimal noise level, finding that they grow as power-laws with the delay. Our analysis reveals quantitatively how optimal levels of noise can rescue neural coding performance in spiking neural networks with delays by preventing the build up of pathological synchrony without overwhelming the overall spiking dynamics. This analysis can serve as a foundation for the further study of precise computation in the presence of noise and delays in efficient spiking neural circuits.

the National Science Foundation Graduate Research Fellowships Program (https://www.nsf.gov/) for funding. K.B. thanks the Office of Naval Research (https://www.onr.navy.mil/) and the Stanford Medical Center Development (https://medicalgiving.stanford.edu/) Discovery Innovation Fund for funding. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Author summary

Remarkably, the brain can perform precise computations, despite being composed of noisy neurons with slow, unreliable synaptic connectivity. To understand how this is possible, we can imagine the classic strategy where neurons are grouped into weakly-coupled subpopulations, creating redundancy to achieve high precision. But interestingly, recent work proposed a tight-balance neural network that instead uses fast, strong connectivity between neurons to achieve much higher precision with the same number of neurons. This efficiency is attractive, but notably, signals take time to propagate in the brain. Such propagation delays alone can lead to pathological synchronization. Intriguingly, while noise commonly degrades the performance of a computational system, it has been observed in simulations that noise can help mitigate synchronization and in fact rescue performance in tight-balance networks. In this work, we develop a theory that quantifies the simultaneous effects of delays and noise in tight-balance networks, and allows us to compute the optimal noise level as a function of delay, yielding conceptual insights into how noise can counteract delay induced synchronization to preserve precise computation in efficient neural networks.

Introduction

The brain is capable of precise, reliable computation—for example, a professional violinist generates fine motor commands to reproduce a given pitch, or an impressionist produces speech patterns to mimic a celebrity's voice. Yet, the underlying computational substrates in the brain—neurons, synapses, and axonal transmission—are noisy, unreliable, and slow [1, 2]. This paradox begs a fundamental question in neuroscience: how do neural networks facilitate precise computation with imprecise computational primitives [3]? And moreover, in light of evolutionary forces favoring energy-efficiency [4, 5], how is this precise computation facilitated *efficiently*?

The simplest setting to study the precision of computation is that of coding [6]—with what fidelity can, say, a dynamical signal $x(t)$ be encoded in a network and read back out as an estimate $\hat{x}(t)$? The classic strategy is to read out $\hat{x}(t)$ as an average over N redundant neurons; this results in a readout error that scales as $1/\sqrt{N}$ as long as single neuron noise is not strongly coupled across the population [1]. However, a recent predictive coding framework [7] introduced a formulation for a tightly-balanced network of spiking leaky-integrate-and-fire (LIF) neurons with readout error that scales *superclassically* as $1/N$. In predictive coding, only the unpredicted difference $\hat{x}(t) - x(t)$ is encoded and passed to downstream processing, saving a great deal of information compared to directly encoding $x(t)$; signatures of predictive coding have been observed in sensory areas of the brain [8]. The predictive coding framework [7] combines this principle with that of a strong, fast inhibitory feedback, known as tight balance, so that each neural spike corrects the error $\hat{x}(t) - x(t)$ when it reaches a threshold and rapidly inhibits the other neurons to prevent overcorrections—this results in a highly efficient code, in which no spike is wasted. Moreover, despite each spike's dedicated purpose, the framework is robust to the death of individual neurons and reproduces the highly irregular spiking activity observed in cortex [9].

Critically, however, axonal and synaptic transmission introduce a delay that renders the rapid inhibition in the framework [7] problematic, reducing the fidelity of the code [10–12]. Namely, if an inhibitory signal arrives late, other neurons may spuriously spike, producing

overcorrections in the error and wasted spikes. However, intriguingly, adding noise to the neural membrane potentials introduces a beneficial variation, spreading out the times at which neurons will spike next so that the delayed inhibition has sufficient time to propagate before spurious spikes occur, and rescues the coding fidelity. Notably, too little noise does not provide a sufficient spread, and too much noise destroys overall fidelity. Thus an optimal noise level exists.

Importantly, several simulation studies have observed the beneficial role of noise in the predictive coding framework [7], regardless of the specific neural model or noise modality. For example, [12] studies a soft-threshold neural model with transmission delays, [11] studies an LIF model with membrane noise, transmission delays, and synaptic delays, and [10] studies conductance-based Hodgkin-Huxley neural dynamics with finite time-scale synapses. Indeed, it is a general phenomenon in the predictive coding framework that a group of neurons compete to correct the same error, and delays preclude timely inhibition, resulting in pathological synchrony known as the hipster effect [13]; noise helps diversify the dynamics, assuaging the effect. While observations from simulations are insightful, however, we lack a quantitative understanding of this fascinating interplay between the delay and the level of noise. How does coding fidelity depend on the length of the delay and the level of noise? And given a delay, what is the highest achievable fidelity and the associated optimal level of noise? Indeed, the simple, efficient spiking network of the predictive coding framework presents a foundational scenario in which to expand the study of stochastic facilitation [14].

We address these fundamental questions by going beyond simulations to derive analytical expressions for coding fidelity as a function of noise level and small delays in tightly balanced networks of LIF neurons. Previous work [15] derived similar expressions for non-spiking "rate" neurons by adapting the predictive coding framework [7] to non-spiking neurons. Our work takes a step closer toward understanding efficient coding in biological neural networks by explicitly including the spiking nature of neural communication in the brain. Indeed, when considering efficiency, spikes are important because action potentials account for a large portion of the brain's energy expenditure [16] and provide a form of digital communication, which may allow the brain to tap into the efficiency associated with a hybrid analog-digital computing system [17, 18]. Moreover, experiments have shown that spike-timing conveys information in several brain regions [19–21]. Thus, we hope that our analytical insights here provide a foundation for further investigation into the interplay of noise and delays in efficient cortical circuits.

Models

Efficient coding with a network of leaky integrate and fire neurons

We ask the question, how well can a spiking network encode a continuous, time-varying input signal in the presence of noise and transmission delays? To operationalize this question, we start with three assumptions: (1) the output signal is linearly decoded from a densely-connected population of spiking neurons, (2) the network minimizes the mean-squared error between its output and the input signal, and (3) for brevity, we assume the input is 1-dimensional, though our results can be extended to multi-dimensional signals. Thus we consider a scalar input signal $x(t)$, and the network's scalar output $\hat{x}(t)$—the network's estimate for $x(t)$. The network itself is a densely-connected recurrent circuit of N leaky integrate and fire (LIF) neurons. The activity of the i'th neuron (where $i = 1, \ldots, N$) is described by the spike train, $o_i(t) = \sum_k \delta(t - t_i^k)$, where $\delta(\cdot)$ is the Dirac δ-function representing a single spike, and t_i^k is the time of the k'th spike of the i'th neuron. To convert this discrete spiking activity into a smoother output signal, the spike trains $o_i(t)$ are first passed through a linear filter to yield the

instantaneous firing rates $r_i(t)$, which obey

$$\tau \dot{r}_i(t) = -r_i(t) + \tau o_i(t), \tag{1}$$

where the dot ($\dot{}$) represents derivative with respect to time, and τ is the time-constant of the filter. And second, these firing rates $r_i(t)$ are linearly summed to yield the network's estimate $\hat{x}(t)$:

$$\hat{x}(t) \coloneqq \frac{1}{N}\sum_{i=1}^{N} w_i r_i(t), \tag{2}$$

where the $w_i \in \mathbb{R}$ are the weights of the linear decoder.

The network's objective is to achieve $\hat{x}(t) \approx x(t)$ by minimizing the mean-squared-error $\varepsilon^2 = \langle (x(t) - \hat{x}(t))^2 \rangle_t$, where the angular brackets $\langle \cdot \rangle_t$ denote average over time. In order to simplify our study of the mean-squared-error ε^2, we choose to work with inputs $x(t)$ that vary slowly compared to the spiking network's timescale τ. With this choice, we can treat the input $x(t) = x$ as effectively constant, and thus ε^2 can be written as $\varepsilon^2 = (\langle \hat{x}(t) \rangle_t - x)^2 + \langle (\delta \hat{x}(t))^2 \rangle_t$, where we have substituted in $\delta \hat{x}(t) \coloneqq \hat{x}(t) - \langle \hat{x}(t') \rangle_{t'}$ and used the fact that $\langle \delta \hat{x}(t) \rangle_t = 0$. Importantly, we see that ε^2 can be divided into two contributions: a contribution from the bias $(\langle \hat{x}(t) \rangle_t - x)^2$ and a contribution from the variance $\langle (\delta \hat{x}(t))^2 \rangle_t$. Since the bias—here, a constant—could be deterministically removed [15], we focus on computing the contribution from the variance, which is simply the square of the standard deviation of the readout,

$$\sigma_{readout} = \sqrt{\langle (\hat{x}(t) - \langle \hat{x}(t') \rangle_{t'})^2 \rangle_t}. \tag{3}$$

We henceforth refer to $\sigma_{readout}$ as the readout error—an inverse measure for coding fidelity.

The dynamics of the N densely-connected LIF neurons are given by the equation

$$\tau \dot{V}_i(t) = -\lambda_V V_i(t) + I_i(t) - \tau J_{ii} o_i(t) - \tau \sum_{j \neq i} J_{ij} o_j(t - \Delta) + \sqrt{\tau}\sigma \eta_i(t), \text{ and}$$
$$\text{and neuron } i \text{ emits a spike when } V_i > T. \tag{4}$$

where $V_i(t)$ is the membrane potential of the i'th neuron, $T = \frac{1}{2}$ is the firing threshold (in the absence of noise, delay, and with small leak, one will find that this choice of threshold will make the average membrane potential approximately zero when the estimation error is zero [7]), λ_V controls the strength of the leak, $I_i(t)$ is the input current, J_{ii} implements the neural self-reset, J_{ij} implements dense connectivity, Δ is the spike propagation delay, and the $\eta_i(t)$ are independent unit-Gaussian noise with σ controlling the membrane noise level.

Now the critical question is, how do we choose the input $I_i(t)$ and recurrent connectivity J_{ij} so that the noisy, delayed, discontinuous, and nonlinear dynamics of Eq 4 result in minimal readout error $\sigma_{readout}$? We start with the predictive coding framework proposed in [7] that specifies these network properties given fixed readout weights $\{w_i: 1 \leq i \leq N\}$. The framework does so by minimizing the mean-squared-error in a setting in which there are no delays ($\Delta = 0$), and noise is negligible. The resulting network [7] has

$$J_{ij} = w_i w_j \tag{5}$$

$$I_i(t) = N w_i x(t). \tag{6}$$

This network exhibits tight balance: the $O(N)$ excitatory input currents $I_i(t)$ are matched by the $O(N)$ inhibitory terms involving J_{ij} in Eq 4. (The weights w_i and the input $x(t)$ are $O(1)$, and

each neuron, on average, spikes at an $O(1)$ rate, hence a total $O(N)$ inhibition when summing across the population.) Tight balance, i.e., balance of $O(N)$, can be contrasted with balance of $O(\sqrt{N})$, known as classical balance [22], or balance of $O(<\sqrt{N})$, representing loose or no balance. Importantly, tight balance facilitates the superclassical $O(1/N)$ scaling of the readout error in [7]—as was shown in [15]—and thus we specialize in this work to tight balance. And for simplicity, we specialize to uniform readout weights, $w_i = 1\ \forall i$, which corresponds to a population of neurons with the same tuning curve. In the following, we study how the nonzero delay Δ and non-negligible noise level σ impact the readout error $\sigma_{readout}$.

Soft-threshold model

In addition to the LIF model, we introduce a soft-threshold model to study the effects of noise and delays in a simpler setting. In the soft-threshold model, the membrane potentials obey

$$\tau \dot{V}_i(t) = I_i(t) - \tau J_{ii} o_i(t) - \tau \sum_{j \neq i} J_{ij} o_j(t - \Delta),$$

$$\text{and neuron i emits spikes with probability rate } \rho(V_i) = \begin{cases} \rho, & V_i > T \\ 0, & V_i \leq T \end{cases}. \quad (7)$$

The soft-threshold model is also known as the escape-rate model [23], and has been used in prior work on the predictive coding framework [12] and in fitting neural spike train recordings to Generalized Linear Models (GLMs) [24].

Notably, the probabilistic firing of the neurons introduces variation in spike-timing. This is similar to how in the LIF model (Eq 4) the noise term $\sqrt{\tau}\sigma\eta_i(t)$ accumulates over time in the membrane potentials, which also results in variation in spike-timing. Intuitively, the standard deviation, $1/\rho$, of the exponentially-distributed spike-times under the probabilistic firing rate ρ corresponds to an effective noise level, which we can tune by adjusting ρ. However, note that in contrast to the LIF model, which requires the leak term $\lambda_V V_i(t)$ to bound accumulated variability from the noise term $\sqrt{\tau}\sigma\eta_i(t)$, we will see in the next section that probabilistic firing introduces naturally-bounded spike-time variability. Thus, the leak term $\lambda_V V_i(t)$ is not necessary in the soft-threshold model, and so we do not include it in Eq 7 for simplicity.

Importantly, for the $\rho \to \infty$, zero delay ($\Delta = 0$) limit, the soft-threshold model becomes equivalent to the original formulation of the predictive coding framework with hard threshold [7]. The LIF model also becomes equivalent to [7] in the zero noise ($\sigma = 0$), zero delay ($\Delta = 0$) limit. Thus, both models serve as good starting points for analyzing the predictive coding framework with small delays and small noise, as we will see below. We analyze the soft-threshold model in addition to the more complex LIF model because it offers simpler derivations, but yields similar conclusions as the more complex LIF model.

Results

Overall behavior of efficient coding spiking models

To understand the nominal dynamics of the LIF and soft-threshold models, let us consider encoding the constant input signal $x(t) = 1$, and first consider the dynamics of the LIF model with a large number of neurons N. The input current $I_i(t)$ (Eq 6) becomes $I_i(t) = N$ with our choice of decoding weights $w_i = 1$, and the connectivity strengths J_{ij} become $J_{ij} = 1\ \forall i, j$ (Eq 5). When there are no spike transmission delays ($\Delta = 0$) and no noise ($\sigma = 0$), the dynamics of every membrane potential $V_i(t)$ are identical (Eq 4): a spike from any neuron inhibits all membrane potentials equally, instantaneously, and simultaneously; and any differences in the membrane

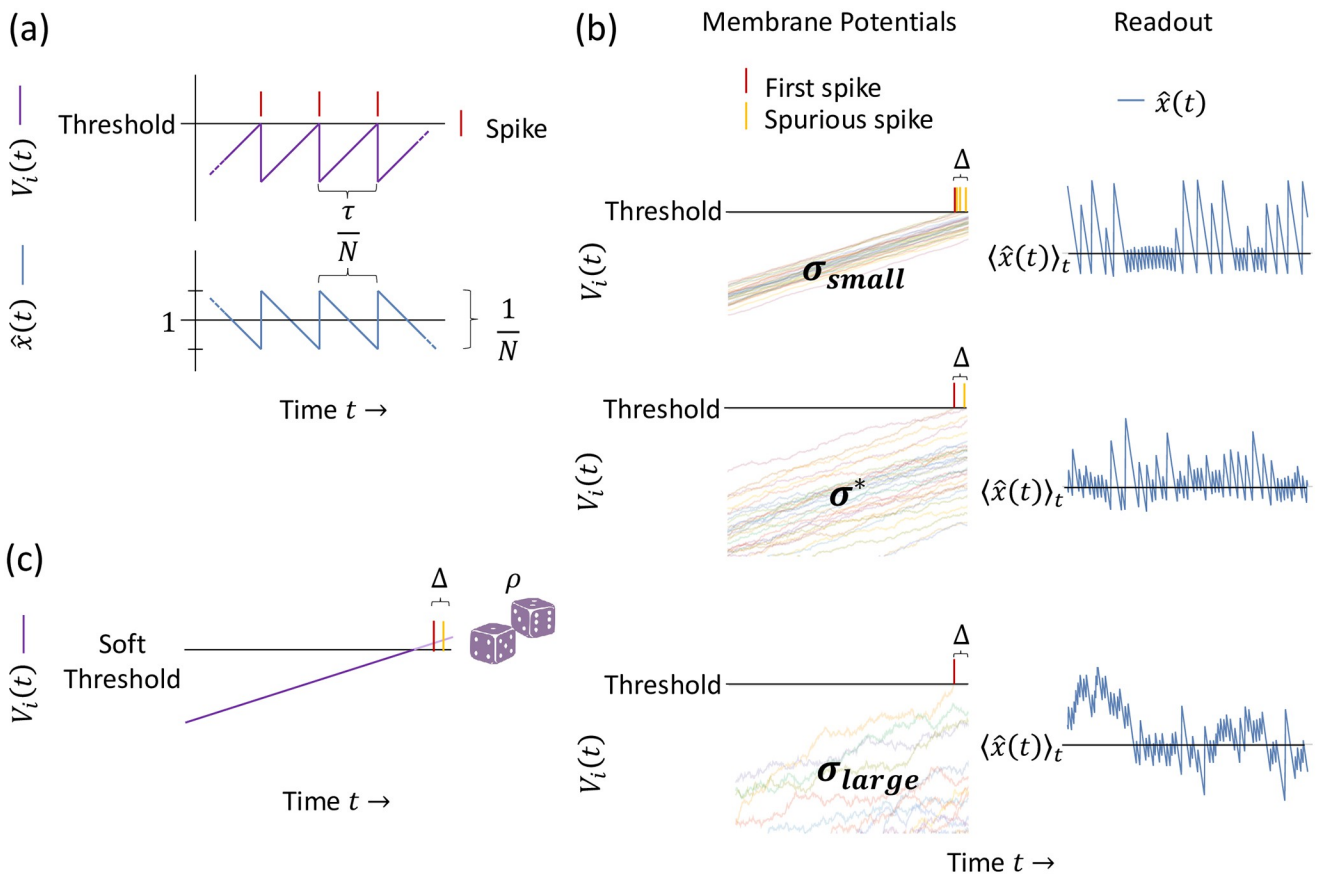

Fig 1. Tight-balance spiking network dynamics and readout. (a) Nominal dynamics. When there are no spike propagation delays and zero noise, the membrane potentials (purple) follow the same trajectory in time. When the population reaches threshold, one neuron's spike (red) instantaneously inhibits all the neurons, preventing further spikes. This produces perfectly regular spikes, like clockwork with an approximate period of $\frac{\tau}{N}$. Each spike contributes $\frac{1}{N}$ to the readout, creating a tight, zig-zag approximation $\hat{x}(t)$ (blue) for the encoded signal, $x(t) = 1$ in this case. (b) The effect of delays and noise. When delays are present and noise is added to the membrane potentials (left, multicolor), two effects appear that decrease the fidelity of the readout $\hat{x}(t)$ (right, blue): variation in spike-timing and synchronous spurious spikes. The noise on the membrane potentials (σ) creates variations in the time it takes membrane potentials to reach threshold—a deviation from the perfectly regular spikes in (a). And after the first neuron in the population crosses threshold and spikes (red), there is a delay Δ until the other neurons receive inhibition, and thus some extra neurons may spike—spurious synchronous spikes (yellow). Given a fixed delay, too little noise σ_{small} does not spread the membrane potentials enough to prevent a large number of spurious spikes, and too much noise σ_{large} destroys the fidelity of the code altogether. An optimal noise level exists, σ^*. (c) Soft-threshold model. In the soft-threshold model, neurons spike probabilistically once their membrane potentials surpass threshold, with a spiking probability rate ρ. As ρ is varied (not illustrated), one finds a relationship analogous to the noise level trade-off for the LIF model shown in (b): too small ρ creates large variations in spike times, and too large ρ creates many spurious spikes during the delay. Thus, an optimal ρ^* exists.

https://doi.org/10.1371/journal.pcbi.1010593.g001

potentials' initial conditions $V_i(0)$ are forgotten on the time-scale τ/λ_V due to the leak term $-\lambda_V V_i(t)$. We are interested in continuously operating networks, so let us consider the network dynamics after a long time $t \gg \tau/\lambda_V$, in which the initial conditions are indeed forgotten. In this case, the membrane potentials $V_i(t)$ are traveling together toward threshold, driven by the input current $I_i(t) = N$ (Fig 1a, top). When a membrane potential reaches threshold, the neuron fires a spike, which immediately self-resets the membrane potential by 1 via the $-\tau J_{ii} o_i(t)$ term and decrements all other membrane potentials by 1 through the $-\tau \sum_{j \neq i} J_{ij} o_j(t)$ term (Eq 4). (Note that here we have assumed that some infinitesimal variation in the membrane potentials persists, e.g., an infinitesimal remnant from the forgotten initial conditions; thus, when the membrane potentials approach threshold, a single membrane potential hits threshold and spikes an instant before the rest of the membrane potentials, allowing sufficient, i.e., infinitesimal, time for the

instantaneous inhibition to prevent additional neurons from spiking.) The membrane potentials then continue to be driven by the input current $I_i(t) = N$, and it takes a time of approximately τ/N for a membrane potential to reach threshold again, where we have assumed that the leak term $-\lambda_V V_i(t)$ is small relative to the $O(N)$ driving current because the membrane potentials themselves are $V_i(t) = O(1)$. When a membrane potential reaches threshold after time of approximately τ/N, it fires a spike, and this process repeats: the network produces spikes like clockwork, with an approximate period of τ/N.

Given this network spiking pattern, we can understand the corresponding readout trajectory $\hat{x}(t)$ by recalling that the readout $\hat{x}(t)$ is a sum of instantaneous firing-rates $r_i(t)$ uniformly weighted by $\frac{1}{N}$ (Fig 1a, bottom). The mean readout $\langle \hat{x}(t) \rangle_t = 1$ because the time constant of the instantaneous firing rates $r_i(t)$ is τ (Eq 1), $\tau \times 1/(\tau/N) = N$ spikes occur during a time τ, and the firing rates are weighted by $\frac{1}{N}$ in the decoder (N spikes $\times \frac{1}{N} = 1$). This mean matches the desired signal, $x(t) = 1$. Furthermore, the readout trajectory $\hat{x}(t)$ forms a zig-zag around the mean value because the readout $\hat{x}(t)$ simply exponentially decays with time constant τ between spike-times, which is approximately linear because the time between spikes τ/N is small compared to τ (since N is large). And the zig-zag is tight; it has magnitude $O(1/N)$, because each spike contributes $\frac{1}{N}$ to the readout. Thus we see here the superclassical $O(1/N)$ scaling of the readout error, because each individual spike is precisely timed to optimally correct the deviations in $\hat{x}(t)$ from $x(t) = 1$.

Next, let us consider the addition of noise $\sigma > 0$ and spike propagation delay $\Delta > 0$ (Fig 1b). Intuitively, the integration of the independent noise terms $\sqrt{\tau}\sigma\eta_i(t)$ spread out the membrane potentials $V_i(t)$, and thus they no longer share exactly the same trajectory when traveling toward threshold and instead travel in a continuously-fluctuating packet of some finite width. Importantly, both the packet width and the variance in the time to first spike increase with the noise level σ. Now, as the packet travels toward threshold, a top neuron in the packet eventually reaches threshold and spikes. This spike instantly self-resets the firing neuron through the $-\tau J_{ii} o_i(t)$ term, but the inhibition arrives a time Δ later to the other membrane potentials through the $-\tau \sum_{j \neq i} J_{ij} o_j(t - \Delta)$ term. Importantly, during the delay time Δ, all membrane potentials continue to be driven by the strong input current $I_i(t) = N$, and so there is a possibility that additional membrane potentials reach threshold before they receive the inhibition from the first neuron's spike. As these neurons hit threshold, they produce extra, spurious spikes that create undesirably large deviations in the readout $\hat{x}(t)$, leading to high readout error. Thus for a fixed delay Δ, we see a trade-off in the noise level: for small noise σ_{small}, the membrane potentials travel in a tight packet, and thus there are likely many membrane potentials crossing threshold during the delay Δ resulting in many spurious spikes and a large readout error. And for large noise σ_{large}, the membrane potentials are more spread out, reducing the number of spurious spikes during the delay Δ, but at the cost of introducing a large deviation in the time-to-spike for the first-spiking, top neuron in the packet. This implies that there exists some intermediate optimal noise level σ^* that balances these effects to minimize the readout error. Below we will analytically compute the readout error $\sigma_{readout}$ as a function of the noise level σ and the delay Δ and calculate this optimal noise level σ^* and the associated minimal readout error $\sigma^*_{readout}$.

Importantly, the soft-threshold model exhibits the same trade-off (Fig 1c). For small probabilistic firing rate ρ, the number of spurious spikes during the delay Δ is small, but the standard deviation in the time-to-spike of a single neuron is large: $1/\rho$. And for large ρ, the number of spurious spikes is large, but the standard deviation in time-to-spike is small. Thus for some fixed delay Δ, an optimal ρ^* (or equivalently, an optimal noise level $1/\rho^*$) exists that minimizes the readout error $\sigma_{readout}$ in the soft-threshold model.

In the following subsections, we quantitatively elucidate this trade-off between minimizing spurious spikes and minimizing spike-timing variability, and we calculate $\sigma_{readout}$ as a function of noise and delay for the soft-threshold model and the LIF model in turn, for networks with a large number of neurons N. We begin with the simpler soft-threshold model, as it will allow us to derive an exact expression for $\sigma_{readout}$ in the limit of small delays and noise. Then, we will analyze small noise and delays in the LIF model, and ultimately derive an approximate upper-bound for $\sigma_{readout}$ and see that both models exhibit the same behavior. We corroborate our analytic results with simulations, whose details are provided in Section B Simulation details in S1 Appendix.

Analysis of noise, delays, and coding error in the soft-threshold model

In this section, we calculate the contributions of spike-time variability and spurious spikes to the readout error $\sigma_{readout}$ for the soft-threshold model. We provide concise derivations here, and detailed derivations in S1 Appendix. To begin, we consider the simple scenario in which the membrane potentials $V_i(t)$ all start with initial condition $V_i(0) = 0$, as would be the case when there is no external input, and if there were an additional leak term to ensure all membrane potentials decay to 0. Then, we consider turning on the dynamics, Eq 7. The input current $I_i(t) = N$ drives the membrane potentials toward threshold together, and all membrane potentials reach threshold simultaneously. At this point, the population begins emitting spikes probabilistically with rate $N\rho$, as there are N membrane potentials superthreshold individually firing with probability rate ρ.

Statistics of spike-time variability. After some time above threshold, the population eventually emits a spike. The variation in this first-spike time (where time is measured relative to when the population had crossed threshold) is a departure from the optimal clockwork one-spike-every-τ/N spiking pattern for a network with zero delay and zero noise as described above, and thus this first-spike time variability increases the readout error $\sigma_{readout}$. To quantify the increase, we therefore wish to describe the statistics of this first-spike time. Now, since the population is emitting spikes probabilistically with a constant rate ($N\rho$), the time it takes until the first spike occurs is an exponentially-distributed random variable. For an exponential distribution, the mean and standard deviation are given by the reciprocal of the rate, thus here the first-spike time has mean $\frac{1}{N\rho}$ and standard deviation $\frac{1}{N\rho}$ (Fig 2a, top). And naturally, this first-spike time variability also creates fluctuations in the readout, with standard deviation $\frac{1}{N\rho\tau}$, which arises from the standard deviation of the first-spike time $\frac{1}{N\rho}$, multiplied by the magnitude of the $-\frac{1}{\tau}$ slope of the approximately linear decay of the readout (Fig 2a, bottom).

Statistics of spurious spikes induced by delays. After the first spike occurs, one membrane potential is instantly reset, and the other $N - 1$ membrane potentials continue to spike probabilistically during the spike propagation delay Δ. Hence, spurious spikes may occur, and we would like to describe their statistics to quantify how they increase the readout error $\sigma_{readout}$. To this end, the mean number of spurious spikes, λ, is the time Δ multiplied by the total spiking probability rate of the population, which is $(N - 1)\rho \approx N\rho$; this yields

$$\lambda = \Delta N \rho. \tag{8}$$

And here, for ease of analysis, we assume the delay Δ is much smaller than the $O(1/N)$ network interspike interval (see the second paragraph of Results for the $O(1/N)$ interspike interval). With this assumption, differences in the N membrane potentials due to delayed inhibition vanish before the membrane potentials impinge upon threshold together again; this guarantees the simple scenario that we are considering here, where N neurons always reach threshold together (see Section A.2 Readout error for the soft-threshold model in S1 Appendix for

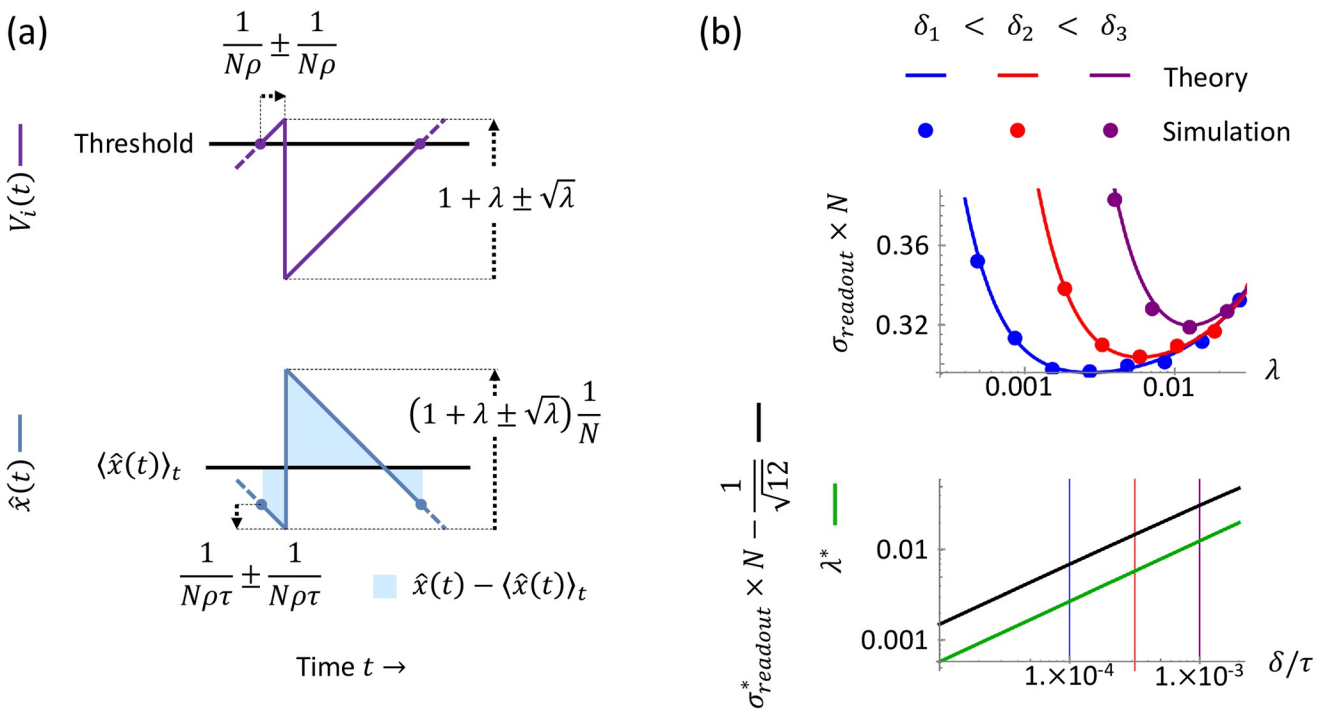

Fig 2. Soft-threshold readout error. (a) The soft-threshold and the delay create variations in the membrane potential dynamics $V_i(t)$, which in turn create variations in the readout $\hat{x}(t)$. When the membrane potentials (top, purple) surpass threshold, the neurons spike probabilistically, and the first-spike time is an exponential random variable with standard deviation $\frac{1}{N\rho}$. After a first spike, the number of spurious spikes that occur during the delay is a Poisson random variable, with standard deviation $\sqrt{\lambda}$, and each spike inhibits the membrane potentials $V_i(t)$ by 1 through recurrent connectivity (see the first paragraph of Results for recurrent connectivity). These variations in spike-timing and spurious spikes carry through to the readout $\hat{x}(t)$ (bottom, blue). Note that since the network input is constant, the readout encoding this input should produce a constant output as closely as possible; however, these variations instead increase the deviation (light blue shaded) from the mean readout $\langle \hat{x}(t) \rangle_t$ (black horizontal line). (b) Readout error as a function of the mean number of spurious spikes λ and the delay δ. Top: for three different values of delay (blue, red, purple), λ is varied in computer simulations (N = 32, dots) and Eq 10 (solid curves), revealing both the U-shaped dependence of the readout error $\sigma_{readout}$ and an excellent match between theory and experiment. Bottom: the optimal readout error $\sigma^*_{readout}$ (black) and the associated optimal λ^* increase as a function of delay δ according to Eqs 12 and 11.

https://doi.org/10.1371/journal.pcbi.1010593.g002

details). Thus we introduce the parameter δ by the definition

$$\Delta := \frac{\delta}{N} \tag{9}$$

where we assume $\delta \ll \tau$ so that the delay Δ is much less than the network inter-spike interval. Note that with these definitions we have $\lambda = \delta\rho$. And since we are interested in high-performing networks, i.e. small readout error $\sigma_{readout}$, we focus on the limit where $\lambda \ll 1$, where there are few undesirable spurious spikes. In this limit, the number of spurious spikes is simply Poisson-distributed with mean λ and standard deviation $\sqrt{\lambda}$. Importantly, the variability in the number of spurious spikes creates variability in the next time the population reaches threshold together because each spike decrements the membrane potentials, thus the input current $I_i(t) = N$ takes a variable amount of time to drive the membrane potentials to threshold again (Fig 2a, top). And the fluctuation in the readout due to spurious spikes has standard deviation $\sqrt{\lambda}/N$, as each spike contributes $\frac{1}{N}$ to the readout (Fig 2a, bottom).

Averaging across time to calculate the readout error. To calculate $\sigma_{readout}$, we recognize that the time-average of the squared deviation of the readout in Eq 3 is by definition an integral of the squared deviation over a long time interval, divided by the interval

duration—i.e., $\langle(\hat{x}(t) - \langle\hat{x}(t')\rangle_{t'})^2\rangle_t = \lim_{\mathcal{T}\to\infty}\frac{1}{\mathcal{T}}\int_0^{\mathcal{T}}(\hat{x}(t) - \langle\hat{x}(t')\rangle_{t'})^2 dt$; thus, we turn our attention to computing this integral. Importantly, because of its long time interval, the integral includes many sequences of the population reaching threshold, a variable amount of time until a first spike occurs, the production of a random number of spurious spikes during the spike propagation delay, and the return of the population to threshold; thus it effectively sums over all possible values of first-spike time and number of spurious spikes, with the values' frequencies weighted by the probability distributions that we quantified above. To help simplify the integral, we approximate the timing of the spurious spikes by treating them as if they occur at the same time as the first spike because we are considering small delays $\delta \ll \tau$; this is illustrated in Fig 2a by the lack of time differences between the random number of spikes, which are depicted together as a single vertical deviation at a single instant. Then performing the integration (see Section A.2.2 Mean readout error in S1 Appendix) reveals that we can express the combined effect of fluctuations from spike-time variability and spurious spikes via the sum of their variances—intuitively, one may expect these independent sources of variation to add in this way, as variances add for independent random variables in general; the integration yields the readout error $\sigma_{readout}$ to leading order in small delay δ and small mean number of spurious spikes λ:

$$\sigma_{readout} = \frac{1}{N}\sqrt{\frac{1}{12} + \frac{\delta^2}{\lambda^2\tau^2} + \lambda}, \tag{10}$$

where under the square root, the first term ($\frac{1}{12}$) arises from the baseline readout error in the case of zero delays and zero noise, the second term is the contribution from spike-time variability ($\frac{1}{\rho^2\tau^2} = \frac{\delta^2}{\lambda^2\tau^2}$ by the relation $\lambda = \delta\rho$), and the third term is the contribution from the mean number of spurious spikes (λ). Importantly, we note that the mean number of spurious spikes λ can simply be thought of as a reparameterization of the probability rate ρ.

Furthermore, we can minimize Eq 10 with respect to λ to find the minimal readout error $\sigma^*_{readout}$ and optimal noise level, parameterized via the optimal mean number of spurious spikes λ^*. This yields

$$\lambda^* = 2^{1/3}(\delta/\tau)^{2/3}, \tag{11}$$

and

$$\sigma^*_{readout} = \frac{1}{N}\sqrt{\frac{1}{12} + \frac{3(\delta/\tau)^{2/3}}{2^{2/3}}} \tag{12}$$

$$\sigma^*_{readout} \approx \frac{1}{N}\left[\frac{1}{\sqrt{12}} + \frac{3^{3/2}}{2^{2/3}}(\delta/\tau)^{2/3}\right]. \tag{13}$$

We corroborate our analytic results with simulations in Fig 2b.

Analysis of noise, delays, and coding error in the LIF model

In this section, we study the readout error $\sigma_{readout}$ as a function of small delays and noise for the LIF model. We provide concise derivations here, and detailed derivations in S1 Appendix. For ease of exposition, we first examine the LIF model with zero delay ($\Delta = 0$) and small noise ($\sigma > 0$) to isolate the contribution of spike-time variability to the readout error $\sigma_{readout}$, as no spurious spikes can occur when spike propagation is instantaneous. Then, we will introduce and study small delay ($\Delta > 0$). We will make use of several approximations and inequalities in

our analysis of the LIF model, which will result in our final analytic expression for $\sigma_{readout}$ being an approximate upper-bound on the actual error.

Readout error in the LIF model with no delays. To understand how the readout error $\sigma_{readout}$ depends on small noise in the case of zero delay, we begin by describing the behavior of the population of membrane potentials under the dynamics of Eq 4 with delay $\Delta = 0$. To gain some intuition, let us first consider what the membrane potential dynamics look like if spiking is disabled, i.e., the firing threshold T is taken to infinity. Importantly, the inhibitory terms $-\tau J_{ii} o_i(t)$ and $-\tau \sum_{j \neq i}^{N} J_{ij} o_j(t)$ are zero, and thus Eq 4 becomes the well-known Ornstein–Uhlenbeck (OU) process [25]. The stochastic process $V_i(t)$ after some time $t \gg \tau$ approaches a stationary Gaussian process distribution with a mean of N/λ_V and a temporal autocovariance given by

$$cov(V_i(s), V_i(t)) \coloneqq \langle (V_i(s) - \mathbb{E}(V_i(s)))(V_i(t) - \mathbb{E}(V_i(t))) \rangle \tag{14}$$

$$= \frac{\sigma^2}{2\lambda_V} \left(e^{-\frac{\lambda_V}{\tau}|t-s|} \right). \tag{15}$$

Thus at any moment of time, the membrane potentials are distributed as a Gaussian packet of constant width

$$\sigma_{OU} \coloneqq \frac{\sigma}{\sqrt{2\lambda_V}}. \tag{16}$$

and over a mixing time scale of τ/λ_V, the membrane potentials diffuse and forget their past values due to the external noise. Note that we are interested in analyzing continuously operating networks, thus these stationary statistics will be relevant in our analysis.

Next, let us consider reintroducing the effects of spiking, returning to the original dynamics of Eq 4 with delay $\Delta = 0$. We consider any initial condition in which the membrane potentials are all subthreshold with $V_i(0) < T, \forall i$. As the membrane potentials travel toward threshold, driven by strong input current $I_i(t) = N$, the top membrane potential will reach threshold first and spike. Importantly, this spike instantly, simultaneously, and uniformly inhibits all membrane potentials. Thus only the mean membrane potential is decremented, but the relative positions of the membrane potentials are preserved—i.e., the entire distribution is shifted lower by a constant value. Thus, this observation reveals a simple overall behavior: the membrane potentials are traveling in a Gaussian packet of constant width, with the entire packet being periodically decremented each time the top neuron in the packet reaches threshold and fires a spike.

With this membrane potential packet dynamics in mind, we will first quantify the spike-time variability of the first spike. It is important to understand this spike-timing variability as it contributes directly to the readout error $\sigma_{readout}$ as we shall see below. We first start by considering the simplifying case of small $\lambda_V \ll 1$. This limit affords two useful simplifications: (1) the membrane potentials fluctuate slowly within the Gaussian packet, with long mixing time-scale τ/λ_V (Eq 15), and thus the same neuron repeatedly wins the race toward threshold (Fig 3a), and (2) the threshold-crossing time of this neuron is well-approximated by the first-passage time of ordinary Brownian motion with drift, as Brownian motion is simply an OU process with $\lambda_V = 0$. Thus on time-scales shorter than the long mixing time τ/λ_V, we need only consider the dynamics of the neuron with the top membrane potential in the packet, as opposed to the entire population. Hence, we can express spike-time variability as random interspike interval durations, which are drawn from the Brownian motion first-passage time distribution. Importantly, the statistics of the first-passage time t_{fp} for Brownian motion are

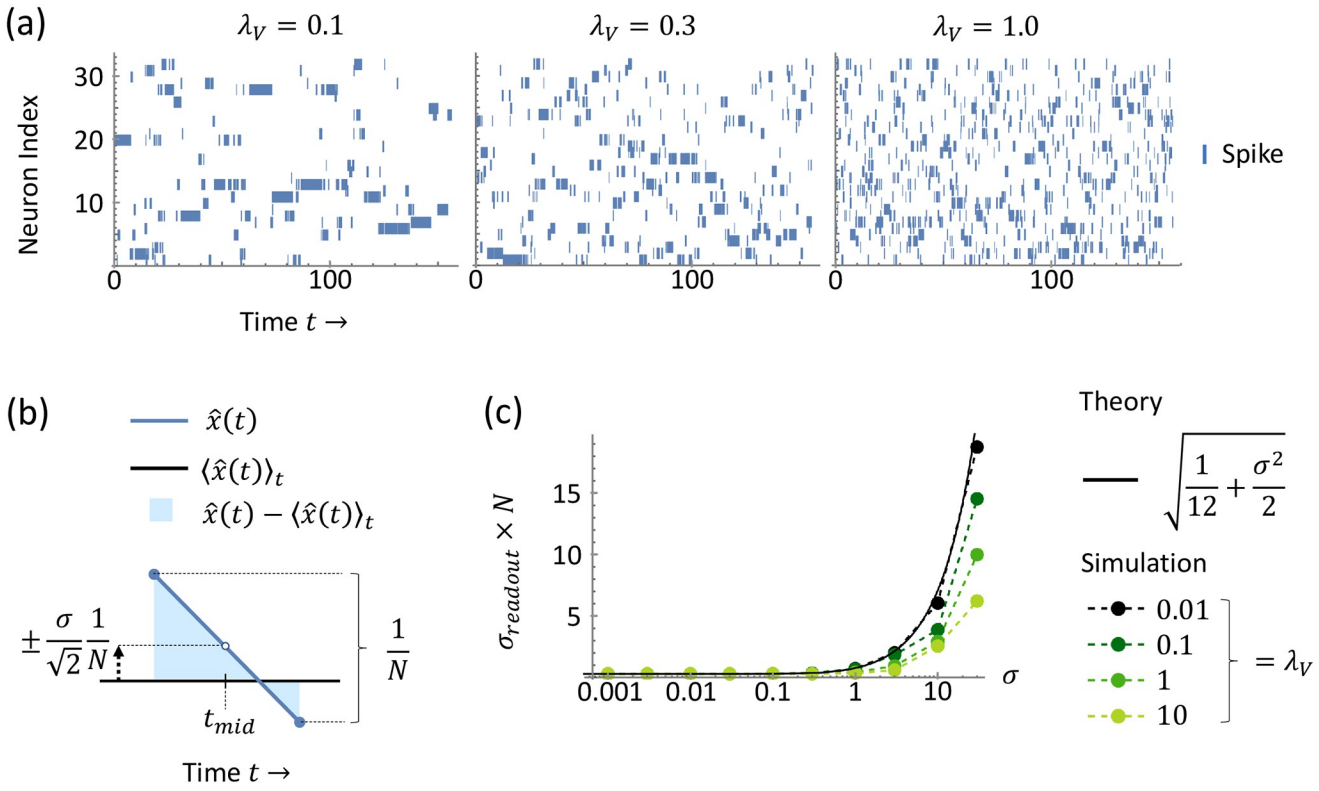

Fig 3. LIF model with zero delay. (a) Spike raster plots from simulations ($N = 32$) for three different values of the membrane potential leak λ_V. Notably, for small λ_V, we observe long runs in which the same neuron repeatedly spikes. Thus in the small λ_V limit, the next spike time is well-approximated by considering only the possibility that the same neuron spikes again. (b) Readout $\hat{x}(t)$ (blue) and its deviation $\hat{x}(t) - x(t)$ (light blue shaded) from the mean $\langle \hat{x}(t) \rangle_t$ for a single interspike interval. The variations in interspike interval durations accumulate to produce a variation in the readout with standard deviation $\frac{\sigma}{\sqrt{2}} \frac{1}{N}$, using the approximation from (a). (c) Readout error $\sigma_{readout}$ as a function of noise σ for different values of λ_V. Integrating the deviation illustrated in (b) across time yields the readout error in the small λ_V limit, Eq 19 (black). Simulations (dots on dashed lines, $N = 64$) with larger λ_V are upper-bounded by Eq 19.

https://doi.org/10.1371/journal.pcbi.1010593.g003

known [26]. For a particle undergoing Brownian motion with time constant τ, noise σ, drift μ, initial position x_0, and the goal of reaching a threshold θ, the mean and variance of its first-passage time are $\langle t_{fp} \rangle = \frac{(\theta - x_0)\tau}{\mu}$ and $\langle (t_{fp} - \langle t_{fp} \rangle)^2 \rangle = \frac{(\theta - x_0)\sigma^2\tau^2}{\mu^3}$. In our case, the dynamics of the top membrane potential have time constant τ, noise σ, drift $I_i(t) = N$, initial membrane potential of $-\frac{1}{2}$ (the threshold $T = \frac{1}{2}$ minus the self-reset of 1 through the $-\tau J_{ii} o_i(t)$ term in Eq 4), and the goal of reaching the threshold $T = \frac{1}{2}$. This yields the moments $\langle t_{fp} \rangle = \frac{\tau}{N}$ and $\langle (t_{fp} - \langle t_{fp} \rangle)^2 \rangle = \frac{\sigma^2\tau^2}{N^3}$.

Now, to work toward calculating $\sigma_{readout}$, which involves integrating $\hat{x}(t)$ over time, we start by considering the readout $\hat{x}(t)$ at a particular time t, under the influence of the spike-time variability we just quantified for small λ_V. Recall that the readout $\hat{x}(t)$ is a uniform sum ($w_i = 1$) of instantaneous firing rates $r_i(t)$ (Eq 2) and that the firing rates $r_i(t)$ are simply leaky integrations of the spike-trains $o_i(t)$ (Eq 1). Thus we can write the readout $\hat{x}(t)$ as a sum of decaying exponentials, with one exponential for each spike-time t_k in the past:

$$\hat{x}(t) = \frac{1}{N} \sum_{k=1}^{\infty} e^{-\frac{\Delta t_k}{\tau}}, \tag{17}$$

where $\Delta t_k \coloneqq t - t_k$. Recalling that the interspike interval durations are random variables drawn from the first-passage time distribution, we recognize that the Δt_k are also random variables, which are simply the sum over past interspike interval durations, t_{fp}^j:

$$\Delta t_k = t - t_1 + \sum_{j=1}^{k-1} t_{fp}^j. \tag{18}$$

Now importantly, although the interspike interval durations t_{fp}^j themselves are independent because the repeatedly-spiking top neuron always starts afresh at its reset potential after spiking and carries no history of the rest of the membrane potentials, we note that in contrast, the Δt_k are correlated random variables because Δt_l contains all the terms in Δt_k, $\forall l > k$. (Δt_2 contains all the terms in Δt_1, Δt_3 contains all the terms in Δt_2 and Δt_1, and so on.) Thus we have in $\hat{x}(t)$ (Eq 17) an infinite sum of correlated random variables. To evaluate the statistics of $\hat{x}(t)$, in particular its variance which contributes to $\sigma_{readout}$, we make a simplifying approximation. Namely, by the central limit theorem, we take the sum $\sum_{j=1}^{k-1} t_{fp}^j$ in the Δt_k (Eq 18) to be Gaussian because it contains many terms for most k in the sum from $k = 1$ to ∞ in $\hat{x}(t)$ (Eq 17). With this Gaussian approximation, the distribution of Δt_k depends only on the mean and variance of t_{fp}, as opposed to the more complex non-Gaussian first-passage time distribution.

Using this Gaussian approximation, we can calculate the variance of $\hat{x}(t)$ in Eq 17 in the limit of small noise, and we do so for a particular point in time t_{mid}, halfway through a given interspike interval (illustrated in Fig 3b). We find that the variance in the readout $\hat{x}(t_{mid})$ is $\frac{\sigma^2}{2N^2}$ (see Section A.3.1 Readout at a single point in time in S1 Appendix for a detailed calculation). Then finally to compute $\sigma_{readout}$ (Eq 3), we integrate the squared deviation of the readout over a long time interval in the same manner as we did for the soft-threshold model (c.f. the paragraph preceding Eq 10). Namely, we recognize that this long interval of integration is simply comprised of many individual interspike intervals, and within each interspike interval, the deviations $\hat{x}(t) - \langle \hat{x}(t) \rangle_t$ are related to the variable $\hat{x}(t_{mid})$, illustrated as a vertical shift of the readout $\hat{x}(t)$ in Fig 3b (note that t_{mid} here does not denote a single instant in time, but rather refers to the time in the middle of a given interspike interval). And since we have quantified the distribution of $\hat{x}(t_{mid})$, we can perform the integration (see Section A.3.2 Mean readout error in S1 Appendix for details), which yields

$$\sigma_{readout} = \frac{1}{N}\sqrt{\frac{1}{12} + \frac{\sigma^2}{2}}. \tag{19}$$

We compare this expression to simulations, and we find empirically that it matches well for small λ_V (Fig 3c, $\lambda_V = 0.01$); this is to be expected because we used the simplifying case of small $\lambda_V \ll 1$ in our derivation of Eq 19. However, importantly, releasing the assumption that λ_V is small, we find empirically that Eq 19 also serves as an upper-bound for general λ_V (Fig 3c, $\lambda_V = 0.1, 1, 10$).

Readout error in LIF model with delays. Next, we build upon our analysis for zero delay $\Delta = 0$, and calculate the readout error $\sigma_{readout}$ for small delay $\Delta > 0$. The primary additional effect from the introduction of nonzero delay Δ is the possibility of spurious spikes. Spurious spikes increase the readout error $\sigma_{readout}$, as we have seen in the soft-threshold model. Similarly, for the LIF model, we would like to quantify the statistics of spurious spikes and their contribution to the readout error $\sigma_{readout}$.

To calculate the statistics of spurious spikes, recall from our analysis for zero delay $\Delta = 0$ that the membrane potentials travel in a Gaussian packet of width σ_{OU} (Eq 16) toward threshold, and eventually a top neuron in the packet reaches threshold and fires a spike. (See Section

A.4.1 Mean number of spurious spikes in S1 Appendix for details on how small delays merely widen this packet slightly.) Then, during the spike propagation delay Δ, the other membrane potentials continue to travel toward threshold, and may fire extra, spurious spikes. We can estimate the mean number of spurious spikes during Δ by considering the tail of the approximately Gaussian membrane potential packet impinging upon threshold (Fig 4a).

The position of the Gaussian packet at the time of the first spike can be estimated via the condition that the tail probability above threshold T of the Gaussian packet is $1/N$, so that out of N neurons the expected number of neurons to spike is 1 (Fig 4a, top). This tail probability condition approximately determines the location of the packet's mean value $\bar{V} \coloneqq \frac{1}{N}\sum_{i=1}^{N} V_i$ via the condition

$$\frac{1}{N} = \int_T^\infty \frac{1}{\sqrt{2\pi}\sigma_{OU}} e^{-\frac{(V-\bar{V})^2}{2\sigma_{OU}^2}} dV. \tag{20}$$

This condition can be solved to yield the approximate location of the mean $\bar{V}$ at the time of the first spike:

$$\bar{V} = T - \sqrt{2}\sigma_{OU}\ \mathrm{erfc}^{-1}\left(\frac{2}{N}\right), \tag{21}$$

where erfc^{-1} is the inverse complementary error function.

Now we consider the mean number of extra spurious spikes that will occur in the time Δ after this first spike. Note that during this time, the Gaussian packet is moving up towards threshold at rate N/τ. Thus over a time $\Delta = \delta/N$ all membrane potentials in the range $T - \delta/\tau$ to T will further cross threshold (Fig 4a, bottom). Thus we can compute the mean number of spurious spikes λ by integrating the density of membrane potentials within this range:

$$\lambda = N\int_{T-\delta/\tau}^{T} \frac{1}{\sqrt{2\pi}\sigma_{OU}} e^{-\frac{(V-\bar{V})^2}{2\sigma_{OU}^2}} dV. \tag{22}$$

Importantly, we recover the intuition that increased noise reduces the number of spurious spikes. Basically, increasing σ corresponds to larger σ_{OU}, i.e. a wider packet, which in turn reduces the total density of membrane potentials in the range $T - \delta/\tau$ to T that are ready to cross threshold after the first spike occurs. This can be readily understood by inspecting the leading-order expression for Eq 22 for small δ/τ, which is

$$\lambda \approx c(N)\frac{\delta/\tau}{\sigma_{OU}}, \tag{23}$$

where $c(N) \coloneqq \frac{Ne^{-(\mathrm{erfc}^{-1}(2/N))^2}}{\sqrt{2\pi}}$ is a coefficient that grows sub-logarithmically with N, $c(N) < O(\log(N))$. Thus interestingly, the total number of neurons N does not contribute strongly to the mean number of spurious spikes λ. Finally, since we are considering high-performing networks with a small mean number of spurious spikes ($\lambda \ll 1$), we expect the number of spurious spikes to be well approximated by a Poisson distribution—the independent probability of each neuron crossing threshold during the delay gives rise to a binomial distribution for the number of spurious spikes, and a binomial distribution with a large number of trials and small per-trial probability (as is the case here) is well-approximated by a Poisson distribution. This completes our characterization of the spurious spike statistics.

Lastly, to calculate the readout error $\sigma_{readout}$, we integrate the squared deviation in the readout $\hat{x}(t)$ across time, taking into account fluctuations due to: (1) spike-time variability from noise, which we isolated and quantified by analyzing the zero-delay case, and (2) the spurious

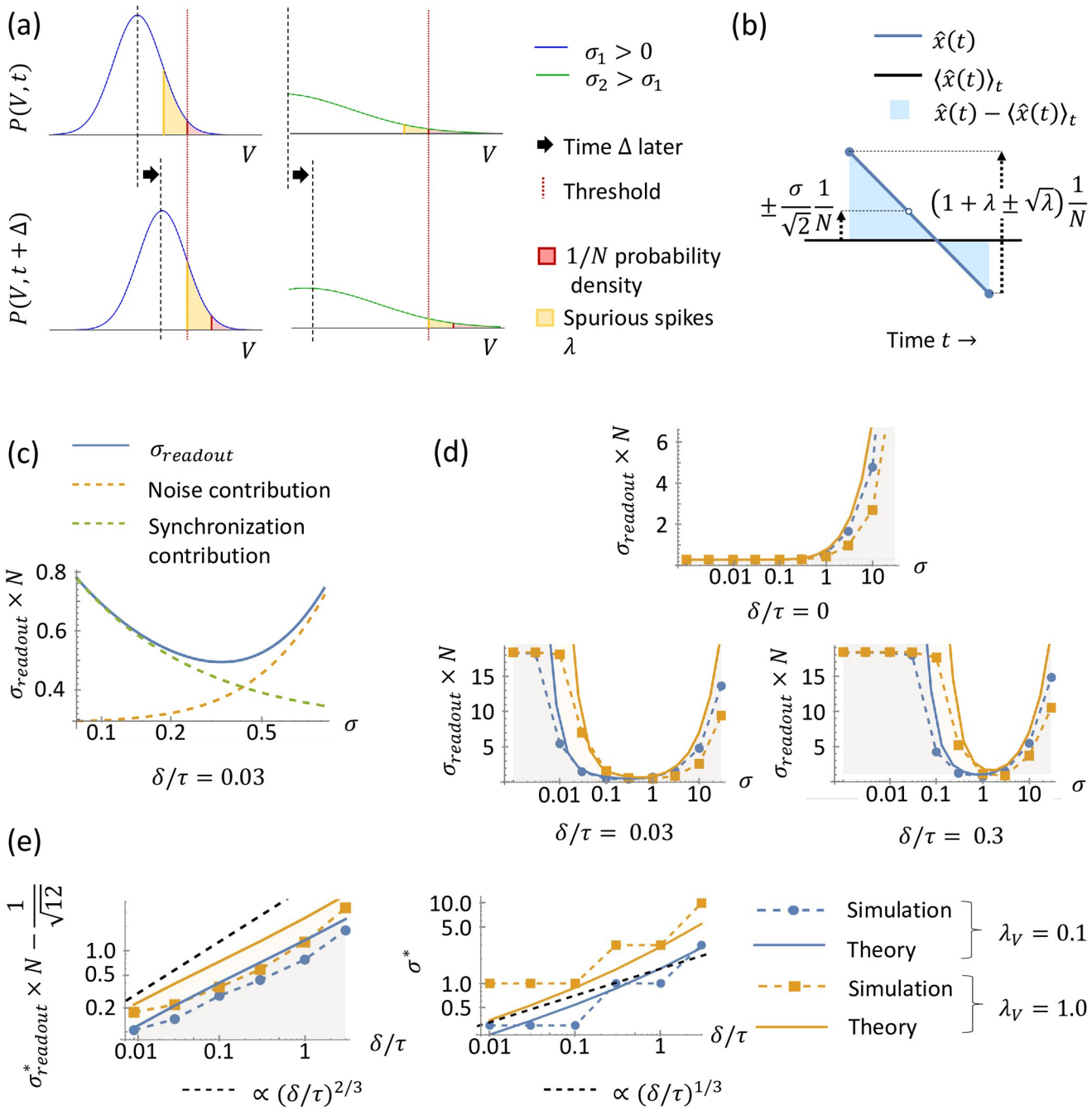

Fig 4. LIF model with nonzero delay. (a) Calculation of the number of spurious spikes. For a nonzero noise level σ_1, the membrane potentials travel in a Gaussian packet with density $P(V, t)$ (blue, top left) toward threshold (vertical red dotted line). The typical position of the packet at the time t when the first neuron spikes is determined by ensuring the tail probability (red, shaded area) above threshold equals $1/N$. During the spike propagation delay Δ, the Gaussian packet continues traveling toward threshold (blue, bottom left), and the mean number of spurious spikes λ is given by the additional probability density that crosses threshold (yellow, shaded area). A larger noise level σ_2 spreads out the Gaussian packet (green, right), thus reducing λ. (b) Readout $\hat{x}(t)$ (blue) and its deviation $\hat{x}(t) - x(t)$ (light blue shaded) from the mean $\langle\hat{x}(t)\rangle_t$. Similar to Fig 3b, the accumulated spike-time variation creates fluctuations in $\hat{x}(t)$ with standard deviation upper-bounded by $\frac{\sigma}{\sqrt{2}}\frac{1}{N}$, but in addition, spurious spikes introduce a Poisson variation in the readout with standard deviation $\sqrt{\lambda}\frac{1}{N}$. (c) Integrating the deviations illustrated in (b) yields an approximate upper-bound for $\sigma_{readout}$, Eq 24 (blue). Conceptually, $\sigma_{readout}$ receives contributions from noise (Eq 24 without the λ term; yellow, dashed), and synchronous spurious spikes (Eq 24 without the $\frac{\sigma^2}{2}$ term; green, dashed). (d) Readout error $\sigma_{readout}$ for varying levels of noise. For zero delay (top), noise is not necessary to prevent spurious spikes, and thus it strictly increases $\sigma_{readout}$. For non-zero delays (bottom), $\sigma_{readout}$ has a U-shaped dependence on σ, and an optimal noise level σ^* exists. The dots/squares on dashed lines represent $\sigma_{readout}$ from simulations ($N = 64$), and solid lines are Eq 24, with the region below shaded, indicating upper-bound. Blue signifies $\lambda_V = 0.1$; yellow $\lambda_V = 1.0$.

(e) Minimal readout error $\sigma^*_{readout}$ and optimal noise level σ^* as a function of delay δ. Minimizing Eq 24 (with higher-order terms, see Eq S91 in S1 Appendix) with respect to σ yields $\sigma^*_{readout}$ (left, solid lines) and σ^* (right, solid lines). $\sigma^*_{readout} - \frac{1}{\sqrt{12}}$ and σ^* asymptotically approach $(\delta/\tau)^{2/3}$ and $(\delta/\tau)^{1/3}$, respectively (dashed black lines). We take the minimal $\sigma_{readout}$ from the simulations in (d) and the associated optimal noise level to generate the dots/squares on the blue/yellow dashed lines, observing that our theory indeed provides an upper-bound for $\sigma^*_{readout}$ and a good estimate for the optimal noise level σ^* in finite-sized simulations.

https://doi.org/10.1371/journal.pcbi.1010593.g004

spikes that we just characterized (Fig 4b). Evaluating the integral for $\sigma_{readout}$ with the composition of these fluctuation sources (see Section A.4.2 Mean readout error in S1 Appendix for details), we obtain the approximate upper-bound

$$\sigma_{readout} \lesssim \frac{1}{N}\sqrt{\frac{1}{12} + \frac{\sigma^2}{2} + \lambda}, \tag{24}$$

to leading order in σ and λ. Importantly, in this calculation we have used the fact that our result for $\sigma_{readout}$ in the zero-delay case (Eq 19) was an approximate upper-bound (for general λ_V, Fig 3c). Also we observed empirically that our calculation for the mean number of spurious spikes λ (Eqs 20 to 22) is an approximate upper-bound as well (see Section A.4.1 Mean number of spurious spikes in S1 Appendix). Thus the composition of these bounds on sources of fluctuation in turn provides an approximate upper-bound on readout error in Eq 24. A plot of our theory for the readout error as a function of noise σ for a fixed nonzero delay δ is shown in Fig 4c. We see that the readout error displays a non-monotonic dependence in the noise level σ, which arises as a trade-off between two competing effects. First, increasing noise contributes to increasing error through added spike-time variation through the middle term in Eq 24. But increasing noise σ also leads to a smaller mean number λ of spurious spikes through desynchronization of the population (see Eqs 23 and 16). This trade-off between reducing spike time variation and preventing spike synchronization leads to an optimal level of noise σ^*.

We compare our approximate upper-bound for the readout error $\sigma_{readout}$ against simulations, and we empirically observe that it indeed bounds the error and reproduces the expected dependence on σ and δ (Fig 4d). Further, we numerically minimize Eq 24 (with higher-order terms, see Eq S91 in S1 Appendix) to obtain an approximate upper-bound for the minimal error $\sigma^*_{readout}$ and an estimate for the associated optimal noise level σ^* for a given delay δ (Fig 4e). Numerically differentiating our calculated $\sigma^*_{readout}$ and σ^* with respect to δ (see Section A.4.2 Mean readout error in S1 Appendix), we find that they grow as

$$\sigma^*_{readout} \lesssim (\delta/\tau)^{2/3}, \tag{25}$$

and

$$\sigma^* \sim (\delta/\tau)^{1/3}. \tag{26}$$

Here, the growth of the minimal error $\sigma^*_{readout}$ (Eq 25) matches that of the soft-threshold model (Eq 12). And furthermore, the growth of the associated optimal noise level σ^* also matches that of the soft-threshold model—the noise level in the soft-threshold model is the standard deviation of the time-to-spike, $\frac{1}{\rho}$, and Eq 11 implies that $\frac{1}{\rho^*} \sim (\delta/\tau)^{1/3}$.

Discussion

In summary, we studied coding fidelity in tightly balanced networks of spiking LIF neurons with small noise and delays by analyzing the standard deviation $\sigma_{readout}$ of a simple linear readout for a slowly-varying 1-D scalar dynamical variable. In contrast to previous works studying noise and delays in complex neural models chiefly via computer simulations, our work obtains

a richer understanding by examining two simple noise modalities, the soft-threshold model and the LIF model, with simple finite transmission delays and deriving *analytical* expressions for the readout error $\sigma_{readout}$ as a function of noise level and delay, revealing a power-law dependence on delay for the optimal noise level σ^* and minimal readout error $\sigma^*_{readout}$.

For the soft-threshold model, we derived exact expressions (Eq 10) for $\sigma_{readout}$ as a function of the delay and the superthreshold probabilistic spiking rate ρ, which we reparameterized as λ, the mean number of spurious spikes during a spike propagation delay—equivalent to an inverse noise level. For a given delay, we recovered the characteristic U-shaped dependence of $\sigma_{readout}$ on λ (Fig 2). Minimizing our expression for $\sigma_{readout}$ with respect to λ, we found that the optimal λ^* grows with the delay as $(\delta/\tau)^{2/3}$ and the associated minimal $\sigma^*_{readout}$ grows with the delay as $(\delta/\tau)^{2/3}$ (Eqs 12 and 11).

For the LIF model, we characterized the dynamics of the membrane potentials as a Gaussian packet from an OU process impinging upon the threshold, and we derived an approximate upper-bound for $\sigma_{readout}$ as a function of small noise and delays. Again, we recovered the characteristic U-shaped dependence of $\sigma_{readout}$ on noise level (Fig 4). Minimizing our approximate upper-bound for $\sigma_{readout}$, we found that $\sigma^*_{readout}$ grows with delay as $(\delta/\tau)^{2/3}$ and the approximate optimal noise level σ^* grows as $(\delta/\tau)^{1/3}$ (Eqs 25 and 26). The behavior of $\sigma^*_{readout}$ matches that of the soft-threshold model, and the behavior of σ^* matches the behavior of λ^* in the soft-threshold model, when λ^* is converted to a noise level.

Our hope is that our analytical results help quantitatively elucidate the fundamental mechanisms underpinning the beneficial role of noise in the presence of delays and provide a foundation for further analysis of more complex neural models. Indeed, our analytical results shed light on previous observations in simulations of more complex models, and they naturally suggest future directions of investigation. For example, in a neural model with more biophysical details than we consider in our work, [11] observes that the optimal noise level increases weakly with population size N, but the mechanisms underlying this phenomenon are not discussed. Interestingly, we can inspect our expression for the mean number of spurious spikes λ (Eq 23), and note that it has a coefficient $c(N)$ that grows sub-logarithmically with population size N. Thus as population size N grows, with all else held constant, the mean number of spurious spikes λ increases weakly with N. To mitigate this increase, one expects a corresponding weak increase in the optimal noise level σ^* (which then widens the membrane potential packet to compensate for the increased coefficient $c(N)$ in the number of spurious spikes λ, Eq 23), just as [11] observes in simulations. Indeed, studying the parameter λ, the mean number of spurious spikes during a propagation delay, could yield fruitful insight in such computational studies.

Generality and future directions

More complex dynamics. We derive our results in the context of encoding a slowly-varying input, but our results may also apply to emulating a slowly-varying dynamical system. Indeed, the predictive coding framework [7] provides a formulation to map an arbitrary linear dynamical system to the connectivity of a network of spiking LIF neurons. To understand how our results relate to this more general task, we note that in the predictive coding framework, efficient coding is facilitated by a set of fast, instantaneous synapses, whereas the underlying linear dynamical system is facilitated by a set of slow synapses with finite-timescale dynamics. Importantly, when small axonal transmission delays are introduced, the fast synapses undergo a major change—they go from instantaneous to non-instantaneous. And in contrast, the slow synapses undergo a relatively minor change—they are now slightly delayed, in addition to already being slow. Thus the main contribution of delay to modifying the dynamics of such networks [7], should arise primarily from the effect of the delay on the fast, not the slow

synapses. It is precisely the effect of delay on fast synapses which we treat in our analysis here. Thus, our results could be used to describe the leading-order degradation due to delays in the more general scenario of emulating a linear dynamical system, with slowly-varying dynamics. We also note that the predictive coding framework [7] can be extended to emulate non-linear dynamical systems [27]; the non-linear dynamics are also facilitated by slow synaptic connections, and thus our approach may be extended to this setting as well, again by capturing the leading-order effect of delays on the fast synapses.

And while we consider encoding a positive, one-dimensional scalar signal, our results can also apply to encoding an arbitrary D-dimensional vector in $\mathbb{R}^D$. Importantly, when coding for a D-dimensional signal in the predictive coding framework [7], each neuron codes for some direction in $\mathbb{R}^D$. And when the coding error grows in any particular direction, the subpopulation of neurons tuned for that direction compete to spike and correct the error. Thus, our calculations may be adapted to approximately describe this scenario by replacing the parameter $N \to N_{eff}$, where N_{eff} is the size of the effective subpopulation actively participating in coding any particular direction.

However, for coding signed signals, we note that our analysis for positive-only coding directions ($w_i = 1, \forall i$) leaves out the possibility of a detrimental ping-pong effect. That is, if one of the neurons has a negative coding direction (if $w_i = -1$ for some i), a spike from a positively-coding neuron excites the negatively-coding neuron (and vice versa), which can initiate a volley of spurious synchronous spikes. Naturally, the ping-pong effect also arises when neurons have antipodal coding directions in $\mathbb{R}^D$. But importantly, the ping-pong effect can be mitigated by introducing an auxiliary coding dimension to eliminate antipodal neurons [7], increasing the firing threshold, or by simply removing the problematic excitatory connections that support the ping-pong effect [28], and then our analysis still holds.

And finally for completeness, while in our analysis we considered encoding the particular *constant* input signal $x(t) = 1$, we can see how our analysis can also apply to slowly *time-varying* signals by considering encoding an input signal $x(t) = a$, where a is an arbitrary $O(1)$ positive constant. Repeating the derivations presented in our analysis, but with the input $x(t) = a$, yields an approximate upper-bound on the readout error $\sigma_{readout}$, analogous to Eq 24, for when a slowly-varying signal has value a. For the LIF model, the current $I_i(t)$ (Eq 6) obtains an additional factor of a, and consequently the lower limit in the integral for the mean number of spurious spikes λ (Eq 22) becomes $T - a\frac{\delta}{\tau}$. Intuitively, a small input current (small a) results in fewer membrane potentials reaching threshold during the spike propagation delay, and thus fewer spurious spikes occur; and vice-versa for a large input current (large a). As a corollary, the optimal noise level increases monotonically with a, as additional noise is only beneficial insofar as it counteracts increased pathological synchronization. Thus, for example, one could use our analysis to calculate a single noise level that performs well overall for an arbitrary slowly-varying input signal by averaging across the signal's distribution, or one could introduce an adaptive mechanism that dynamically tunes the noise level to optimal, depending on the network's estimate $\hat{x}(t)$. However, importantly, we note that if one chooses to simply fix the noise level to the optimal noise level associated with a particular input value, say e.g., $a = a_{max}$, then our approximate upper bound for the readout error (Eq 24, with $x(t) = a_{max}$ used in preceding derivations) still upper-bounds the readout error for when the input signal is less than a_{max}—for values less than a_{max}, there is simply more noise than is necessary to optimally desynchronize the network. Hence, the optimal noise level associated with a_{max} facilitates the aforementioned readout error bound for any slowly-varying input signal bounded by a_{max}.

Other forms of heterogeneity. We studied the soft-threshold and membrane noise as specific mechanisms that can provide beneficial heterogeneity in spike-times which can

prevent pathological synchronization leading to excess spurious spikes. However, other mechanisms may provide such a beneficial heterogeneity as well. As a first example, consider again coding for D-dimensional signals. In high dimension $D \gg 1$, one can choose the neural coding directions in the predictive coding framework [7] such that few neurons are similarly-tuned. For example, if each neuron were allocated a random coding direction in $\mathbb{R}^D$, the typical cosine angle between the tuning, or coding directions, of any pair of neurons would be $O(1/\sqrt{D})$, while number of neurons could become exponential in D before the *maximal* cosine angle between the coding directions of any pair of neurons exceeds a given $O(1)$ threshold. With such a choice, the size N_{eff} of the effective subpopulation actively competing to correct the error in any particular direction becomes small, because few neurons are similarly tuned—there are fewer redundantly-coding neurons and thus a lesser propensity for spurious spikes, which improves coding fidelity. However, we note that details such as refractory period become important here, because the minimal subpopulation that codes nearest the direction of the error cannot spike continuously, thus other neurons with nearby coding directions must be recruited, expanding the active subpopulation. The beneficial effects of reduced redundancy for coding high-dimensional signals in the presence of delays has been observed in simulations [28], and extending our results to treat the details of particular choices of neural coding directions is an interesting future direction.

Many other sources of additional heterogeneity exist. For example, synaptic failures [1] provide a tunable source of noise—synapses are in general unreliable, but redundant synapses can be added to increase reliability, or different synapse morphologies can be used to achieve different levels of reliability. Indeed, synaptic failures have been observed to benefit coding in the presence of delays [11]. Modified spiking dynamics can also foster heterogeneity, such as the L2 penalty described in [7, 11, 12] where neurons self-reset themselves more strongly than they inhibit others, encouraging more diverse neural activity. Heterogeneous temporal filters have been shown to benefit efficient coding in spiking neural networks as well [29]. Imprecise connectivity, i.e., adding a frozen noise to the connectivity in Eq 5, can also provide a beneficial heterogeneity via chaotic fluctuations [30, 31] and has been observed to do so in the rate-version of the predictive coding framework [15]. Other biophysical details, such as refractory periods, distribution of transmission delays, distribution of synaptic dynamics, distribution of leak λ_V, and per-neuron membrane noise levels, could all serve to provide spike-timing heterogeneity in a manner that prevents excess synchronization and spurious spikes. Our work provides a foundation upon which these mechanisms can be further studied in an analytic framework. More generally, our work reveals a conceptual framework whereby spike-timing heterogeneity, originating from either single neuron noise, imprecise connections, network level chaos, or other sources, can endow spiking neural networks with superior computational capabilities in the presence of transmissions delays, by preventing the build up of pathological synchrony.

Supporting information

S1 Appendix. Supporting derivations and simulation details.
(PDF)

Acknowledgments

Some of the computing for this project was performed on the Sherlock cluster. We would like to thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results.

Author Contributions

Conceptualization: Jonathan Timcheck, Jonathan Kadmon, Kwabena Boahen, Surya Ganguli.

Investigation: Jonathan Timcheck, Jonathan Kadmon.

Methodology: Jonathan Timcheck, Jonathan Kadmon, Surya Ganguli.

Resources: Kwabena Boahen, Surya Ganguli.

Software: Jonathan Timcheck.

Supervision: Kwabena Boahen, Surya Ganguli.

Visualization: Jonathan Timcheck.

Writing – original draft: Jonathan Timcheck, Jonathan Kadmon, Kwabena Boahen, Surya Ganguli.

Writing – review & editing: Jonathan Timcheck, Jonathan Kadmon, Kwabena Boahen, Surya Ganguli.

References

1. Faisal AA, Selen LP, Wolpert DM. Noise in the nervous system. Nature reviews neuroscience. 2008; 9(4):292–303. https://doi.org/10.1038/nrn2258 PMID: 18319728
2. Greengard P. The neurobiology of slow synaptic transmission. Science. 2001; 294(5544):1024–1030. https://doi.org/10.1126/science.294.5544.1024 PMID: 11691979
3. Von Neumann J. Probabilistic logics and the synthesis of reliable organisms from unreliable components. Automata studies. 1956; 34:43–98.
4. Niven JE, Laughlin SB. Energy limitation as a selective pressure on the evolution of sensory systems. Journal of Experimental Biology. 2008; 211(11):1792–1804. https://doi.org/10.1242/jeb.017574 PMID: 18490395
5. Aiello LC, Wheeler P. The expensive-tissue hypothesis: the brain and the digestive system in human and primate evolution. Current anthropology. 1995; 36(2):199–221. https://doi.org/10.1086/204350
6. Barlow HB, et al. Possible principles underlying the transformation of sensory messages. Sensory communication. 1961; 1(01).
7. Boerlin M, Machens CK, Denève S. Predictive coding of dynamical variables in balanced spiking networks. PLoS computational biology. 2013; 9(11):e1003258. https://doi.org/10.1371/journal.pcbi.1003258 PMID: 24244113
8. Huang Y, Rao RP. Predictive coding. Wiley Interdisciplinary Reviews: Cognitive Science. 2011; 2 (5):580–593. PMID: 26302308
9. Abeles M. Corticonics: Neural circuits of the cerebral cortex. Cambridge University Press; 1991.
10. Schwemmer MA, Fairhall AL, Denéve S, Shea-Brown ET. Constructing Precisely Computing Networks with Biophysical Spiking Neurons. The Journal of neuroscience: the official journal of the Society for Neuroscience. 2015; 35(28):10112–10134. https://doi.org/10.1523/JNEUROSCI.4951-14.2015 PMID: 26180189
11. Chalk M, Gutkin B, Denève S. Neural oscillations as a signature of efficient coding in the presence of synaptic delays. eLife. 2016; 5. https://doi.org/10.7554/eLife.13824 PMID: 27383272
12. Rullán Buxó CE, Pillow JW. Poisson balanced spiking networks. PLOS Computational Biology. 2020; 16(11):e1008261. https://doi.org/10.1371/journal.pcbi.1008261 PMID: 33216741
13. Touboul JD. The hipster effect: When anti-conformists all look the same. Discrete & Continuous Dynamical Systems-Series B. 2019; 24(8).
14. McDonnell MD, Ward LM. The benefits of noise in neural systems: bridging theory and experiment. Nature Reviews Neuroscience. 2011; 12(7):415–425. https://doi.org/10.1038/nrn3061 PMID: 21685932
15. Kadmon J, Timcheck J, Ganguli S. Predictive coding in balanced neural networks with noise, chaos and delays. Advances in Neural Information Processing Systems. 2020; 33.

16. Attwell D, Laughlin SB. An energy budget for signaling in the grey matter of the brain. Journal of Cerebral Blood Flow & Metabolism. 2001; 21(10):1133–1145. https://doi.org/10.1097/00004647-200110000-00001 PMID: 11598490

17. Sarpeshkar R. Analog versus digital: extrapolating from electronics to neurobiology. Neural computation. 1998; 10(7):1601–1638. https://doi.org/10.1162/089976698300017052 PMID: 9744889

18. Boahen K. A neuromorph's prospectus. Computing in Science & Engineering. 2017; 19(2):14–28. https://doi.org/10.1109/MCSE.2017.33

19. Gollisch T, Meister M. Rapid neural coding in the retina with relative spike latencies. science. 2008; 319(5866):1108–1111. https://doi.org/10.1126/science.1149639 PMID: 18292344

20. Joris PX, Smith PH, Yin TC. Coincidence detection in the auditory system: 50 years after Jeffress. Neuron. 1998; 21(6):1235–1238. https://doi.org/10.1016/S0896-6273(00)80643-1 PMID: 9883717

21. VanRullen R, Guyonneau R, Thorpe SJ. Spike times make sense. Trends in neurosciences. 2005; 28(1):1–4. https://doi.org/10.1016/j.tins.2004.10.010 PMID: 15626490

22. Landau ID, Sompolinsky H. Coherent chaos in a recurrent neural network with structured connectivity. PLoS computational biology. 2018; 14(12):e1006309. https://doi.org/10.1371/journal.pcbi.1006309 PMID: 30543634

23. Gerstner W, Kistler WM, Naud R, Paninski L. Neuronal dynamics: From single neurons to networks and models of cognition. Cambridge University Press; 2014.

24. Paninski L, Pillow J, Lewi J. Statistical models for neural encoding, decoding, and optimal stimulus design. Progress in brain research. 2007; 165:493–507. https://doi.org/10.1016/S0079-6123(06)65031-0 PMID: 17925266

25. Uhlenbeck GE, Ornstein LS. On the theory of the Brownian motion. Physical review. 1930; 36(5):823. https://doi.org/10.1103/PhysRev.36.823

26. Tuckwell HC. Introduction to theoretical neurobiology: volume 2, nonlinear and stochastic theories. vol. 8. Cambridge University Press; 1988.

27. Alemi A, Machens CK, Deneve S, Slotine JJ. Learning nonlinear dynamics in efficient, balanced spiking networks using local plasticity rules. In: Thirty-Second AAAI Conference on Artificial Intelligence; 2018.

28. Calaim N, Dehmelt FA, Gonçalves PJ, Machens CK. Robust coding with spiking networks: a geometric perspective. bioRxiv. 2020;.

29. Zeldenrust F, Gutkin B, Denéve S. Efficient and robust coding in heterogeneous recurrent networks. PLoS computational biology. 2021; 17(4):e1008673. https://doi.org/10.1371/journal.pcbi.1008673 PMID: 33930016

30. Sompolinsky H, Crisanti A, Sommers HJ. Chaos in random neural networks. Physical review letters. 1988; 61(3):259. https://doi.org/10.1103/PhysRevLett.61.259 PMID: 10039285

31. Kadmon J, Sompolinsky H. Transition to chaos in random neuronal networks. Physical Review X. 2015; 5(4). https://doi.org/10.1103/PhysRevX.5.041030

S1 Appendix
Supporting derivations and simulation details

Contents

A Supporting derivations **1**
- A.1 Models . . . 1
- A.2 Readout error for the soft-threshold model . . . 4
  - A.2.1 Readout at a single point in time . . . 5
  - A.2.2 Mean readout error . . . 9
  - A.2.3 Additional variance from random interspersing of 2-spike events . . . 14
- A.3 Readout error for the LIF model with zero delay . . . 17
  - A.3.1 Readout at a single point in time . . . 20
  - A.3.2 Mean readout error . . . 24
- A.4 Readout error for the LIF model with delay . . . 28
  - A.4.1 Mean number of spurious spikes . . . 30
  - A.4.2 Mean readout error . . . 32

B Simulation details **36**
- B.1 Soft-threshold model . . . 37
- B.2 LIF model . . . 37
- B.3 Simulation code . . . 38

A Supporting derivations

A.1 Models

We consider a network of N leaky integrate-and-fire neurons whose membrane potentials undergo the dynamics

$$\tau \dot{V}_i(t) = -\lambda_V V_i(t) + N - \tau o_i(t) - \tau \sum_{j \neq i} o_j \left(t - \frac{\delta}{N} \right) + \sqrt{\tau}\sigma\eta_i(t), \text{ and} \tag{S1}$$

neuron i emits a spike when $V_i > \frac{1}{2}$.

V_i is the i^{th} neuron's membrane potential, " $\dot{}$ " indicates a time derivative, λ_V controls the strength of the leak, N is the input current, $o_j(t)$ is the j^{th} neuron's spike train (represented as a sum of Dirac δ functions), $\Delta = \frac{\delta}{N}$ is the spike propagation delay, σ controls the noise level, η_i are independent Wiener processes, and the firing threshold is $\frac{1}{2}$. Note that neurons self-reset instantaneously to $V_i = -\frac{1}{2}$. Note that the power of the time scale τ is chosen to ensure that each term in Eq. (S1) has the same units as membrane voltage, which here we take to be dimensionless. To see this note that a δ-function has units of inverse time (so multiplying any spike train $o_i(t)$, which is composed of δ-functions, by τ makes these terms dimensionless) and note that the Wiener process $\eta_i(t)$ obeys $\langle \eta_i(t)\eta_j(t')\rangle_t = \delta_{ij}\delta(t-t')$, implying that $\eta_i(t)$ itself has units of inverse square root of time. Thus we multiply the last term by $\sqrt{\tau}$ to ensure that the standard deviation σ has the same units as the dimensionless membrane voltage. Finally, note the leak term λ_V is also dimensionless.

Our goal is to derive an expression that describes the readout error, which we define as the standard deviation $\sigma_{readout}$ of the readout $\hat{x}(t)$, as a function of the number of neurons N, the delay δ, and the noise level σ.

$$\hat{x}(t) = \frac{1}{N}\sum_{i=1}^{N} r_i(t) \tag{S2}$$

$$\tau \dot{r}_i(t) = -r_i(t) + \tau o_i(t) \tag{S3}$$

$$\sigma_{readout} = \sqrt{\left\langle \left(\hat{x}(t) - \langle \hat{x}(t')\rangle_{t'}\right)^2 \right\rangle_t} \tag{S4}$$

We derive our central result—an approximate upper-bound for $\sigma_{readout}$ in the $\delta \ll \tau$ limit—in Subsection A.4. But before tackling the full complexity of this system, we study a simpler system that exhibits the same noise tradeoff in the presence of delay: a tight-balance network of soft-threshold neurons. For this soft-threshold dynamics, we derive an exact expression for $\sigma_{readout}$ in the large N, small delay $\delta \ll \tau$ limit. The neurons undergo the dynamics

$$\begin{aligned} &\tau \dot{V}_i(t) = N - \tau o_i(t) - \tau \sum_{j \neq i} o_j(t-\Delta), \text{ and} \\ &\text{neuron i emits spikes with probability rate } \rho(V_i) = \begin{cases} \rho, & V_i > \frac{1}{2} \\ 0, & V_i \leq \frac{1}{2} \end{cases}. \end{aligned} \tag{S5}$$

To understand the behavior of this model, consider a simple situation where the membrane voltages V_i of all the neurons are the same and equal to 0, as would be the case when there is no external input, and if there were an additional leak term to ensure all membrane potentials decay to 0. Then suppose the external input equal to N is turned on at time $t = 0$. All the membrane potentials will rise linearly at a rate N/τ and become superthreshold when they reach $V_i = \frac{1}{2}$ together at the same time $\tau/2N$. The entire population will remain superthreshold until the first neuron in the network spikes. Below, we will denote the time interval from the simultaneous superthreshold crossing

time to the time of first spike by the random variable t_{first}. At this time t_{first}, the membrane potential of the first neuron to spike will be immediately reset (i.e. decremented by 1). The rest of the network however, will first receive inhibition from this initial spike only at time $t_{\text{first}} + \Delta$, due to the transmission delay Δ. Of course during the entire time all membrane potentials are still increasing at a rate N/τ due to the external input. Thus, as long as the time $t_{\text{first}} + \Delta$ is less than τ/N, the entire population will be subthreshold again after responding to delayed inhibition from the first spike. More generally, we refer to the interval of time between the entire population crossing above threshold and then returning to subthreshold as a superthreshold interval. The smallest this superthreshold interval can be is the $t_{\text{first}} + \Delta$, assuming the delayed inhibition from a single spike is sufficient to return the entire population to subthreshold state. Of course it could be longer if delayed inhibition from multiple spikes is required to return the entire population to a subthreshold state. As we see below, we will be working with short delays Δ and high enough superthreshold firing rates ρ so that typically a single spike suffices.

To find interesting, high performing operating regimes of this model, we consider the statistics of spiking during a superthreshold interval. First, for any individual superthreshold neuron, the probability of emitting a spike in a short time interval of duration dt is $\rho \times dt$, and the time to the next spike of that neuron is exponentially distributed with mean $1/\rho$. Thus the mean inter-spike interval (ISI) of a single super-threshold neuron is $1/\rho$. However, if all N neurons cross threshold simultaneously, at some reference time $t = 0$, then the time t_{first} to the first spike in the entire network is exponentially distributed with mean $\frac{1}{N\rho}$. Now if any individual neuron spikes, it can potentially immediately return to a subthreshold state through its own reset mechanism that decrements the membrane voltage by 1. However we will be operating in a large N regime with a small number of neurons spiking between the first spike at t_{first} and the first network wide inhibition due to that spike at time $t_{\text{first}} + \Delta$. So for large N we can neglect the small number of neurons that may have returned to a subthreshold state due to their own spiking during this interval $[t_{\text{first}} \ldots t_{\text{first}} + \Delta]$, and simply assume all N neurons are superthreshold during this interval. Thus the superthreshold network ISI, defined to be the mean interval between any successive pair of spikes occurring anywhere in the network, is $\frac{1}{N\rho}$. So during any part of the superthreshold interval of duration equal to the delay Δ, the expected number of spikes to occur is a Poisson random variable with mean $\lambda = N\rho\Delta$. In order to ensure in the large N limit that a large number of spikes do not impact the readout before the first spike has a chance to inhibit the network, we therefore choose $\Delta = \delta/N$ where δ is $O(1)$. This ensures that $\lambda = \rho\delta$ is $O(1)$, and so $O(1)$ spikes occur between the time the first spike occurs at t_{first}, and the time that spike first has a chance to inhibit the network at time $t_{\text{first}} + \Delta$. Also, with this scaling of the delay Δ, the shortest possible superthreshold interval, on average is simply the sum of the mean of the time to first spike t_{first} (which is $\frac{1}{N\rho}$) plus the delay (which is $\Delta = \frac{\delta}{N}$). During this mean time, the membrane voltages, which are still integrating at a rate $\frac{N}{\tau}$, rise

above threshold by an amount $\frac{N}{\tau}(\frac{1}{N\rho}+\frac{\delta}{N}) = \frac{1}{\tau\rho}+\frac{\delta}{\tau}$. As long as this quantity is less than 1, then on average, delayed inhibition from a single spike is sufficient to end the superthreshold interval. This quantity will be less than 1 at small delays δ, and small mean time $\frac{1}{\rho}$ to first spike in a *single* neuron, all relative to the single neuron integration time τ.

In summary, if the population first becomes superthreshold at a reference time $t = 0$, then: (1) the mean of the time t_{first} to the first network spike is $\frac{1}{N\rho}$; (2) the mean time when the network first receives delayed inhibition from this first spike is $\frac{1}{N\rho}+\frac{\delta}{N}$; (3) the mean number of extra, or spurious, network spikes during this delay period is $\lambda = \rho\delta$; and (4) the final membrane voltage at the end of this period (right before the inhibition) is $\frac{1}{\tau\rho}+\frac{\delta}{\tau}$. We will focus our analysis on the following limits. First, in order to focus on regimes of good encoding performance with low error, we assume the mean excess number of spurious spikes $\lambda = \rho\delta \ll 1$. Second, for ease of analysis we focus on the regime in which once the network is superthreshold, a single spike can on average return the network to a subthreshold state, and moreover, the few neurons that do spike during the superthreshold interval and are reset, do not return to a superthrehsold state before the rest of the network becomes subthreshold. For the first condition, we require $\frac{1}{\tau\rho}+\frac{\delta}{\tau} = \frac{\delta}{\tau}(\frac{1}{\lambda}+1) \ll 1$ which implies $\frac{\delta}{\tau} \ll 1$ since we are already assuming $\lambda \ll 1$. Furthermore, consider a neuron that fired during the superthreshold state and was immediately reset to a subthreshold state corresponding to a membrane voltage that is below threshold by an $O(1)$ amount. Due to integration of the external stimulus at a rate $\frac{N}{\tau}$ it will return to a superthreshold state within a time $O(\frac{\tau}{N})$. As long as the delay $\Delta = \frac{\delta}{N}$ is much less than $\frac{\tau}{N}$, the entire population will become subthreshold before this reset neuron becomes superthreshold again. Thus, in the combined limit $\lambda \ll 1$ and $\delta \ll \tau$, the entire network operates in a simple fashion: all membrane potentials cross threshold together, a single first spike occurs, and then after a delay Δ the entire network becomes subthreshold again. Then after a time $O(\frac{\tau}{N})$ all the membrane potentials cross threshold simultaneously again. Interestingly as we see below, performance is actually best when the network behaves in this fashion.

A.2 Readout error for the soft-threshold model

First, we integrate Eq. (S3) to write the readout at a given time t as a sum of exponential kernels from all spikes in the past.

$$\hat{x}(t) = \frac{1}{N}\sum_{k=1}^{\infty} e^{-\frac{\Delta t_k}{\tau}}. \tag{S6}$$

Here, $\Delta t_k = t - t_k$ where t_k is the time of the k'th spike in the past from any neuron in the network. Note that the time between any successive pair of network spikes scales with N as $O(1/N)$. To see this, consider two possible cases. First consider the case where the entire population is superthreshold, and a spike occurs. After a delay of $\Delta = \delta/N$, this spike will inhibit the entire

network, causing the entire population to become subthreshold. However, before this happens, spikes can occur in any neuron at a rate $N\rho$ and therefore the typical ISI between any pair of spikes that occur in the same superthreshold interval is $O(\frac{1}{N\rho})$. Now consider the the second case: the final spike in one superthreshold interval and the first spike in the next superthreshold interval. The time interval between these two spikes decomposes into 3 parts: (1) the waiting time for network-wide inhibition to return to a subthreshold state; (2) the time it takes for the external stimulus to drive the population back to a superthreshold state; (3) the waiting time t_{first} for the first network spike to occur after becoming superthreshold. For (1), because we are working in a limit where a single spike can typically return the network to a subthreshold state, the waiting time for returning to a subthreshold state is at most the delay $\Delta = \frac{\delta}{N}$ from the last spike (and is shorter if there were earlier spikes in the same superthreshold event before the last spike). For (2), we note that the mean number of spikes in a superthreshold event is $1 + \lambda$ (the first spike plus the mean number λ of spurious spikes). Each of these spikes will eventually decrement the membrane voltage by an $O(1)$ amount. Because $\lambda \ll 1$, the membrane voltage of all neurons will never descend below threshold by more than an $O(1)$ amount, and because the membrane voltages integrate the external drive at a rate $\frac{N}{\tau}$, they will recover to a superthreshold state in a time $O(\frac{\tau}{N})$. Finally, for (3), the mean of the time t_{first} to the first network spike is $\frac{1}{N\rho}$. Thus the sum of these 3 intervals is $O(\frac{\delta}{N} + \frac{\tau}{N} + \frac{1}{N\rho})$, and is overall $O(\frac{1}{N})$.

A.2.1 Readout at a single point in time

In order to compute $\sigma_{readout}$, we need to compute the standard deviation in $\hat{x}(t)$ over a time window. To do so, we first compute the statistics of the readout at a particular point in time t_{thres}, defined as the time when the entire network reaches threshold together. We denote the readout at this time by $\xi := \hat{x}(t_{thres})$. To compute the mean and variance of ξ, we need to understand the statistics of the past times Δt_k of all previous spikes, as these times contribute to the readout through Eq. (S6). Figure A shows a schematic for the trajectory of the population of membrane potentials undergoing the dynamics Eq. (S5). Assuming the neural population starts with initial conditions $V_i = 0\ \forall i$, the population reaches threshold together (the first threshold-crossing shown in the schematic). Let z_k denote the duration of time between this threshold crossing and the first spike anywhere in the network. As discussed above z_k is an exponentially-distributed random variable with mean $\frac{1}{N\rho}$. Thereafter, there is a delay time $\Delta = \frac{\delta}{N}$ during which the other $N - 1 \approx N$ neurons remain superthreshold and continue to fire probabilistically. In a case where no extra spurious spikes occur, the rest of the membrane potentials rejoin the first-firing neuron's membrane potential after receiving the inhibitory spike. The population then continues toward threshold, and the process repeats. Occasionally (due to our assumption of small λ) one (or more) spurious spikes may occur. For example, in the third superthreshold interval in Figure A, two spikes occur at times z_2 and z_3 after the time the population crosses threshold. Each of these spikes will decrement the membrane

voltage of all neurons by 1, and the entire population will recover arriving at the next superthreshold crossing over a time proportional to the total number of spikes in the previous superthreshold interval.

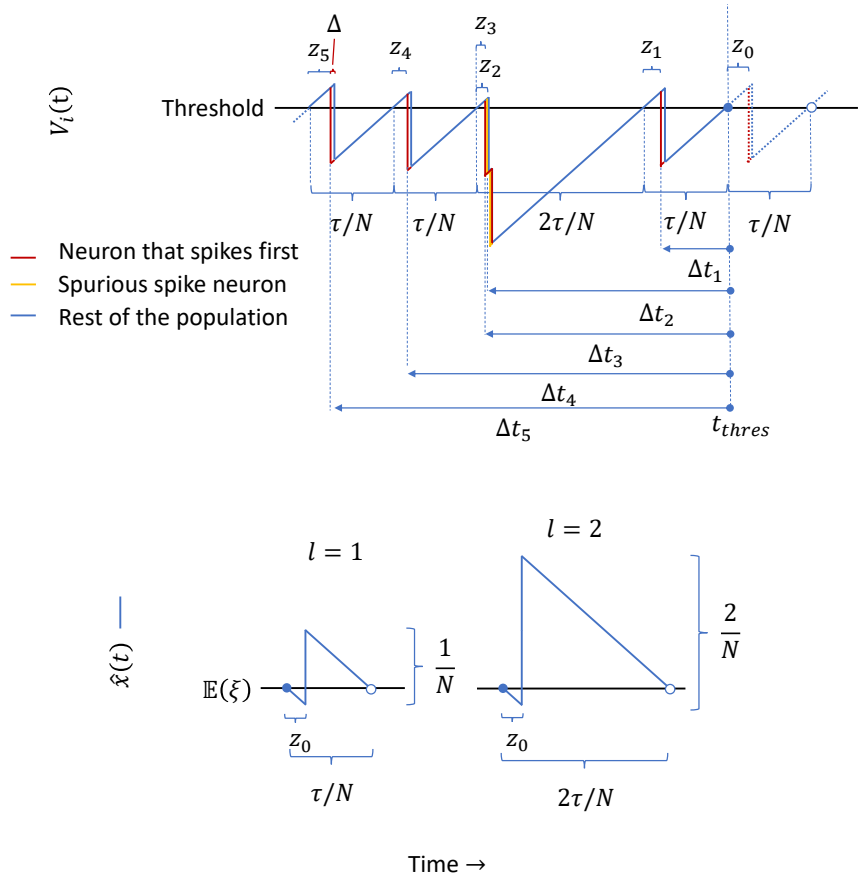

Figure A: Schematic for spike generation and readout error in the soft-threshold model. Top: the membrane potential dynamics Eq. (S5) generate a history of spike-times Δt_k relative to the moment the membrane potnetials reach threshold again, t_{thres}. The spike-times include variation z_k from the probabilistic first-spike wait-time due to the soft threshold, and the possibility of spurious synchronous spikes during the delay Δ. Bottom: integrating the readout deviation over l-spike events in the computation of $\sigma_{readout}$. $\mathbb{E}(\xi)$, the mean readout at the time t_{thresh}, is used as a reference point to compute the mean readout and the squared deviation from that mean, integrated over time.

For ease of exposition, let us first compute the mean and variance of ξ in the limit of extremely small λ. In this limit, the most probable scenario by far in each superthreshold interval, is that only a single first spike occurs, with no

subsequent extra spurious spikes. Later we will consider corrections due to extra spurious spikes that become relevant at larger λ. In this small λ limit, there is a one-to-one correspondence between individual spikes and the previous superthreshold crossing event of the population. We can use this correspondence to calculate the times $t_k = t_{thres} - \Delta t_k$ of the k spike in the past. First, let c_k be the time at which the population crossed threshold just before the time t_k of the k'th spike in the past. Thus by the definition of z_k we have $t_k = c_k + z_k$. Furthermore note that for any k the time between successive population superthreshold obeys $c_k - c_{k+1} = \frac{\tau}{N}$. To see this, note that by definition, at time c_{k+1} the membrane potentials are at threshold $V_i = \frac{1}{2}$. During the entire time from c_{k+1} to c_k, the membrane potentials are rising at a rate $\frac{N}{\tau}$, but they also suffer a decrement of 1 when the neuron that spiked resets its own membrane potential and when the rest of the neurons receive delayed inhibition from this spike. This combination of integration at a constant rate $\frac{N}{\tau}$, plus a decrement of 1, implies the next time the population crosses threshold will be at time $c_k = c_{k+1} + \frac{\tau}{N}$. This constancy of the time between population superthreshold crossings then implies that $c_k = t_{thres} - k\frac{\tau}{N}$ and so $t_k = c_k + z_k = t_{thres} - k\frac{\tau}{N} + z_k$, and thus $\Delta t_k = t_{thres} - t_k = k\frac{\tau}{N} - z_k$.

Inserting this formula for Δt_k into Eq. (S6), we obtain

$$\xi = \frac{1}{N}\sum_{k=1}^{\infty} e^{-\frac{k}{N}} e^{\frac{z_k}{\tau}}. \tag{S7}$$

We can see that ξ is an infinite sum of uncorrelated random variables. Using the fact that the variables $\frac{z_k}{\tau}$ are exponentially distributed with mean $\frac{1}{N\rho\tau}$, we have

$$\mathbb{E}(e^{\frac{z_k}{\tau}}) = \frac{N\rho\tau}{N\rho\tau - 1} \approx 1 + \frac{1}{N\rho\tau} \tag{S8}$$

$$var(e^{\frac{z_k}{\tau}}) = \frac{N\rho\tau}{(N\rho\tau - 1)^2(N\rho\tau - 2)} \approx \left(\frac{1}{N\rho\tau}\right)^2, \tag{S9}$$

where the approximate expressions denote the terms to leading orders in $\frac{1}{N}$. We can sum the geometric series, using the independence of the z_k, to obtain

$$\mathbb{E}(\xi) = \frac{1}{N}\sum_{k=1}^{\infty}(e^{-\frac{1}{N}})^k \frac{N\rho\tau}{N\rho\tau - 1} \tag{S10}$$

$$= \frac{1}{N}\frac{N\rho\tau}{N\rho\tau - 1}\frac{e^{-\frac{1}{N}}}{1 - e^{-\frac{1}{N}}} \tag{S11}$$

$$\approx 1 + \frac{(2 - \rho\tau)}{2\rho\tau}\frac{1}{N} + O\left(\frac{1}{N^2}\right) \tag{S12}$$

and

$$var(\xi) = \frac{1}{N^2}\sum_{k=1}^{\infty}(e^{-\frac{2}{N}})^k var(e^{\frac{z_k}{\tau}}) \tag{S13}$$

$$= \frac{1}{N^2} var(e^{\frac{z_k}{\tau}}) \frac{e^{-\frac{2}{N}}}{1-e^{-\frac{2}{N}}} \tag{S14}$$

$$= \frac{1}{2\rho^2\tau^2}\frac{1}{N^3} + O\left(\frac{1}{N^4}\right) \tag{S15}$$

We note some important properties of $\mathbb{E}(\xi)$ and $var(\xi)$. First, to leading order in $\frac{1}{N}$, $\mathbb{E}(\xi) = 1$, which is expected for the constant stimulus $x(t) = 1$. The next leading order term in $\mathbb{E}(\xi)$ decreases as $O(\frac{1}{N})$. Interestingly, in the limit of large ρ, the subleading term approaches $-\frac{1}{2N}$. This simple behavior can be understood as follows. In the large ρ limit, the network spikes very soon after it crosses superthreshold. This single spike increases the readout by exactly $1/N$. The readout subsequently decays at a rate τ but within a time $O(1/N)$ the membrane potentials recover to be subthreshold. By this time the readout has decayed to below 1 and is ready to be boosted to above 1 by the next spike. Thus, when we measure the readout specifically at the time the population has crossed threshold, before this next spike, the readout will tend to underestimate the input by an $O(1/N)$ amount, while right after the spike, the readout will overestimate the input by an $O(1/N)$ amount. In this fashion, the readout will zig-zag about the correct value. In fact it zig-zags symmetrically about the correct value (in the large ρ limit), and is thus $\frac{1}{2}\frac{1}{N}$ below 1 at threshold crossing and $\frac{1}{2}\frac{1}{N}$ above 1 just after the spike. In fact this zig-zag behavior of the readout cannot be reduced either by reducing delays (by reducing δ) or reducing variability in the time to first spike (by increasing ρ); it remains when $\delta = 0$ and ρ is large.

Further note that the standard deviation of the readout at the time the population crosses threshold (i.e. $\sqrt{var(\xi)}$ in Eq. (S15)) is $O(\frac{1}{N^{3/2}})$, and so is much smaller than the typical $O(1/N)$ amplitude of the zig-zag behavior of the readout about the correct value. Thus the variability in spike timing does not contribute appreciable variance to the readout ξ at the single time t_{thres}. This can be understood intuitively, as follows. First, the time to spike after a threshold crossing has variance $O(1/N^2)$ because N neurons are racing to spike first. However, even though the decoder only integrates over a time scale τ, the network generates spikes at a rate that is $O(N)$. Thus the decoder value ξ at time t_{thres} feels variability from the last $O(N)$ spikes in the past; the variabiilty in the timing of these spikes all contribute to the variance of the readout at t_{thres}. The combination of $O(N)$ spikes thus leads to an amplification of readout variance by a factor of N due to the addition law of variance of independent variables. But furthermore, the readout averages over N neurons leading to a multiplicative reduction of variance by $1/N^2$. Thus the total variance of the readout at any fixed time t_{thres} due to the combination of decoder averaging, decoder integration, and network racing to first spike is $O(\frac{1}{N^2}N\frac{1}{N^2}) = O(\frac{1}{N^3})$.

Note that we have shown that the standard deviation of ξ, or the readout at time t_{thres} is negligible and only $O(\frac{1}{N^{3/2}})$ in the small λ limit where only single spikes in any given superthreshold interval occur. Below, we will show that this conclusion still holds, even at larger λ when multiple spikes could occur with appreciable probability in a single superthreshold interval.

A.2.2 Mean readout error

So far, we have computed the mean and variance of the readout at a specific time t_{thres}. However, our fundamental goal is to compute the mean and variance of the readout across time. In doing so we will find that both the bias and the variance of the readout will be $O(1/N)$. To see this, consider integrating the readout across a long, contiguous time interval that starts at time $t = 0$ at the beginning of a superthreshold interval and that ends at the start of a much later superthreshold interval, at a time $t_{end} \gg \tau/N$. Then the mean readout is given by

$$\langle \hat{x}(t) \rangle_t := \frac{1}{t_{end}} \int_0^{t_{end}} \hat{x}(t) dt \tag{S16}$$

and with this definition, the bias of the readout is $\langle \hat{x}(t) \rangle_t - \langle x(t) \rangle_t = \langle \hat{x}(t) \rangle_t - 1$. The variance of the readout is given by

$$\sigma^2_{readout} := \frac{1}{t_{end}} \int_0^{t_{end}} (\hat{x}(t) - \langle \hat{x}(t) \rangle_t)^2 dt. \tag{S17}$$

Importantly, many superthreshold intervals occur in this long contiguous time interval from $t = 0$ to t_{end}. To see this, recall that each superthreshold interval contains $O(1)$ spikes because the mean number of spurious spikes $\lambda \ll 1$, and that a single spike inhibits the membrane potential population by 1. Thus after each superthreshold interval, it takes $O(1/N)$ time for the population to return to superthreshold because the membrane potentials are driven by an input current N/τ. For $t_{end} \gg \tau/N$ then, many superthreshold intervals occur in the time between $t = 0$ and t_{end}.

Now, to evaluate Eq. (S16) and Eq. (S17), we wish to express the trajectory of $\hat{x}(t)$ during and between these superthreshold intervals. Thus it is useful to introduce the notion of an "event", which we define to be the interval of time from the beginning of a superthreshold interval to the beginning of the subsequent superthreshold interval. By definition, each event contains a single superthreshold interval at its beginning. During this superthreshold interval, recall that a random number $l + 1$ spikes occur, where l is a Poisson random variable with mean λ. And after the superthreshold interval and before the end of the event, no further spikes occur since the population is subthreshold. Thus we can categorize events with the following nomenclature: events containing 1 spike are referred to as 1-spike events, events containing 2 spikes are referred to as 2-spike events, and so on. We can use this definition of event to simplify our primary task of integrating over the long time interval from $t = 0$ to t_{end} in Eq. (S16) and Eq. (S17) because the trajectory $\hat{x}(t)$ can be represented as

a sequence of events—one event starting after another, stitched together—and thus the integrals in Eq. (S16) and Eq. (S17) can be subdivided into a sum of many smaller integrals, with one integral for the duration of each event. Let M be the (large) number of events in the long time interval from $t = 0$ to t_{end}, then Eq. (S16) can be written as

$$\langle \hat{x}(t) \rangle_t = \frac{1}{t_{end}} \sum_{i=1}^{M} \mu_i \tag{S18}$$

where

$$\mu_i := \int_{t_i}^{t_{i+1}} \hat{x}(t) dt \tag{S19}$$

is the readout integrated from the i'th event's beginning, t_i, to its end, t_{i+1}. (The beginning of a subsequent event is the end of the preceding event, and the last event ends at time $t_{M+1} = t_{end}$.) Similarly, Eq. (S17) can be written as

$$\sigma^2_{readout} = \frac{1}{t_{end}} \sum_{i=1}^{M} s_i \tag{S20}$$

where

$$s_i := \int_{t_i}^{t_{i+1}} (\hat{x}(t) - \langle \hat{x}(t) \rangle_t)^2 dt. \tag{S21}$$

Now we must compute the sums of μ_i and s_i in Eq. (S18) and Eq. (S20), and to help us do so, we will take advantage of an important property of $\hat{x}(t)$ that we derived earlier. Namely, that the value of the readout at the time of threshold-crossing $\xi := \hat{x}(t_{thres})$ has $O(1/N^{3/2})$ fluctuations. Since we will soon see that the variance of the readout $\sigma_{readout}$ is $O(1/N)$, which is large compared to $O(1/N^{3/2})$, we can consider the value at the readout at threshold-crossing as effectively fixed at the value $\mathbb{E}(\xi)$. And since every event begins and ends at threshold-crossings, the readout at the beginning and end of each event is effectively fixed at the value $\mathbb{E}(\xi)$. Thus each μ_i and s_i do not depend on the history of $\hat{x}(t)$—the value of the readout at the endpoints of integration in Eq. (S19) and Eq. (S21) are always $\mathbb{E}(\xi)$ ($\hat{x}(t_i) = \mathbb{E}(\xi), \forall i$), and the spike times in an event, relative to the start time of the event, are independent of all other events, as the superthreshold interval in the event produces spikes with spike-time distribution independent of other superthreshold intervals. Thus the μ_i are independent random variables and the s_i are independent random variables. (μ_i is independent of μ_j and s_j $\forall j \neq i$, and s_i is independent of μ_j and s_j $\forall j \neq i$; and of course, μ_i is correlated with s_i $\forall i$, because they are from the same event.)

With this independence, we are free to perform the sums in Eq. (S18) and Eq. (S20) in any order. We take advantage of this to group the events by their spike number, which we will see makes a further simplification. We write

$$\langle \hat{x}(t) \rangle_t = \frac{1}{t_{end}} \sum_{l=0}^{\infty} \sum_{i=1}^{Mp(l)+O(\sqrt{M})} \mu_i(l) \tag{S22}$$

$$\sigma^2_{readout} = \frac{1}{t_{end}} \sum_{l=0}^{\infty} \sum_{i=1}^{Mp(l)+O(\sqrt{M})} s_i(l) \tag{S23}$$

where

$$p(l) := \frac{\lambda^l e^- \lambda}{l!} \tag{S24}$$

is a Poisson distribution with mean λ—the probability for an event to have $l+1$ spikes. Here, we have written that there are $Mp(l)+O(\sqrt{M})$ $(l+1)$-spike events for each $l = 0, 1, ...$ because the number of $l+1$ spike events is a random variable with mean $Mp(l)$ and a deviation of order $O(\sqrt{M})$, as M is large, and we have defined the random variables $\mu_i(l)$ and $s_i(l)$ to have the same distribution as μ_i and s_i, but conditioned on the number of spikes in the event i being $l+1$. (The $O(\sqrt{M})$ deviation can be intuitively understood as the standard deviation from a binomial distribution, where a successful trial is an event having $l+1$ spikes, and a failure as having any other number of spikes.) Importantly, each $\mu_i(l)$ and $s_i(l)$ maintain independence from the other events because the conditional distribution is simply nomenclature. Since the $\mu_i(l)$ are independent random variables, we recognize the sum $\sum_{i=1}^{Mp(l)+O(\sqrt{M})} \mu_i(l)$ as an expectation value $\langle \mu_i(l) \rangle_i$ times $Mp(l) + O(\sqrt{M})$, where the angle brackets $\langle \cdot \rangle_i$ denote average over the distribution of $\mu_i(l)$. A parallel argument holds for $s_i(l)$, and we can write

$$\langle \hat{x}(t) \rangle_t = \frac{1}{t_{end}} \sum_{l=0}^{\infty} \Big(Mp(l) + O(\sqrt{M}) \Big) \langle \mu_i(l) \rangle_i \tag{S25}$$

$$\sigma^2_{readout} = \frac{1}{t_{end}} \sum_{l=0}^{\infty} \Big(Mp(l) + O(\sqrt{M}) \Big) \langle s_i(l) \rangle_i. \tag{S26}$$

which distills our task down to calculating $\langle \mu_i(l) \rangle_i$, $\langle s_i(l) \rangle_i$, and t_{end}.

Since we had replaced the integral over the long time interval from $t = 0$ to t_{end} in Eq. (S16) and Eq. (S17) with a sum over M events, we also need to express the normalizing fraction $\frac{1}{t_{end}}$ in terms of M so that we can eventually cancel out M that appears in Eq. (S25) and Eq. (S26)—M should not appear in the final result because we introduced it simply to express the long time window over which we are averaging. Fortunately, it is easy to express t_{end} by recalling that there are $Mp(l) + O(\sqrt{M})$ $(l+1)$-spike events $\forall l$ in the long time interval from $t = 0$ to t_{end}, and that each $l+1$ spike event has duration $(l+1)\tau/N$—each spike inhibits the membrane potential population by 1, yielding a total inhibition of $l+1$, and the input current N/τ constantly drives the membrane potentials back to threshold, counteracting the inhibition, ending the event when threshold is reached again at a time $(l+1)\tau/N$ after the start of the event. Thus we can simply sum up the duration of all events, to express $t_{end} = \sum_{l=0}^{\infty} (Mp(l) + O(\sqrt{M}))(l+1)\tau/N$, and we can substitute this expression into Eq. (S25) and Eq. (S26) to yield

$$\langle \hat{x}(t) \rangle_t = \frac{\sum_{l=0}^{\infty} \Big(Mp(l) + O(\sqrt{M}) \Big) \langle \mu_i(l) \rangle_i}{\sum_{l=0}^{\infty} (Mp(l) + O(\sqrt{M}))(l+1)\tau/N} \tag{S27}$$

$$\sigma^2_{readout} = \frac{\sum_{l=0}^{\infty} \left(Mp(l) + O(\sqrt{M}) \right) \langle s_i(l) \rangle_i}{\sum_{l=0}^{\infty} (Mp(l) + O(\sqrt{M}))(l+1)\tau/N}. \tag{S28}$$

Now we can see that dividing both the numerator and denominator by M makes the M disappear from the expression: the terms with factor $Mp(l)$ become $Mp(l)/M = p(l)$, and the terms of order $O(\sqrt{M})$ become $O(1/\sqrt{M})$, which go to zero because M is large; thus dividing numerator and denominator by M yields

$$\langle \hat{x}(t) \rangle_t = \frac{\sum_{l=0}^{\infty} p(l) \langle \mu_i(l) \rangle_i}{\sum_{l=0}^{\infty} p(l)(l+1)\tau/N} \tag{S29}$$

$$\sigma^2_{readout} = \frac{\sum_{l=0}^{\infty} p(l) \langle s_i(l) \rangle_i}{\sum_{l=0}^{\infty} p(l)(l+1)\tau/N}. \tag{S30}$$

Finally, it remains to calculate the expectation values $\langle \mu_i(l) \rangle_i$ and $\langle s_i(l) \rangle_i$, which we recall are averages over the random variables Eq. (S19) and Eq. (S21), conditioned on there being $l+1$ spikes in the event in the time interval t_i to t_{i+1} in Eq. (S19) and Eq. (S21). Let us consider an instantiation of an $l+1$-spike event, compute $\mu_i(l)$ and $s_i(l)$ for this event, and then consider taking an average over instantiations of $l+1$-spike events to compute $\langle \mu_i(l) \rangle_i$ and $\langle s_i(l) \rangle_i$. Recall that for an instantiation of an $l+1$-spike event, the readout $\hat{x}(t)$ has value $\mathbb{E}(\xi)$ at the beginning and end of the event, that the time to first spike in the superthreshold interval at the beginning of the event is a exponentially-distributed random variable z_0 with mean $1/N\rho$ and standard deviation $1/N\rho$, and that l spikes occur during the delay Δ, after the first spike. We illustrate the readout's trajectory during this event in Figure A (bottom).

We highlight some important features of this trajectory of $\hat{x}(t)$ during the $l+1$ spike event. First, the readout $\hat{x}(t)$ is $\mathbb{E}(\xi) + O(1/N)$ during the entirety of the event, because each spike contributes $O(1/N)$ to the readout (Although we consider $l = 0, ..., \infty$ in Eq. (S27) and Eq. (S28), we recall that $\lambda \ll 1$, and thus $l > O(1)$ events are exponentially suppressed in $p(l)$, Eq. (S24).) And since the duration of the event is short, with duration $(l+1)\tau/N = O(1/N)$, compared to the decay of the readout, with time constant $\tau = O(1)$, the readout trajectory can be expressed as $(\mathbb{E}(\xi) + O(1/N))e^{-t'/\tau} = (\mathbb{E}(\xi) + O(1/N))(1 - t'/\tau + O(t'^2))$, where time t' is time since the start of the event (time t in $\hat{x}(t)$ is $t = t_i + t'$, where t_i is the start time of the event). Thus to first order in time, the readout decreases linearly during the event (with the exception of discontinuous increases due to spikes), with slope $-(\mathbb{E}(\xi) + O(1/N))/\tau$, which is $-1/\tau$ to leading order in N, because $\mathbb{E}(\xi)$ is 1 to leading order in N (Eq. (S12)). We do not consider higher-order corrections to this approximately linear decrease in the readout, because the constant slope $-1/\tau$ provides consistency with our assumption that the readout at the beginning and end of the event, $\hat{x}(t_i)$ and $\hat{x}(t_{i+1})$, are effectively fixed at the expectation value of ξ—the constant slope of $-1/\tau$ over the duration of the event $(l+1)\tau/N$ perfectly balances the contribution $(l+1)/N$ to the readout from the $(l+1)$ spikes, i.e. $0 = (l+1)/N - 1/\tau \times (l+1)\tau/N$, returning the readout to $\mathbb{E}(\xi)$ at the end of

the event—and the constant slope is sufficient to find $O(1/N)$ contributions to $\sigma_{readout}$ as we will see.

Second, the standard deviation of the first-spike time $t_i + z_0$ (where t_i is the start time of the event) is much larger than the variability of the l spurious spike-times that occur during the delay Δ; importantly, this fact will allow us to collapse the spike-times of the l spurious spikes to a single time for the purposes of calculating the most salient contributions to $\langle \hat{x}(t) \rangle_t$ and $\sigma_{readout}$, as a more finely-timed treatment would only provide higher-order corrections. To see why this fact is true, recall that we are working in the $\lambda = \delta\rho \ll 1$ limit. Dividing both sides of this inequality by $N\rho$, we arrive at $\delta/N \ll \frac{1}{N\rho}$, and substituting in $\Delta = \delta/N$, we have $\Delta \ll \frac{1}{N\rho}$. Recalling that the standard deviation of z_0 is $1/N\rho$, we see here that the standard deviation of z_0 is much larger than the delay Δ. Now, while the first spike time is exponentially distributed, for the spike-times of the l spurious spikes we must remember that the $l+1$ spike event is an event that is conditioned on having l spurious spikes occur during the delay Δ. (This assumes, as we did earlier, that we are working in a regime in which the first spike's inhibition is sufficient to return the population to subthreshold after the delay Δ, preventing the possibility of further spikes until the population reaches threshold again, which marks the end of the event.) Thus these l spikes are not exponentially distributed like the first spike—indeed, they must occur between time $t_i + z_0$ and $t_i + z_0 + \Delta$. Regardless of the spike-time distribution in this interval, the l spurious spikes have spike-times that are bounded in a time interval of width Δ, which is much smaller than the standard deviation of z_0. Thus the chief contribution to spike-time variability for the l spurious spikes comes from z_0, and we can collapse the l spikes to the same spike-time $t_i + z_0$, which we illustrate in Figure A (bottom).

Using these features together—the constant slope of the readout decay and the collapsing of the l spurious spike-times—we can write down $\mu_i(l)$ and $s_i(l)$ for an instantiation of an $l+1$ spike event:

$$\mu_i(l) = \int_0^{z_0} \left(\mathbb{E}(\xi) - \frac{t'}{\tau} \right) dt' + \int_{z_0}^{(l+1)\tau/N} \left(\mathbb{E}(\xi) + \frac{1}{N}(l+1) - \frac{t'}{\tau} \right) dt' \quad \text{(S31)}$$

$$s_i(l) = \int_0^{z_0} \left[\left(\mathbb{E}(\xi) - \frac{t'}{\tau} \right) - \langle \hat{x}(t) \rangle_t \right]^2 dt' + \int_{z_0}^{(l+1)\tau/N} \left[\left(\mathbb{E}(\xi) + \frac{1}{N}(l+1) - \frac{t'}{\tau} \right) - \langle \hat{x}(t) \rangle_t \right]^2 dt' \quad \text{(S32)}$$

where the first integral is from the beginning of the event to the time of the first spike, and the second integral is from the time of the first spike to the end of the event. We next want to average over instantiations i to obtain $\langle \mu_i(l) \rangle_i$ and $\langle s_i(l) \rangle_i$. Importantly, recall that z_0 is an exponentially distributed random variable with mean and standard deviation $1/N\rho$, so we must average over this distribution. Averaging over this distribution yields

$$\langle \mu_i(l) \rangle_i = (l+1)\frac{\tau}{N}\mathbb{E}(\xi) + \frac{1}{N^2}\frac{(l+1)(l\rho\tau + \rho\tau - 2)}{2\rho} \quad \text{(S33)}$$

$$\langle s_i(l)\rangle_i = \frac{1+l}{3N^3\rho^2\tau} \\ (6+\rho\tau(-3(1+l-2(\langle\hat{x}(t)\rangle_t - \mathbb{E}(\xi))n) + ((1+l)^2 - \\ 3(1+l)(\langle\hat{x}(t)\rangle_t - \mathbb{E}(\xi))N + 3(\langle\hat{x}(t)\rangle_t - \mathbb{E}(\xi))^2N^2)\rho\tau))) \tag{S34}$$

Now we can calculate the mean readout $\langle\hat{x}(t)\rangle_t$ by substituting the expression for $\langle\mu_i(l)\rangle_i$ (Eq. (S33)) into the expression for $\langle\hat{x}(t)\rangle_t$ (Eq. (S29)), which yields

$$\langle\hat{x}(t)\rangle_t = \mathbb{E}(\xi) + \frac{1}{N}\frac{\lambda^2\rho\tau + 3\lambda\rho\tau + \rho\tau - 2\lambda - 2}{2(\lambda+1)\rho\tau}. \tag{S35}$$

And then we can calculate the variance of the readout $\sigma^2_{readout}$ by substituting the mean readout (Eq. (S35)) into the expression for $\langle s_i(l)\rangle_i$ (Eq. (S34)), and substituting Eq. (S34) into the expression for $\sigma^2_{readout}$ (Eq. (S30)), which yields

$$\sigma^2_{readout} = \frac{1}{N^2}\left[\frac{(\delta/\tau)^2}{\lambda^2} + \frac{1+\lambda(14+\lambda(19+\lambda(10+\lambda)))}{12(1+\lambda)^2}\right] \tag{S36}$$

$$= \frac{1}{N^2}\left[\frac{1}{12} + \frac{(\delta/\tau)^2}{\lambda^2} + \lambda + O(\lambda^2)\right], \tag{S37}$$

where we have used $\lambda = \rho\delta$ to substitute ρ for $\frac{\lambda}{\delta}$.

We can find the optimal noise level and minimal $\sigma_{readout}$ by differentiating $\sigma_{readout}$ with respect to λ and setting the derivative equal to zero. This yields

$$\lambda^* = 2^{1/3}(\delta/\tau)^{2/3} \tag{S38}$$

$$\sigma^*_{readout} = \alpha\sqrt{\frac{1}{12} + \frac{3(\delta/\tau)^{2/3}}{2^{2/3}}} \tag{S39}$$

We corroborate these results with simulations (Figure 2).

A.2.3 Additional variance from random interspersing of 2-spike events

Importantly, we had assumed in our calculations for the mean readout $\langle\hat{x}(t)\rangle_t$ and the readout fluctuations $\sigma_{readout}$ that the variance of the readout at the time of threshold-crossings $\xi := \hat{x}(t_{thresh})$ is to leading order $var(\xi) = O(1/N^3)$; we used this fact to consider the value of the readout at the start and end of events as effectively fixed, compared to the leading order fluctuations in $\sigma_{readout}$, which were $O(1/N)$. However, we had only shown $var(\xi) = O(1/N^3)$ if one considers that only 1-spike events occur, and no $l+1$ spike events occur (where $l > 0$) (the derivation leading to Eq. (S15)). So it remains to confirm that, when there is an appreciable probability that $l+1$ spike events occur (where $l > 0$), the variance $var(\xi)$ is still $O(1/N^3)$.

To see why this is the case, we first imagine a readout trajectory $\hat{x}(t)$ that consists only of 1-spike events, of which we know $var(\xi) = O(1/N^3)$; then

we modify $\hat{x}(t)$ so that it includes l-spike events, calling this new trajectory $\hat{\tilde{x}}(t)$; and finally we calculate how much the variance $var(\tilde{\xi})$ of the readout at threshold-crossing $\tilde{\xi} := \hat{\tilde{x}}(t_{thres})$ changes under the perturbation $\hat{x}(t) \to \hat{\tilde{x}}(t)$, compared to the original variance $var(\xi)$. If the change $var(\tilde{\xi}) - var(\xi)$ is an $O(1/N^3)$ amount, then the $var(\tilde{\xi})$ is $O(1/N^3)$ as well, since we already know $var(\xi)$ is $O(1/N^3)$.

Recalling that the mean number of spurious spikes $\lambda \ll 1$, the most prevalent type of event is a 1-spike event (0 spurious spikes), and the second-most prevalent type of event is a 2-spike event (1 spurious spike). Thus we can consider the perturbation $\hat{x}(t) \to \hat{\tilde{x}}(t)$ to leading order in λ by only considering the addition of 2-spike events. Now let us consider what adding 2-spike events looks like. The readout trajectory $\hat{x}(t)$ is a sequence of 1-spike events, while the readout trajectory $\hat{\tilde{x}}(t)$ is a sequence of 1-spike events and 2-spike events, with 1-spike events accounting for the fraction $p(0) = 1 - \lambda + O(\lambda^2)$ of all events and 2-spike events accounting for the fraction $p(1) = \lambda + O(\lambda)^2$ of all events. Importantly, the few 2-spike events are randomly interspersed among the many 1-spike events, as we recall that the number of spikes in a given event is independent of all other events. Thus, we can imagine adding 2-spike events to $\hat{x}(t)$ by randomly replacing a fraction λ of (randomly selected) 1-spike events with 2-spike events. Now importantly, 2-spike events have duration $2\tau/N$, while 1-spike events have duration τ/N, which means replacements must shift other events (events before the replacement to earlier times and/or events after the replacement to later times) in order to create room for the longer 2-spike event. Thus instead of considering this complicated dependency of the other event times due to each single replacement, let us instead consider replacing subsequent pairs of 1-spike events with 2-spike events; a subsequent pair of 1-spike events has duration $2\tau/N$, which matches that of a 2-spike event, and thus the other event times do not change when such a pair replacement is made.

To understand how many such replacements are necessary to result in a fraction λ of 2-spike events in $\hat{\tilde{x}}(t)$, consider starting with a large number M of events in a long time interval of $\hat{x}(t)$ as we did earlier. We wish to perform replacements of 1-spike events in $\hat{x}(t)$ so that the fraction of 2-spike events after replacements $\hat{\tilde{x}}(t)$ is λ. The question is—how many 1-spike pair replacements should we make? Let us consider replacing K 1-spike event pairs. The number of 1-spike events is initially M (there are only 1-spike events in $\hat{x}(t)$). Replacing K 1-spike pairs removes $2K$ 1-spike events and introduces K 2-spike events. Thus the number of 1-spike events becomes $M - 2K$ and the number of 2-spike events becomes K. We wish for the fraction of 2-spike events $\frac{K}{M-2K}$ to be λ. Solving for K, we find $K = M\lambda + O(\lambda^2)$. This means that we must replace a fraction $K/M = \lambda + O(\lambda^2)$ pairs of 1-spike events to make the fraction of 2-spike events to be λ.

Now that we know the fraction of 1-spike event pairs to replace with 2-spike events to represent the perturbation $\hat{x}(t) \to \hat{\tilde{x}}(t)$, we need to consider what the effect such a perturbation has on $var(\tilde{\xi}) - var(\xi)$. First, let us consider the replacement of one pair of 1-spike events, whose end is at a time $t_{replace}$,

which is $n\tau/N$ before the time t_{thresh} ($t_{replace} = t_{thresh} - n\tau/N$). Recall that the replacement only changes the spike times of the 2 spikes in the pair of 1-spike events, and the spike times in all other events remain the same. The perturbation in the readout at t_{thresh}, $\tilde{\xi} - \xi$, is

$$d(n) := \frac{1}{N}((e^{-(n+1)/N} + e^{-n/N}) - 2e^{-(n+1)/N}), \tag{S40}$$

where we have neglected any differences due to first-spike times because they are much smaller than the change due to the replacement. (The first-spike times in the original 1-spike events each have standard deviation $1/N\rho$, which is much smaller than the $O(\tau/N)$ jump in spike-time due to replacing the pair of 1-spike events with a 2-spike event. As discussed earlier, we are considering the regime in which a single spike's inhibition is sufficient to return the entire population to subthreshold, which means $1/N\rho \ll \tau/N$, and thus all spikes happen near the beginning of an event—the second spike here, which is originally in the second 1-spike event, now takes place near the beginning of the 2-spike event, being pushed back $O(\tau/N)$.)

Importantly, individual replacements are probabilistic, happening with probability λ for each pair of 1-spike events. Thus, now that we know the difference $\tilde{\xi} - \xi$ due to a replacement, we can use it to express how a single *probabilistic replacement* effects $var(\tilde{\xi}) - var(\xi)$. Consider any pair of 1-spike events in $\hat{x}(t)$ (indexed by n in their end time $t_{replace} = t_{thresh} - n\tau/N$); the probability that this n'th pair is replaced by a 2-spike event is λ. Thus the change in mean readout at t_{thresh}, $\mathbb{E}(\tilde{\xi}) - \mathbb{E}(\xi)$, due to this single probabilistic replacement is

$$\langle d(n) \rangle := d(n)\lambda + 0(1-\lambda). \tag{S41}$$

where the change $d(n)$ due to the potential replacement is weighted by its probability λ. Furthermore, the contribution to the variance in the readout at t_{thresh}, $var(\tilde{\xi}) - var(\xi)$, due to this single probabilistic replacement is

$$var(d(n)) := (d(n) - \langle d(n) \rangle)^2\lambda + (0 - \langle d(n) \rangle)^2(1-\lambda). \tag{S42}$$

where we have used the fact that the probabilistic replacement is independent of other sources of variance.

Finally, we must sum over all probabilistic replacements to express the total effect of 2-spike events on $var(\tilde{\xi}) - var(\xi)$. We can sum the additional variance from each probabilistic replacement $var(d(n))$, where each pair of 1-spike events considered for replacement is indexed by n in their end time $t_{replace} = t_{thresh} - n\tau/N$:

$$var(\tilde{\xi}) - var(\xi) = \sum_{n=1}^{\infty} var(d(n)) \tag{S43}$$

$$= -\frac{1}{N^2} \frac{e^{-2\alpha}(e^{1/N} - 1)(\lambda - 1)\lambda}{1 + e^{1/N}} \tag{S44}$$

$$= -\frac{1}{2}((\lambda - 1)\lambda)\frac{1}{N^3} + O(\frac{1}{N^4}). \tag{S45}$$

where we have safely neglected the low-probability event that two pairs of 1-spike events that share a 1-spike event (subsequent pairs of 1-spike events, overlapping in time) are both replaced, which has probability $O(\lambda^2)$. We can see that the leading-order term in Eq. (S45) is $O(1/N^3)$, which completes the argument that $var(\tilde{\xi}) = O(1/N^3)$ because we had already shown that the variance considering 1-spike events only, $var(\xi)$, is $O(1/N^3)$.

A.3 Readout error for the LIF model with zero delay

Our ultimate goal is to study the standard deviation $\sigma_{readout}$ as a function of the delay Δ and noise σ in the membrane potential dynamics Eq. (S1); as mentioned in Models, our study will uncover an upper-bound for $\sigma_{readout}$ as a function of delay Δ and noise σ. But for ease of exposition, let us first study $\sigma_{readout}$ in the case of zero delay $\Delta = 0$, with noise $\sigma > 0$. In this case, the membrane potentials undergo the dynamics

$$\begin{aligned} &\tau \dot{V}_i(t) = -\lambda_V V_i(t) + N - \tau \sum_{j=1}^{N} o_j(t) + \sqrt{\tau}\sigma\eta_i(t), \text{ and} \\ &\text{neuron } i \text{ emits a spike when } V_i > \frac{1}{2}. \end{aligned} \tag{S46}$$

Studying this delay $\Delta = 0$ case, we will uncover properties of the membrane potential dynamics Eq. (S46) that will serve as a baseline for studying the dynamics when there is a delay $\Delta > 0$ (the dynamics in Eq. (S1))—we study $\Delta > 0$ in the next section.

To understand the behavior of the population of membrane potentials V_i in Eq. (S46), consider for the purpose of pedagogy the situation in which the membrane potentials start with some initial condition $V_i(0)$, and inhibitory spike terms are disabled ($o_j(t)$ terms are removed, and the neurons never spike, even when they surpass threshold). The dynamics become

$$\tau \dot{V}_i(t) = -\lambda_V V_i(t) + N + \sqrt{\tau}\sigma\eta_i(t). \tag{S47}$$

These dynamics correspond to an OU process [1, 2]—each membrane potential $V_i(t)$ undergoes an independent OU process with time constant τ, leak constant λ_V, drift term N, and independent noise $\sqrt{\tau}\sigma\eta_i(t)$. In particular if the process starts at a well specified initial condition $V_i(0)$, then its mean $\mathbb{E}(V_i(t))$ evolves in time as

$$\mathbb{E}(V_i(t)) = V_i(0)e^{-\frac{\lambda_V}{\tau}t} + \frac{N}{\lambda_V}\left(1 - e^{-\frac{\lambda_V}{\tau}t}\right). \tag{S48}$$

Notably, the initial condition $V_i(0)$ is exponentially forgotten on a time scale of $\frac{\tau}{\lambda_V}$ and the external drive N forces the mean to saturate at the value N/λ_V, again over a time scale of $\frac{\tau}{\lambda_V}$. Importantly, however, the temporal autocovariance of the stochastic process $V_i(t)$, is independent of the drift term:

$$cov(V_i(s), V_i(t)) := \langle (V_i(s) - \mathbb{E}(V_i(s)))\,(V_i(t) - \mathbb{E}(V_i(t)))\rangle \tag{S49}$$

$$= \frac{\sigma^2}{2\lambda_V}\left(e^{-\frac{\lambda_V}{\tau}|t-s|} - e^{-\frac{\lambda_V}{\tau}(t+s)}\right). \tag{S50}$$

where s and t are two times and the angle brackets $\langle\cdot\rangle$ denote an average over the stochastic trajectory η_i. The second term $e^{-\frac{\lambda_V}{\tau}(t+s)}$ reflects the non-stationarity of the autocovariance due to the definite initial condition at time 0, and for large times $s, t \gg \tau$ this non-stationary term vanishes. Let us consider large times, as we are interested in the setting in which the network is continuously operating. And for a time $s = t$, the first term $e^{-\frac{\lambda_V}{\tau}|t-s|}$ is 1, showing us that for a particular large time $t \gg \tau$, the variance of $V_i(t)$ is

$$var(V_i(t)) := \langle (V_i(t) - \mathbb{E}(V_i(t)))^2 \rangle \tag{S51}$$

$$= \frac{\sigma^2}{2\lambda_V} =: \sigma^2_{OU}. \tag{S52}$$

Thus, for an instant in time t, each $V_i(t)$ is a random variable with mean $\mathbb{E}(V_i(t))$ and variance $var(V_i(t)) = \frac{\sigma^2}{2\lambda_V}$, which we have also named σ^2_{OU} for a more brief notation in future calculations. Furthermore, it is known that $V_i(t)$ is Gaussian-distributed (a property of the OU process). Thus a simple picture emerges here—for any moment in time $t \gg \tau$, the membrane potentials are Gaussian-distributed with variance $\frac{\sigma^2}{2\lambda_V}$, forming what we call a "packet" that is centered at $\mathbb{E}(V_i(t)) = N/\lambda_V$. Of course, the individual membrane potentials fluctuate as time goes on, but the packet remains the same size, with variance $\sigma^2_{OU} = \frac{\sigma^2}{2\lambda_V}$.

Next, let us consider reintroducing the inhibitory terms $o_j(t)$, and describe the dynamics Eq. (S46). Compared to our preceding description of the dynamics without inhibitory terms, we see one difference: the drift term N is now counteracted by the inhibitory spike term $-\tau \sum_{j=1}^{N} o_j(t)$. The question is, what effect does this have on the evolution of the packet we previously illustrated? To answer this question, let us highlight one important property of the autocovariance equation describing the packet, Eq. (S50). Namely, that this equation does not depend on the initial condition $V_i(0)$. Then imagine the following: the packet of membrane potentials evolves according to our previous description, until one neuron reaches threshold. This neuron fires a spike, and, importantly, this spike is instantaneously delivered to all other neurons through the $-\tau \sum_{j=1}^{N} o_j(t)$ term (as the delay $\Delta = 0$), which simultaneously decrements the entire membrane potential packet. Since the decrement is simultaneous, the relative positions of the membrane potentials in the packet are preserved over that instant in time (but of course, the mean membrane potential is decremented). Thus the development of the autocovariance of the packet is preserved (it continues to grow or saturate to a steady state as in Eq. (S50)) because we can consider the new position of the packet after the spike to simply be a new set of initial conditions, and the autocovariance does not depend on the initial conditions. Of course, the mean membrane potential oscillates in a zig-zag fashion, driving upward due to the drift term N and being decremented with each spike, but the packet eventually grows to the same steady-state variance of $\sigma^2_{OU} = \frac{\sigma^2}{\lambda_V}$.

Thus to summarize the dynamics of Eq. (S46), we have a Gaussian packet of membrane potentials $V_i(t)$ rising toward threshold at a rate N/τ (driven

by the input current N) and discouraged from deviating from zero due to the leak term $-\lambda_V V_i(t)$. Since we are interested in an operating regime where the network dynamics are driven by tight balance (the input current N term), let us assume the leak term $-\lambda_V V_i(t)$ is small compared to the input current N, i.e. $|\lambda_V V_i(t)| \ll N$; thus the input current N surely drives the packet to threshold. When the top neuron in the packet hits threshold at a time we call t_{thresh}, it fires a spike that instantaneously decrements the entire population of membrane potentials by 1 (the $-\tau \sum_{j=1}^{N} o_j(t)$ term integrates to 1 because of the time constant τ multiplying $\dot{V}_i(t)$ in Eq. (S46)). Then, the membrane potential packet continues traveling toward threshold, and the process repeats. Note that in this repeated process, the same neuron does not necessarily fire successively—the membrane potentials continuously fluctuate within the packet, and thus a different neuron can become the top neuron in the membrane potential packet at any time.

Importantly, we wish to consider high-performing networks, and it is helpful to identify what regime of parameters facilitates high performance, i.e. small readout standard deviation $\sigma_{readout}$. First, let us consider the noise level σ. The width of the membrane potential packet σ_{OU} is controlled by σ through $\sigma_{OU} = \sigma^2/2\lambda_V$. Thus for small σ, the packet width is small, and for larger σ, the packet width is large. Now, to understand how the width of the packet affects the readout standard deviation $\sigma_{readout}$, let us consider first small σ, such that $\sigma_{OU} \ll 1$. In this case, the fluctuation in the membrane potential of the top neuron in the packet is small because the packet itself is tight. Thus the time between spikes Δt from when the top neuron, say, neuron i, spikes at a time $t_{thresh,1}$, and when the top neuron, neuron j (neuron j is not necessarily the same neuron as neuron i), subsequently spikes again at time $t_{thresh,2}$ ($\Delta t := t_{thresh,1} - t_{thresh,2}$) has very little variation. To see this, recall that the inhibition from the first spike at time $t_{thresh,1}$ inhibits the membrane potential packet by 1. The packet then travels toward threshold due to the input current N (recall that we can ignore the leak term $-\lambda_V V_i(t) \ll N$). Now, if the packet is infinitely tight, i.e. $\sigma_{OU} = 0$, then it takes a time $\Delta t = \tau/N$ for the population to reach threshold again at time $t_{thresh,2}$, with no fluctuations in the time Δt. (a point packet is travelling toward threshold with constant current N and repeatedly inhibited) This results in perfectly regular interspike intervals of τ/N, which correspond to an the ideal zig-zag readout as discussed in the large ρ limit of the soft-threshold model—the best performance achievable, given that the readout is defined as a sum of firing rates that are filtered spike-trains Eq. (S2). Next, consider a small $\sigma > 0$, in which $\sigma_{OU} > 0$. Now the fluctuations in the membrane potential of the top neuron can be significant because the membrane potentials fluctuate within a packet of finite width; thus variations are introduced in Δt. These variations are a departure from the perfectly regular spike-times of the $\sigma = 0$ case, thus they increase $\sigma_{readout}$. Thus we find here that high-performing networks correspond to low levels of noise σ.

Importantly, the magnitude of the fluctuations in the interspike interval time Δt is controlled by the standard deviation of the membrane potential

packet $\sigma_{OU} = \sigma/2\sqrt{\lambda_V}$. Thus another way to achieve consistent interspike interval times, aside from small σ, is to choose large λ_V because this will make σ_{OU} small as well. But we cannot adjust λ_V arbitrarily, as we recall that we would like to consider values of λ_V that correspond to the regime in which the network dynamics are dominated by tight balance, i.e., $|\lambda_V V_i(t)| \ll N$. Now, importantly, we are considering $\sigma_{OU} \ll 1$, and the membrane potentials $V_i(t)$ exhibit $O(1)$ fluctuations due to decrements from inhibitory spikes when the top neuron reaches threshold—thus the combined effects of the packet's width $\sigma_{OU} \ll 1$ and the inhibitory spikes create an overall membrane potential dynamics that exist in an $O(1)$ dynamic range. Hence, the leak term $\lambda_V V_i(t)$ indeed satisfies $|\lambda_V V_i(t)| \ll N$, so long as λ_V scales less than $O(N)$. Let us assume that λ_V is a given property of the network dynamics that satisfies this constraint.

Equipped with an understanding of the membrane potential dynamics Eq. (S46) and a qualitative understanding of how noise σ affects the dynamics, let us work toward our goal of quantitatively expressing the readout $\hat{x}(t)$'s standard deviation $\sigma_{readout}$ as a function of the noise σ for a network with some given leak parameter λ_V. We start by recalling that the readout $\hat{x}(t)$ is a sum of decaying exponentials, with each term corresponding to a spike-time in the network:

$$\hat{x}(t) = \frac{1}{N} \sum_{k=1}^{\infty} e^{-\frac{\Delta t_k}{\tau}}. \tag{S53}$$

Here, we have written $\Delta t_k := t - t_k$ where t_k is the time of the k'th spike in the past from any neuron in the network. For ease of exposition, it is useful to define the notion of an "event" here as the interval of time between two subsequent spikes. Thus the time interval from t_2 to t_1 is an event, the time interval from t_3 to t_2 is an event, and so on. (Note that this is the same definition of "event" as in our description of the soft-threshold model, because spike-times are equal to membrane potential threshold-crossing times in the LIF model; also note that we only have 1-spike events here, as there are no delays.)

A.3.1 Readout at a single point in time

Now, we eventually want to compute the integral of $\hat{x}(t)$ over time to determine $\sigma_{readout}$ (Eq. (S17)), but to begin let us consider the readout $\hat{x}(t)$ at just one point in time, $t_{mid} := t_1 + \frac{\tau}{2N}$, that is near the middle of the event that starts at time t_1. Let us call the value of the readout at this time $\xi := \hat{x}(t_{mid})$. To see why this time t_{mid} is near the middle of the event, consider the typical duration of an event—the membrane potential packet has been inhibited by 1 from the spike at the beginning of the event, and the driving current N takes a time τ/N to push the packet back to threshold, if we ignore fluctuations in the membrane potential packet. Thus, $t_{mid} = t_1 + \tau/2N$ is about halfway through the duration of the event.

Now importantly, the spike times t_k are random variables that depend on the fluctuations of the membrane potentials due to the noise term $\sqrt{\tau}\sigma\eta_i(t)$ in

Eq. (S46), and so ξ is also a random variable that depends on the the spike times Δt_k through Eq. (S53). Thus we need to study the spike times Δt_k. To simplify our analysis, let us first consider small λ_V, where $\lambda_V \ll 1$. To understand how small λ_V simplifies the dynamics of the membrane potentials, first consider Eq. (S50). Notice that λ_V sets the time-scale of autocorrelations in membrane potential trajectory $V_i(t)$ to τ/λ_V, as seen in the first term of Eq. (S50). Thus a small λ_V corresponds to a long mixing time τ/λ_V of the membrane potential packet. In essence on time scales much less than that of τ/λ_V, the relative order of membrane potentials will be preserved. Note also that the time it takes for the mean of the packet to travel to threshold right after experiencing inhibition is τ/N. Thus if the mixing time τ/λ_V is much greater than the time to rise to threshold τ/N, from event to event the identity of the top-neuron that first spikes will largely be preserved for successive events. We can see this effect in spike raster plots from simulations of the membrane potential dynamics Eq. (S46) (Figure 3(a)). Notably, as λ_V is decreased, the same neuron repeatedly spikes; i.e., the same neuron remains the top neuron in the packet over many events. Thus for small $\lambda_V \ll N$, we can approximate the spike-time generation dynamics by only considering the dynamics of a single neuron's membrane potential—that of the top neuron.

Considering just the single membrane potential of the top neuron evolving according to Eq. (S46), we can easily calculate the statistics of event durations, i.e., the time between the last spike of the top neuron and its next spike. This will in turn allow us to calculate the statistics of Δt_k in Eq. (S53). Let us consider an event that starts at time t_s and ends at time t_f, and let us call the membrane potential of the top neuron $V_\alpha(t)$ (α is the index of the top neuron). At the instant of t_s, the membrane potential $V_\alpha(t)$ has just reached threshold, the top neuron spikes, and the membrane potential $V_\alpha(t)$ is decremented by 1 by the inhibitory spike. Next, the membrane potential $V_\alpha(t)$ is driving back toward threshold by the input current N, and experiences two other effects during its journey: the leak $-\lambda_V V\alpha(t)$ and the noise $\sigma\eta_\alpha(t)$. Since λ_V is small and $V_\alpha(t)$ is $O(1)$, we can neglect the leak term. But the noise term $\sqrt{\tau}\sigma\eta_\alpha(t)$ is accumulated in the integration of Eq. (S46) while the membrane potential $V_\alpha(t)$ travels toward threshold, and the total time $t_{fp} := t_f - t_s$ it takes for the top neuron to reach threshold is a random variable. Fortunately, we can recognize t_{fp} as the first-passage time of a particle undergoing Brownian motion with drift N, noise level σ, time constant τ, and with a goal that is a distance 1 away. The moments of t_{fp} are known [2]. The mean is

$$\langle t_{fp} \rangle = \frac{\tau}{N} \tag{S54}$$

and the variance is

$$var(t_{fp}) := \left\langle (t_{fp} - \langle t_{fp} \rangle)^2 \right\rangle = \frac{\sigma^2 \tau^2}{N^3}. \tag{S55}$$

Furthermore, the duration of the next event is simply another instance of the random variable t_{fp}, as the history of previous events is not recorded in any

way in the state of the top neuron. Thus, defining the random variable t_{fp}^n as the time between spike time t_{n+1} and spike time t_n, i.e., $t_{fp}^n := t_n - t_{n+1}$, we can write the spike times Δt_k in Eq. (S53) a time $t = t_{mid}$ as

$$\Delta t_k = t_{mid} - t_1 + \sum_{n=1}^{k-1} t_{fp}^n \tag{S56}$$

$$= \tau/2N + \sum_{n=1}^{k-1} t_{fp}^n, \tag{S57}$$

where by definition $t_{mid} - t_1 = \tau/2N$.

Now that the Δt_k are expressed as sums of random variables t_{fp}^n, we can consider evaluating the readout at time t_{mid}, $\xi = \hat{x}(t_{mid})$. We rewrite Eq. (S53) as

$$\xi = \frac{1}{N} \sum_{k=1}^{\infty} e^{-\left(\frac{\tau}{2N} + \sum_{n=1}^{k-1} t_{fp}^n\right)/\tau}. \tag{S58}$$

Here we can recognize that ξ is a sum of correlated random variables, because the exponent of the k'th term is a sum of the first $k-1$ t_{fp}^ns; for instance, the $k = 2$ term contains t_{fp}^1, and all $k > 2$ terms also contain t_{fp}^1. Furthermore, the first-passage times t_{fp}^n are not Gaussian-distributed [2]. Thus evaluating this sum in non-trivial, in contrast to our analysis for the soft-threshold model. However, to make evaluating this sum more tractable, we note one important property: in the vast majority of $e^{-(\dots)}$ terms, the sum $\sum_{n=1}^{k-1} t_{fp}^n$ itself contains many terms. (For only the first few $e^{-(\dots)}$ terms ($k = 1, 2, 3, ...$), does the sum $\sum_{n=1}^{k-1} t_{fp}^n$ have few terms.) Thus using the central limit theorem, we can approximate the sum $\sum_{n=1}^{k-1} t_{fp}^n$ as a Gaussian distribution. Furthermore, this Gaussian distribution only depends on the mean and variance of the t_{fp}^n. Of course, even with this simplification, we still need to take into account that the $e^{-(\dots)}$ terms are correlated.

To formalize our central limit theorem approximation, let us rewrite Eq. (S58) as

$$\xi = \frac{1}{N} \sum_{k=1}^{\infty} e^{-\left(\frac{\tau}{2N} + \sum_{n=1}^{k-1} \tau/N + \gamma_n\right)/\tau}, \tag{S59}$$

where

$$\gamma_n \sim \mathcal{N}(0, \text{std} = \frac{\sigma\tau}{N^{3/2}}) \tag{S60}$$

is a zero-mean Gaussian-distributed random variable with variance equal to that of t_{fp}^n. (We pulled out the mean of t_{fp}^n to create the τ/N term in *Eq.*(S59)).

Now recall that we are ultimately interested in computing the variance of the readout $\sigma_{readout}^2$, and so as a starting point, let us compute the variance of the readout at the particular time t_{mid}, $\sigma_\xi^2 := \langle(\xi - \langle\xi\rangle)^2\rangle$. We begin calculating σ_ξ^2 by making some simplifications. First, we pull out the common factor for all terms $e^{-\tau/2N}$, and we substitute in the simple relation $\sum_{n=1}^{k-1} \tau/N = (k-1)\tau/N$

to obtain

$$\xi = \frac{1}{N} e^{-1/2N} \sum_{k=1}^{\infty} e^{-((k-1)\tau/N + \sum_{n=1}^{k-1} \gamma_n)/\tau}. \tag{S61}$$

Second, we note that the sum $h(k) := \sum_{n=1}^{k-1} \gamma_n$ has zero terms for $k = 1$, i.e., $h(1) = \sum_{n=1}^{0} \gamma_n = 0$, one term for $k = 2$, i.e., $h(2) = \sum_{n=1}^{1} \gamma_n = \gamma_1$, and so on. Thus there is a mismatch between the term-number k and the number of γ_n in the sum for term k. We would like to remove this mismatch to make the expression simpler, so we therefore pull out the first $k = 1$ term in Eq. (S61), and re-choose the index $k \to k - 1$ so that it now starts at $k = 1$ with the term that contains one γ_n (that is, γ_1). This results in an equivalent expression to Eq. (S61),

$$\xi = \frac{1}{N} e^{-1/2N} + \frac{1}{N} e^{-1/2N} \sum_{k=1}^{\infty} e^{-(k\tau/N + \sum_{n=1}^{k} \gamma_n)/\tau}. \tag{S62}$$

Now, we can see that the first term in Eq. (S62) is constant and not a random variable, so the variance σ_ξ^2 does not depend on it, thus we can neglect it in our calculation of σ_ξ^2. Thus our goal is to compute the variance due to the other terms in Eq. (S62). Importantly, the $e^{-1/2N}$ that we factored from all terms is simply a multiplicative factor, thus if we define

$$\zeta := \sum_{k=1}^{\infty} e^{-\left(k\tau/N + \sum_{n=1}^{k} \gamma_n\right)/\tau} \tag{S63}$$

then the variance of ξ is the variance of ζ times $(e^{-1/2N})^2$, i.e., $\sigma_\xi^2 = (\frac{1}{N} e^{-1/2N})^2 \sigma_\zeta^2$. Keeping this conversion from σ_ζ^2 to σ_ξ^2 in mind, we can turn our focus to computing σ_ζ^2.

Looking at Eq. (S63), we need to understand how we will compute the variance of a sum of correlated random variables. We note that we can break down the k'th term into a factor $e^{-k/N}$ multiplying the random variable $e^{-\sum_{n=1}^{k} \gamma_n/\tau}$. And we can see that the random variable $e^{-\sum_{n=1}^{k} \gamma_n \tau}$ is log-normal distributed, because the γ_n are Gaussian-distributed and so is their sum. Thus we see that we have a sum of weighted, correlated log-normal random variables. Importantly, the variance of the sum depends on the variance of the individual correlated random variables (and how they are correlated), thus it is useful to introduce here a definition for the variance σ_0^2 of the log-normal random variable $e^{-\gamma_n/\tau}$:

$$\sigma_0^2 := e^{\sigma^2/N^3}(e^{\sigma^2/N^3} - 1). \tag{S64}$$

Now, we can actually compute the variance of σ_ζ^2 by adding variances and splitting according to the contributions from each γ_n to account for correlations. To do this, we introduce a simple approximation that is accurate for small $\sigma \ll 1$; namely, that the variance of the random variable $e^{-\sum_{n=1}^{k} \gamma_n/\tau}$ is approximately

equal to the variance of the random variable $\sum_{n=1}^{k} e^{-\gamma_n/\tau}$ for all k. Thus, using this approximation as we consider the contribution to the variance from each term in ζ, we encounter a sum of now uncorrelated random variables, with coefficients determined by the term number k. Namely, the first few terms when considering the variance of ζ are $e^{-1/N}e^{-\gamma_1/\tau}$, $e^{-2/N}(e^{-\gamma_1/\tau} + e^{-\gamma_2/\tau})$, and so on. Thus we can sum the coefficients for each independent random variable (each γ_n), and add the coefficient sums in quadrature to yield the variance σ_ζ^2. We write

$$\sigma_\zeta^2 = \sigma_0^2 \left[\left(\sum_{k=1}^{\infty} r^k \right)^2 + \left(\sum_{k=2}^{\infty} r^k \right)^2 + ... \right] \tag{S65}$$

where we have introduced the definition

$$r := e^{-1/N} \tag{S66}$$

for brevity.

Next, to simplify Eq. (S65), we can recognize that the sums $\sum_{k=j}^{\infty} r^k$ are geometric series, whose totals are $\frac{r^j}{1-r}$ and write

$$\sigma_\zeta^2 = \sigma_0^2 \left[\left(\frac{r}{1-r} \right)^2 + \left(\frac{r^2}{1-r} \right)^2 + \left(\frac{r^3}{1-r} \right)^2 + ... \right]. \tag{S67}$$

And factoring out $\frac{r^2}{(1-r)^2}$, we see that the result Eq. (S67) is itself a geometric series, with constant ratio r^2, which we can substitute for its total $\frac{1}{1-r^2}$, writing

$$\sigma_\zeta^2 = \frac{\sigma_0^2 r^2}{(1-r)^2} \left[1 + r^2 + r^4 + ... \right] \tag{S68}$$

$$= \frac{\sigma_0^2 r^2}{(1-r)^2} \frac{1}{1-r^2}. \tag{S69}$$

Thus finally we can substitute back in the definitions for r and σ_0^2 into Eq. (S69), and recall the conversion $\sigma_\xi^2 = (\frac{1}{N} e^{-1/2N})^2 \sigma_\zeta^2$ to obtain

$$\sigma_\xi^2 = \left(\frac{1}{N} e^{\frac{1}{2N}} \right)^2 \frac{e^{-2/N}}{(1-e^{-1/N})^2} \frac{1}{1-e^{-2/N}} e^{\sigma^2/N^3} (e^{\sigma^2/N^3} - 1) \tag{S70}$$

$$\sigma_\xi^2 = \frac{\sigma^2}{2} \frac{1}{N^2} + O\left(\frac{1}{N^3} \right). \tag{S71}$$

A.3.2 Mean readout error

To summarize the result Eq. (S71), it tells us that the value of the readout at a time $\tau/2N$ into an event, $\hat{x}(t_{mid})$, is a random variable with standard deviation $\sigma/\sqrt{2}N$ to leading order. This is a useful fact, because it gives us a starting point for calculating $\sigma_{readout}$, as we now illustrate. Recall from our discussion of

the soft-threshold model, that we can express the trajectory of $\hat{x}(t)$ as a series of events (here events go from spike to subsequent spike, while in the soft-threshold model events went from superthreshold crossing to superthreshold crossing), and we can express the integrals over time that define the mean readout $\langle \hat{x}(t) \rangle$ and the variance of the readout $\sigma^2_{readout}$ as sums of of smaller integrals over the duration of individual events (Eq. (S18), Eq. (S19), Eq. (S20), and Eq. (S21)). But importantly, in contrast to the soft-threshold model, the value of the readout $\hat{x}(t)$ during each event is not independent of all other events; in fact, for the LIF model, our calculation for Eq. (S71) shows spike-time variability accumulated over many spikes (the $\sum_{n=1}^{k} \gamma_n$) creates fluctuations in ξ that are $O(1/N)$. Indeed, individual spike-time variability (the standard deviation of each γ_n) is only $\sigma\tau/N^{3/2} = O(1/N^{3/2})$, which is much smaller than the $O(1/N)$ spike-time variability in the soft-threshold model (the standard deviation of the first spike time) that was required to produce fluctuations $\sigma_{readout} = O(1/N)$ in the soft-threshold model where spike-time variability does not accumulate. Recognizing this important property that the readout $\hat{x}(t)$ is correlated from event to event, we can still however use our result for σ_ξ (Eq. (S71)) to compute $\sigma_{readout}$ because we need to take a sum over many events, and addition is commutative. The sum for $\sigma_{readout}$ (Eq. (S20)) can be executed in any order—thus all we need to know is the distribution of the integrated squared deviation s_i for an individual event.

Thus, we can perform a parallel computation for $\sigma_{readout}$ to that in the soft-threshold model (Equation Eq. (S18) to Eq. (S37)), except here we only have 1-spike events. Eq. (S29) becomes

$$\langle \hat{x}(t) \rangle_t = \frac{\langle \mu_i \rangle_i}{\tau/N}, \tag{S72}$$

and Eq. (S30) becomes

$$\sigma^2_{readout} = \frac{\langle s_i \rangle_i}{\tau/N}, \tag{S73}$$

where we recall that the denominator in these equations expresses the mean duration of an event, which for LIF model we recall is simply $\langle t_{fp} \rangle = \tau/N$. To express the expectation values $\langle \mu_i \rangle_i$ and $\langle s_i \rangle_i$, we can first consider a particular instance of an event, integrate over the duration of this event to compute μ_i and s_i, and then take the average over i to yield $\langle \mu_i \rangle_i$ and $\langle s_i \rangle_i$. For a particular event then, we can write

$$\mu_i = \int_0^{t^i_{fp}} \left(\xi_i + \frac{1}{2N} - \frac{1}{\tau} t' \right) dt' \tag{S74}$$

$$s_i = \int_0^{t^i_{fp}} \left[\left(\xi_i + \frac{1}{2N} - \frac{1}{\tau} t' \right) - \langle \hat{x}(t) \rangle_t \right]^2 dt' \tag{S75}$$

where ξ_i is an instance of the random variable ξ, and the time t' starts at the beginning of the event with $t' = 0$. The expression in () in Eq. (S74) and

Eq. (S75) is the readout $\hat{x}(t')$. Importantly, $\hat{x}(t')$ starts with value $\xi_i + 1/2N$ at $t' = 0$, and decays approximately linearly with slope $-1/\tau$ (by the same reasoning for the constant slope of $-1/\tau$ in the soft-threshold model). Thus at time $t' = \tau/2N$, the readout equals ξ_i, as the definition of ξ_i requires. Finally, the readout continues to decay until the event ends at time $t' = t^i_{fp}$.

To evaluate $\langle \mu_i \rangle_i$, let us represent the deviation of ξ_i from $\mathbb{E}(\xi)$ with the unit Gaussian random variable y_i multiplied by σ_ξ. Eq. (S74) becomes

$$\mu_i = \int_0^{t^i_{fp}} \left(\mathbb{E}(\xi) + \sigma_\xi y_i + \frac{1}{2N} - \frac{1}{\tau} t' \right) dt' \tag{S76}$$

and its expectation value (averaging over y_i and t^i_{fp}) is

$$\langle \mu_i \rangle_i = \mathbb{E}(\xi) + O(1/N^{3/2}) \tag{S77}$$

because y_i has zero mean, the integral from $t' = 0$ to the mean t^i_{fp} at $t' = \tau/N$ evaluates to exactly $\mathbb{E}(\xi)\tau/N$, and any additional contributions from the deviations of t^i_{fp} from its mean τ/N are order $O(1/N^{3/2})$ (Eq. (S55)). Thus the mean readout Eq. (S73) to leading order is

$$\langle \hat{x}(t) \rangle_t = \mathbb{E}(\xi). \tag{S78}$$

This result can be understood intuitively by comparing to the nominal operation of the predictive coding framework (the large ρ limit discussed in the soft-threshold model). Intuitively, for high-performing networks (low noise), the network produces perfectly regular spikes, and thus the decaying readout $\hat{x}(t) \approx 1$ by tracing a zig-zag around the encoded variable $x(t) = 1$. Since each spike contributes $1/N$ to the readout, the value of the readout immediately before each spike is $1 - 1/2N$, so that when the spike arrives, the readout is $1 + 1/2N$. This centers the zig-zag perfect around $\hat{x}(t) \approx 1$, and notably, since the time between spikes is short ($O(1/N)$), the readout decays linearly to 1 at the mid-point time between subsequent spikes. Hence, $\langle \hat{x}(t) \rangle_t = \mathbb{E}(\xi)$ for high-performing networks.

Next, to evaluate $\langle s_i \rangle_i$, let us again introduce the random variable y_i and substitute into s_i Eq. (S78) to obtain

$$s_i = \int_0^{t^i_{fp}} \left[\left(\mathbb{E}(\xi) + \sigma_\xi y_i + \frac{1}{2} - \frac{1}{\tau} t' \right) - \mathbb{E}(\xi) \right]^2 dt' \tag{S79}$$

$$= \int_0^{t^i_{fp}} \left[\sigma_\xi y_i + \frac{1}{2N} - \frac{1}{\tau} t' \right]^2 dt'. \tag{S80}$$

Performing this integral and taking the expectation value over i (averaging over y_i and t^i_{fp}), we obtain to leading order

$$\langle s_i \rangle_i = \frac{\sigma^2 \tau}{N^3} \left(\frac{1}{12} + \frac{\sigma^2}{2} \right) \tag{S81}$$

where we have again neglected any higher-order contributions from the deviations of t^i_{fp} from its mean, $\langle t_{fp} \rangle = \tau/N$. Substituting the leading-order term in this expression into Eq. (S73) and taking a square root, we finally obtain

$$\sigma_{readout} = \frac{1}{N}\sqrt{\frac{1}{12} + \frac{\sigma^2}{2}}. \tag{S82}$$

Now importantly, we recall that we made two important assumptions to arrive at this result. The first was that $\lambda_V \ll 1$, and the second was that the sum of spike-time variations $\sum_{n=1}^{k} \gamma_n$ could be well-approximated as Gaussian with the central limit theorem and that any deviations from this approximation are small because only a few sums $\sum_{n=1}^{k} \gamma_n$ have a small number of terms (small k, while in the calculation, k ranges from 1 to ∞). Thus for small λ_V, we evaluate our Gaussian approximation by comparing our result Eq. (S82) against Monte Carlo simulations that only contain the top neuron (Figure B). Secondly, to understand how our result relates to networks with larger λ_V—when the top neuron alone is not sufficient to describe the network's dynamics—we perform simulations with a variety of values of λ_V. Interestingly, we find empirically that Eq. (S82) provides an upper-bound for $\sigma_{readout}$ (Figure 3c). This upper-bound is an important result that will be used in the next section.

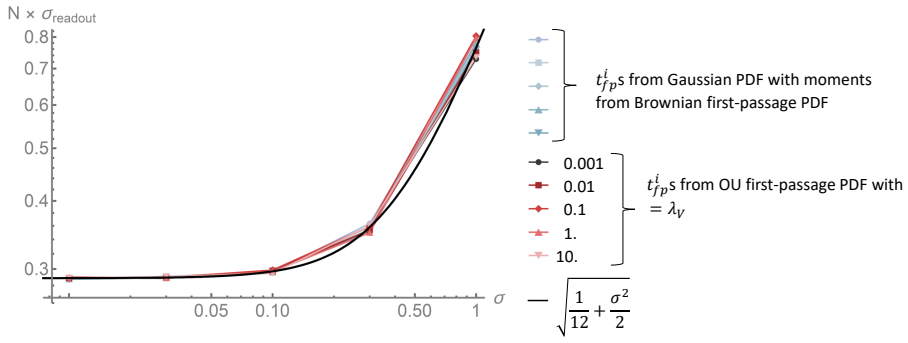

Figure B: The Gaussian approximation Eq. (S59) to the first-passage time distribution provides a good approximation for $\sigma_{readout}$ when compared to Monte Carlo simulations that contain only the top neuron. The standard deviation, $\sigma_{readout}$, of the readout (Eq. (S4)) when the t^i_{fp}s are drawn from the first-passage-time distribution of an OU process (red lines of different shades for different λ_V), appear to match $\sigma_{readout}$ when the t^i_{fp}s are drawn from a Gaussian distribution with moments from the first-passage-time distribution of Brownian motion (light blue lines, redundant simulations; black line, Eq. (S82)). These single-neuron simulations are equivalent to assuming the top neuron in the membrane potential packet remains the top neuron forever, like in the $\lambda_V \ll 1$ limit in Figure 3a.

A.4 Readout error for the LIF model with delay

In the previous section, we studied the dynamics Eq. (S1) with no delay (delay $\Delta = 0$) to determine the readout standard deviation $\sigma_{readout}$ as a function of the noise level σ. This provides a good starting point to determine $\sigma_{readout}$ as a function of both σ and nonzero delay $\Delta > 0$. To begin, let us consider the regime of parameters we would like to consider; in particular, we are interested in studying networks that exhibit high performance, i.e. small $\sigma_{readout}$, and networks that are driven by tight-balance, i.e., the leak term $-\lambda_V V_i(t)$ in Eq. (S1) is small compared to the driving current N. As discussed in the previous section, this regime corresponds to noise level $\sigma \ll 1$, and we consider λ_V as a given network property that satisfies the constraint that the leak term is small. Thus it remains to consider what values of delay Δ correspond to high performance. Intuitively, from our analysis of the soft-threshold model, we understand that small delays correspond to high performance—small delays allow one neuron to quickly inhibit the others, and this helps prevent spurious spikes that would otherwise increase readout error. Therefore, we consider small delays Δ for the LIF model. And more precisely, for ease of analysis, we will consider delays that are much smaller than the typical time between subsequent network spikes. The typical time between subsequent network spikes is $\langle t_{fp} \rangle = \tau/N$ (see previous section), and so we will consider delays $\Delta = \delta/N$, where $\delta \ll \tau$.

To understand how the membrane potentials evolve in the presence of delay $\Delta > 0$, let us first recall our description of the dynamics when $\Delta = 0$ from the previous section. The membrane potentials travel in a Gaussian packet of width $\sigma_{OU} = \sigma/\sqrt{2}\lambda_V$ toward threshold, driven by the input current N. When the top neuron in the packet hits threshold it fires a spike, and this spike then propagates to the other neurons, decrementing their membrane potentials, and thus prevents the other neurons from firing (until the packet reaches threshold again). Now this is where we see a departure from these dynamics due to the propagation delay Δ in Eq. (S1). The top neuron indeed fires a spike and resets itself when it reaches threshold (through the self-reset $-\tau o_i(t)$ term), but the spike takes a time Δ to decrement the other membrane potentials (through the $-\tau \sum_{j \neq i} o_j(t - \Delta)$ term). Thus the other neurons in the packet continue to travel toward threshold while they are waiting to receive the top neuron's inhibitory spike, and they may themselves cross threshold and fire spurious spikes. We will see, just as we saw in the soft-threshold model, that spurious spikes increase the readout error $\sigma_{readout}$, and thus one of our goals in this section will be to calculate the number of spurious spikes and the effect they have on the readout.

Now as we continue to describe the dynamics with delay $\Delta > 0$ and eventually calculate quantities like the mean number of spurious spikes, we will see as in the previous section, that defining the notion of an "event" is helpful. Thus we define an event here as the interval of time from when a top neuron in the membrane potential packet reaches threshold, at a time we refer to as $t_{thresh,1}$, to the next time a top neuron reaches threshold again (this latter top

neuron need not be the same as the previous top neuron), at a time we refer to as time $t_{thresh,2}$. Importantly, we define neurons that spike within the time interval $t_{thresh,1}$ to $t_{thresh,1} + \Delta$ as spuriously spiking neurons, and do not consider these neurons as candidates for marking the end of the event at $t_{thresh,2}$. Instead, the neuron that next spikes after $t_{thresh,1} + \Delta$ is denoted as the next top neuron to reach threshold, marking the end of the event when it reaches threshold at time $t_{thresh,2}$. To understand why this notion of event is well-defined, recall that we are focusing on the limit of small noise and small delays. In particular, note that the inhibition from one or more spikes (the first spike from the top neuron at $t_{thresh,1}$ and any additional spurious spikes that may occur) decrements the membrane potentials by 1 or more, and the membrane potentials are driven toward threshold by the current N. Thus it takes a mean time of $O(\tau/N)$ for the next spike to occur after $t_{thresh,1}+\Delta$ (the time when the entire population is first inhibited due to delayed inhibition from the first spike from the top neuron), with deviations from this mean time being small because we are considering the limit of small noise. And since we are considering delays $\delta \ll \tau$, which is equivalent to $\Delta \ll \tau/N$, we see that the time interval during which spurious spikes can occur is much smaller than the $O(\tau/N)$ time for the next nonspurious spike to occur. Thus there is a clear separation between the spike times of spurious spikes (between $t_{thresh,1}$ and $t_{thresh,1}+\Delta$) and the spike of the next top neuron $t_{thresh,2} = t_{thresh,1} + O(\tau/N)$.

Importantly, examining the interval from $t_{thresh,1}$ to $t_{thresh,1} + O(\Delta)$ more closely, we see another departure from the zero-delay dynamics described in the previous section when we consider that in any particular event, the top neuron is decremented at time $t_{thresh,1}$, while the other $N-1$ neurons are decremented at time $t_{thresh,1}+\Delta$. (And if spurious spikes occur in the event, further differences in inhibition times in the population are of $O(\Delta)$.) Interestingly, this creates a "head start" toward threshold for the top neuron in the packet, relative to the other neurons, as the membrane potentials travel toward threshold until the next top neuron spikes at time $t_{thresh,2}$. To see this, recall that the top neuron's membrane potential is reset to $-1/2$ when it self-resets at time $t_{thresh,1}$. At this point in time, it is receiving an extra small, excitatory current from the leak term $-\lambda_V V_i$ because V_α (where α is the index of the top neuron) is negative—this is in addition to the input current N. Meanwhile, the other membrane potentials are still traveling toward threshold, and the membrane potentials near the top of the packet are near the threshold at $1/2$. Thus these neurons experience a small, additional inhibitory current from the leak term $-\lambda_V V_i(t)$, slightly slowing their travel toward threshold. So we see that after all membrane potentials have received the inhibitory spike after delay Δ, the top neuron has integrated extra current relative to the other neurons near the top of the packet, and so it has a head start on its way toward threshold. This corresponds to a stretching of the upper tail of the membrane potential packet from the nominal Gaussian distribution of width σ_{OU}. We will return to this subtle but important departure from a Gaussian packet when the nature of the packet's tail becomes important in our calculations.

A.4.1 Mean number of spurious spikes

Equipped with our understanding of the dynamics Eq. (S1) and the definition of an event, let us now turn to quantifying the number of spurious spikes in a given event, as spurious spikes increase the readout error and thus must be included in our study of $\sigma_{readout}$. With the objective of quantifying the number of spurious spikes in an event, consider an event that starts at time $t_{thresh,1}$ and ends at time $t_{thresh,2}$. At time $t_{thresh,1}$, the top neuron in the membrane potential packet has just reached threshold, and the rest of the neurons in the packet are below threshold. At time $t_{thresh,1}$, we can estimate the typical position of the mean of the membrane potential packet, which we denote as $\bar{V}$, by computing what the value of the mean must be so that the tail of the membrane potential distribution that is above threshold is $1/N$. When this tail probability is above $1/N$ it is likely that out of the N neurons, the top neuron has just crossed threshold. The probability density of the membrane potential packet is approximately Gaussian with width σ_{OU}. Thus the approximate position of the mean $\bar{V}$ of the membrane potential packet relative to the threshold, i.e. $\theta := 1/2 - \bar{V}$, is then estimated via the condition

$$\frac{1}{N} = \int_{\theta}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_{OU}} e^{-\frac{z^2}{2\sigma_{OU}^2}} dz, \tag{S83}$$

or equivalently,

$$\theta = \sqrt{2}\sigma_{OU} \ \text{erfc}^{-1}\left(\frac{2}{N}\right). \tag{S84}$$

Next, we integrate the passage of the Gaussian packet across the threshold during the delay Δ to estimate the mean number of spurious spikes, which we define as λ:

$$\lambda = N \int_{\theta-\delta/\tau}^{\theta} \frac{1}{\sqrt{2\pi}\sigma_{OU}} e^{-\frac{z^2}{2\sigma_{OU}^2}} dz. \tag{S85}$$

To understand this expression, note that between the time $t_{thresh,1}$ to time $t_{thresh,1} + \Delta$, the mean of the membrane potential distribution is shifting up at a rate N/τ. So over this duration of length $\Delta = \delta/N$ the mean shifts up by δ/τ. Thus the tail above threshold also shifts up. This integral reflects an integral over the additional probability mass of the membrane potential distribution that is above threshold between the time of the first spike $t_{thresh,1}$ and the time $t_{thresh,1} + \Delta$ of the first inhibition to all neurons. Multiplying this by N then yields an estimate for the mean number of spurious spikes λ. This simple approximation of Eq. (S83) to Eq. (S85) for the mean number of spurious spikes λ yields a good agreement with numerical simulations (Figure C) when the distribution of membrane potentials is actually drawn from a Gaussian. We note though that the approximation error consistently corresponds to a slight overestimate at finite N; this appears to be a finite size effect as this error diminishes as N becomes large, as demonstrated in (Figure C).

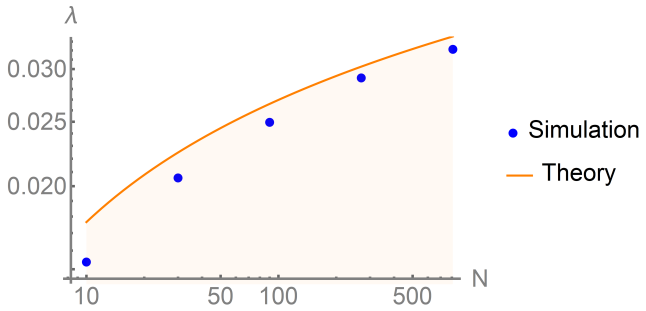

Figure C: The mean number of spurious spikes λ from Monte Carlo simulations is upper-bounded by Eq. (S85). In these simulations, we sample the number of spurious spikes 2×10^5 times, using the following procedure for each sample. First, we take N draws from a Gaussian of width $\sigma_{OU} = 1$ and mean zero, with each draw representing the value of a membrane potential relative to the mean membrane potential $\bar{V}$. Second, we take the maximum of these draws to represent the top neuron's membrane potential; we call this θ. Third, we count how many membrane potentials lie between θ and $\theta - \delta$ (we choose $\delta = \tau \times 10^{-2}$ for this figure), and record the count as the number of spurious spikes. Finally, we average over the 2×10^5 samples obtained using this procedure to compute the mean number of spurious spikes λ (blue dots). We also compute λ using Eq. (S83) to Eq. (S85) with the same parameters, $\sigma_{OU} = 1$ and $\delta = \tau \times 10^{-2}$ (orange curve).

Now importantly, we recall the detail that the $O(\Delta)$ time-difference between the top-neuron's self-reset and the inhibition of the rest of the population slightly stretches the upper tail of the membrane potential packet due to the leak term $-\lambda_V V_i(t)$ through the head start effect. Thus the actual tail of the membrane potential packet is slightly thinner than the Gaussian tail we used in our calculation for λ. However, we note that a thinner tail would facilitate even fewer spurious spikes, because the membrane potentials are more spread out. Thus our calculation for λ is a further overestimate, when this effect is taken into account. Also importantly, we have neglected variations due to the noise term $\sqrt{\tau}\sigma\eta_i(t)$ in the membrane potentials that may cause extra neurons to cross threshold and spuriously spike—but on the timescale of the delay Δ, these diffusive fluctuations have a standard deviation of $O(\sigma\sqrt{\Delta}) = O(\sigma\sqrt{\delta/N})$, which is much smaller than the mean membrane potential increase due to the driving current during the delay, which is $O(\Delta N) = O(\delta)$. Thus, these diffusive fluctuations can be safely neglected in computing the mean number of spurious spikes.

Importantly, the mean number of spurious spikes λ is small for high-performing networks. While we have computed an estimate of the mean here, we will further make the approximation that the distribution of the number of spurious spikes is Poisson. In the limit of small λ, where large numbers of spurious spikes are extremely rare, the detailed distribution of spurious spikes is not expected

to have a large impact on the readout error, and so a Poisson approximation with the same mean should suffice to approximately calculate the readout error. We will confirm this expectation below.

A.4.2 Mean readout error

With this knowledge of the distribution of spurious spikes in an event, let us compute $\sigma_{readout}$ by generalizing the calculation in the previous section. In the previous section, we focused on the case where $\lambda_V \ll 1$ for ease of analysis, and found empirically that this case provided an upper-bound for $\sigma_{readout}$ for the case of larger λ_V. Thus here, we will again start by analyzing the $\lambda_V \ll 1$ case, with the expectation that this corresponds to an upper-bound. Now, the primary difference between the dynamics of the network in the previous section with $\Delta = 0$, and this section with $\Delta > 0$, is that we have $1 + l$-spike events, where $l \geq 0$ as opposed to only 1-spike events. So to generalize our calculations for the variation of the readout at a particular time in the previous section, let us define here the value of the readout at a time t_{mid} which is $l\tau/N + \tau/2N$ after the start of an event, to be $\xi := \hat{x}(t_{mid})$. Since we are considering the case of high network performance, i.e. $\lambda \ll 1$, by far the most prevalent type of event is 1-spike events, followed by 2-spike events. Thus our calculation for the standard deviation σ_ξ of the random variable ξ in the previous section carries through to zeroth order in λ. But remarkably, one can also repeat the calculation for σ_ξ by imagining a readout consisting of only 2-spike events, and one recovers the same result, $\sigma_\xi = \sigma/\sqrt{2}N$. (Intuitively, the sums of spike-time variation $\sum_{n=1}^{k} \gamma_n$ in the calculation now contain half as many terms, but each γ_n has twice the variance because 2-spike events have twice the duration of 1-spike events and integrate twice as much noise $\sqrt{\tau}\sigma\eta_i(t)$.) Thus, aside from additional variance from the random interspersing of 2-spike events among 1-spike events, σ_ξ is still $\sigma/\sqrt{2}N$. Notably, we have seen in the soft-threshold model that such an additional variance is higher-order in $1/N$. However, in the LIF model, we also have the possibility that the membrane potential packet retains a leaky memory of its history, thus 2-spike events may tend to occur in *runs* of some typical length χ (a correlation length) as opposed to the single, randomly interspersed 2-spike events that we considered in the soft-threshold model. In the LIF model, a 2-spike event is more likely to follow a preceding 2-spike event than a preceding 1-spike event, because in the preceding 2-spike event, a second-to-top neuron is already close to threshold, ready to create a spurious spike with relatively high probability. To understand the magnitude of the additional variance contributed to σ_ξ^2 due to 2-spike event runs of length χ, one can repeat the calculation for Eq. (S45) with 2-spike event run replacements instead of individual 2-spike event replacements, and one finds that the additional variance to ξ scales as χ/N^3. So as long as χ scales $< O(N)$, we can still safely neglect this additional variance contribution, and our result of $\sigma_\xi = \sigma/\sqrt{2}N$ still holds to leading order in N.

Equipped with the variance σ_ξ^2 of the readout at time t_{mid}, the next step toward computing $\sigma_{readout}$ is computing the integrated squared deviation s_i

for an event, because $\sigma_{readout}$ is simply a sum of the s_i (Eq. (S20)). We can categorize events by spike-number just like we did in the soft-threshold model, so that we can use the simplified expression for $\sigma_{readout}$, Eq. (S30), that expresses the sum over all s_i as a weighted sum of expectation values of $s_i(l)$, where $s_i(l)$ is defined as s_i conditioned on the event containing $l+1$-spikes. To compute an instance of $s_i(l)$ then, we can take our expression for s_i in the previous section (Eq. (S75)) and modify it to include the fact that the event $s_i(l)$ contains l spurious spikes. We obtain the equation below, and explain the modifications we made in turn.

$$s_i(l) = \int_0^{t^i_{fp}(l)} \left[\left(\xi_i + \frac{1}{2N} + \frac{l}{N} - \frac{1}{\tau} t' \right) - \langle \hat{x}(t) \rangle_t \right]^2 dt'. \tag{S86}$$

First, we have introduced the random variable $t^i_{fp}(l)$, which is the duration of an $l+1$ spike event. Since the membrane potentials of an $l+1$-spike event must recover to threshold from a decrement of $l+1$, the moments of $t^i_{fp}(l)$ are given by the first-passage time distribution of a particle undergoing Brownian motion (this is using our assumption of small λ_V) with mean $(l+1)\tau/N$ and standard deviation that is $O(1/N^{3/2})$, which we can neglect from our calculations as we did in the previous section. Second, we have added the term $\frac{l}{N}$ to denote the increment in the readout near the beginning of the event due to the spurious spikes that occur within a time Δ from the beginning of the event. Note that we have collapsed the spike-times of the l spurious spikes over a time interval of size Δ, to a point time-interval at the beginning of the event. To understand why this approximation accounts for the leading-order effect of delays, recall that the time $\Delta \ll \tau/N$, and thus the contribution to the integral in Eq. (S86) from the $O(\Delta)$ spike-time variability of the l spurious spikes is much less than the contribution of the $\frac{l}{N}$ term, which is integrated over the entire $O(\tau/N)$ duration of the event. Thus, collapsing the spike-times serves to capture the leading-order effect of the delay Δ—that of the $\frac{l}{N}$ term.

Now, to evaluate Eq. (S86), we still need to compute the mean readout $\langle \hat{x}(t) \rangle_t$. But instead of going through another calculation to evaluate $\langle \hat{x}(t) \rangle_t$, we note that we are already describing an upper-bound for $\sigma_{readout}$ because the result from the previous section (Eq. (S82)) is an upper-bound for general λ_V, and our estimate for the number of spurious spikes λ (Eq. (S85)) is also an overestimate. Thus here too, we have the freedom to make an overestimate in our calculation of $s_i(l)$, and the usage of such an $s_i(l)$ will ultimately result in an upper-bound for $\sigma_{readout}$. We take advantage of this freedom to avoid the additional complexity of calculating $\langle \hat{x}(t) \rangle_t$, and instead substitute the result $\langle \hat{x}(t) \rangle_t = \mathbb{E}(\xi)$ from the case of zero delays (Eq. (S78)) into Eq. (S86). Clearly, $\langle \hat{x}(t) \rangle_t$ may deviate from this result because in the case of delay $\Delta > 0$, there are $1 + l$-spike events to consider, and not just 1-spike events. But crucially, the actual value of $\langle \hat{x}(t) \rangle_t$ minimizes the mean squared deviation $\sigma_{readout}$ because of a general property of computations for mean-squared-error (Eq. (S17)). Namely, calculating deviations relative to the mean of a random variable minimizes mean-squared error. Thus any value we substitute for $\langle \hat{x}(t) \rangle_t$ can only

yield the exact value or overestimate for the expectation of $s_i(l)$, $\langle s_i(l)\rangle_i$, which is the quantity we will use to compute $\sigma_{readout}$ in Eq. (S30). Therefore, let us implement this substitution, keeping in mind that this provides an overestimate. Eq. (S86) becomes

$$s_i(l) = \int_0^{t^i_{fp}(l)} \left[\left(\xi_i + \frac{1}{2N} + \frac{l}{N} - \frac{1}{\tau}t'\right) - \mathbb{E}(\xi)\right]^2 dt' \tag{S87}$$

$$= \int_0^{t^i_{fp}(l)} \left[\sigma_\xi y_i + \frac{1}{2N} + \frac{l}{N} - \frac{1}{\tau}t'\right]^2 dt' \tag{S88}$$

where we have cancelled $\mathbb{E}(\xi)$ in the second line and introduced the unit Gaussian random variable y_i to express the variation in ξ_i. Taking the expectation over the $s_i(l)$, we then obtain

$$\langle s_i(l)\rangle_i = \frac{1}{N^2}\frac{1}{12}(l+1)(1+2l+4l^2+6\sigma^2) \tag{S89}$$

Then, to compute $\sigma_{readout}$, we can substitute Eq. (S89) into Eq. (S30):

$$\sigma^2_{readout} = \frac{\sum_{l=0}^{\infty} p(l)\langle s_i(l)\rangle_i}{\sum_{l=0}^{\infty} p(l)(l+1)\tau/N} \tag{S90}$$

$$= \frac{1}{N^2}\left(\frac{\sigma^2}{2} + \frac{1+13\lambda+18\lambda^2+4\lambda^3}{12(1+\lambda)}\right) \tag{S91}$$

$$= \frac{1}{N^2}\left(\frac{1}{12} + \frac{\sigma^2}{2} + \lambda + O(\lambda^2)\right) \tag{S92}$$

where $p(l)$ is a Poisson distribution as defined in Eq. (S24), but with λ from Eq. (S85), and we recall that the denominator represents the mean duration of an event. Now, recalling that we made several overestimates during our derivation—the small λ_V limit corresponds to an upper-bound for $\sigma_{readout}$ for general λ_V, our estimate for λ was an overestimate (Eq. (S85)), and our choice to compute deviations relative to the $\mathbb{E}(\xi)$ instead of the mean readout was an overestimate—and that we made the central limit theorem approximation in the computation of σ_ξ, this expression for $\sigma_{readout}$ is in fact an approximate upper-bound. So we write

$$\sigma^2_{readout} \lesssim \frac{1}{N^2}\left(\frac{1}{12} + \frac{\sigma^2}{2} + \lambda\right). \tag{S93}$$

where we have also dropped high-order terms in λ because we are interested in networks with high performance ($\lambda \ll 1$).

Importantly, Eq. (S91) can be evaluated numerically (numerically evaluating Eq. (S85)) and compared against simulation (Figure 4). Furthermore, we can numerically compute the minimal readout standard deviation, $\sigma^*_{readout}$, and the associated optimal noise level, σ^* (Figure 4). Finally, we numerically differentiate $\sigma^*_{readout}$ and σ^* with respect to the delay δ and observe that $\sigma^*_{readout}$

and σ^* grow as

$$\sigma^*_{readout} \sim (\delta/\tau)^{2/3} \qquad \text{and} \tag{S94}$$

$$\sigma^* \sim (\delta/\tau)^{1/3} \tag{S95}$$

for $\delta \ll \tau$, which matches the behavior of the soft-threshold model, Eq. (S39) and Eq. (S38) (Figure D).

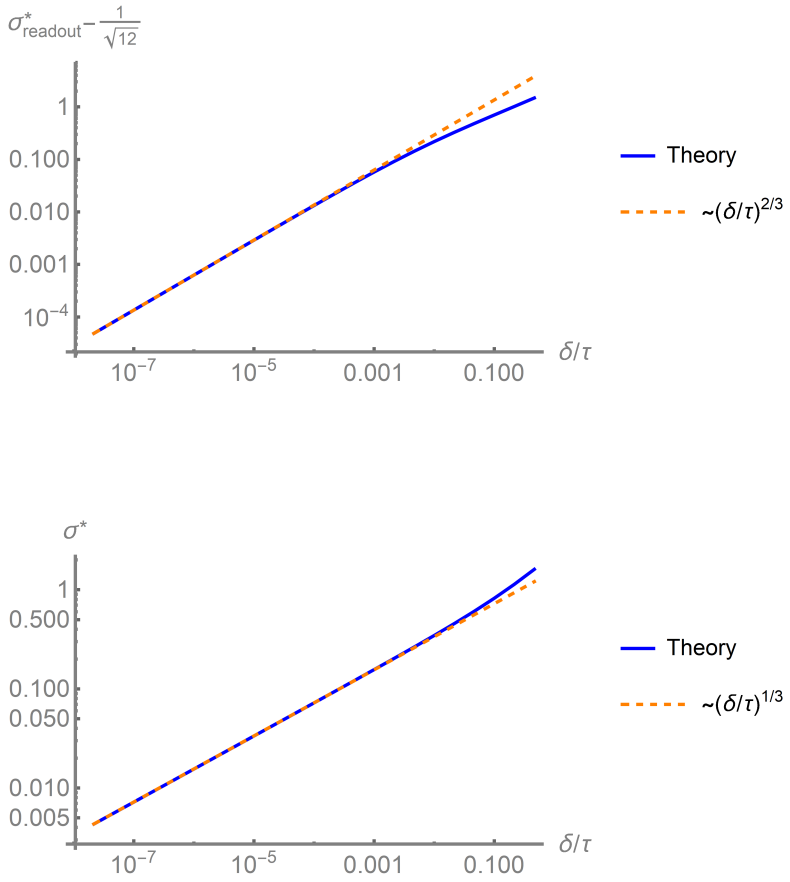

Figure D: Numerically minimizing Eq. (S91) with respect to σ, we find that $\sigma^*_{readout}$ and σ^* (top/bottom, solid blue curve) grow with small delay $\delta \ll \tau$ as $\sim (\delta/\tau)^{2/3}$ and $\sim (\delta/\tau)^{1/3}$, respectively (top/bottom, dashed orange line). Note that this limiting behavior matches that of the soft-threshold model. Noise enters the soft-threshold model as the standard deviation of the first-spike time, $\frac{1}{N\rho}$. And in the soft-threshold model, $\lambda^* \sim (\delta/\tau)^{2/3}$ implies that $\frac{1}{\rho*} = \frac{\delta}{\lambda^*} \sim (\delta/\tau)^{1/3}$, which matches the $\sigma^* \sim (\delta/\tau)^{1/3}$ that we have here for the LIF model.

B Simulation details

We use $\tau = 1$ in all simulations.

B.1 Soft-threshold model

In Figure 2b, our simulations take advantage of the small delay Δ ($\Delta = \delta/N$, where $\delta \ll \tau$) and large N limit of the soft-threshold model to efficiently compute spike-times t_k in Eq. (S6) and empirically estimate $\sigma_{readout}$. We first consider when all the neurons cross threshold together, and the neurons spike with total probability rate $N\rho$. The first-spike time is an exponential random variable with mean $1/N\rho$, and so in our simulations, we simply draw the first-spike time from this exponential distribution. Next, we consider the time after the first spike, during the delay Δ. The other $N-1 \approx N$ neurons continue to spike probabilistically, with each neuron's next-spike time exponentially distributed with mean $1/\rho$, in the absence of inhibition. Here we draw N next-spike times from this exponential distribution. Of course, inhibition arrives after a time Δ, and so the only neurons that spike are the ones with next-spike time less than Δ; we take these as the spuriously spiking neurons, and their spike times are given by the first-spike time, plus the next-spike time for each neuron. After the inhibition arrives from the first-spike neuron, the entire population is subthreshold, and the population returns to threshold a time $(1+l)\tau/N$ after the first spike, where l is the number of spurious spikes. At this point, we repeat our procedure, starting by drawing a new first-spike time, and so on. This yields a list of spike-times, which we insert into Eq. (S6). We use $N = 32$ neurons, run the simulation for 3×10^4 first-spikes, and sample 1.5×10^4 random times in $\hat{x}(t)$ to empirically calculate $\sigma_{readout}$. We sample from within the last 2×10^4 spikes in the simulation, to ensure that the readout has reached the steady state.

This simulation protocol has the advantage that it avoids traditional Euler-timestep discretized time, which allows us to easily examine small delays. However, notably, this protocol does not consider the possibility that the first spike takes so long to occur that the population travels so far above threshold and the first spike is insufficient to return the population to subthreshold. Importantly, for small δ, ρ is large, and the mean first-spike time is small. Furthermore, very large first-spike times are exponentially suppressed; thus the simulation is accurate for small δ.

B.2 LIF model

In Figures 3 and 4, we perform Euler time-step simulations of Eq. (S1) with $dt = 0.0001$, and we use the second half of the simulation when computing $\sigma_{readout}$.

For Figure 3a, we use $N = 32$, $\sigma = 0.1$, and 1562500 time-steps in each simulation (each dot). For Figure 3c, we use $N = 64$ and 781250 time-steps.

For Figure 4d, we use $\sigma = 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1., 3., 10.$, or $30.$, and 781250 time-steps for each simulation. In Figure 4e, we take the minimum over the $\sigma_{readout}$s from Figure 4d, for each delay value δ, to compute $\sigma^*_{readout}$ and σ^*.

B.3 Simulation code

We include here our Euler time-step simulation code used for the LIF model. We have described the other, simpler computations supporting this work directly in the text.

```
# Julia 1.3.1
using LinearAlgebra

# Spiking neural network simulation
function snn_sim(;
    D = 1,                  # Dimension of x(t), NOTE D=1 for this work
    N = 32,                 # Number of neurons
    lambda_V = 0.1,         # Membrane leak
    sigma_V0 = 0.,          # Standard deviation of membrane potential
                            # initialization
    n_steps = 1562500,      # Number of simulation time steps
    n_step_save = 5000,     # Number of steps between each snapshot save of
                            # simulation state
    n_step_delay = 30,      # Length of axonal propagation delay in number of
                            # time steps
    hard_spike = true,      # true => hard threshold, false => soft threshold
    dt = .0001,             # Time step size
    W_input = Nothing,      # Input weight connectivity matrix
    membrane_noise = 1.,    # Noise level sigma
    thres = Nothing,        # Neuron thresholds
    prob_fire_in_dt = 1.,   # Probability of spiking during time step dt (for
                            # soft threshold)
    omega_f = Nothing,      # Connectivity weight matrix
    read_out = Nothing,     # Readout weights
    A = Nothing,            # Emulated D-dimensional dynamical system
    only_one_spike_per_dt = false # true => only allow 1 spike per time step
                                  # (for zero delay), false => multiple
                                  # neurons can spike in a single time step
    )

    # Example values
    if W_input == Nothing
        W_input = N * ones(N,D)
    end
    if omega_f == Nothing
        omega_f = ones(N, N)
    end
    if read_out == Nothing
```

```
    read_out = 1.0 / N * ones(1, N)
end
if thres == Nothing
    thres = 0.5 * ones(N)
end

# Setup for emulating x(t) as a low-pass filter of constant 1-D input c(t) = 1
if A == Nothing
    A = -1. * I
end
c = ones(n_steps,1)

# Initialize simulation parameters
V0 = sigma_V0 * randn(N) # Membrane potentials
r0 = zeros(N) # Firing rates
x0 = [0.] # x(t)

V = copy(V0)
x = copy(x0)
r = copy(r0)

# Storage for simulation history
history_length = trunc(Int, n_steps/n_step_save)
V_history = zeros(history_length, N)
x_history = zeros(history_length,D)
r_history = zeros(history_length,N)
x_hat_history = zeros(history_length,D)

# Initialize readout error metrics
x_hat_squared_sum = 0.
x_hat_sum = 0.
error_calc_start = trunc(Int, n_steps/2) # Burn-in period

# Delayed spike delivery queue and spike time list
spike_queue = []
spike_times = []

# Euler time-step simulation
for step in 1:n_steps

    V_dot = -lambda_V * V + W_input * c[step,:]

    if membrane_noise != 0.
        V_dot += membrane_noise/sqrt(dt) * randn(N)
    end
```

```
V += V_dot * dt

# Check for neurons crossing threshold
for n in 1:N
    if V[n] > thres[n]
        if hard_spike || rand() <= prob_fire_in_dt
            index_and_delivery = (n, step + n_step_delay)
            push!(spike_queue, index_and_delivery)
            push!(spike_times, index_and_delivery)
            V[n] -= omega_f[n,n] # Immediate self-reset

            if only_one_spike_per_dt
                break
            end
        end
    end
end

# Deliver delayed spikes
while length(spike_queue) > 0 && step >= spike_queue[1][2]
    spike_index = splice!(spike_queue,1)[1]
    V -= omega_f[:,spike_index]
    V[spike_index] += omega_f[spike_index, spike_index] # Undo extra
                                                        # self-reset from
                                                        # previous line
    r[spike_index] += 1.
end

# Update state of D-dimensional dynamical system
x_dot = A * x + c[step,:]
x += x_dot * dt

# Update firing rates and output
r_dot = -r
r += r_dot * dt

x_hat = read_out * r

# Compute error starting after burn-in period
if step > error_calc_start
    x_hat_squared_sum += norm(x_hat) ^ 2
    x_hat_sum += norm(x_hat)
end

# Log simulation parameters
if step % n_step_save == 0
```

```
            history_index = trunc(Int, step/n_step_save)
            V_history[history_index, :] = V
            r_history[history_index, :] = r
            x_history[history_index, :] = x
            x_hat_history[history_index, :] = x_hat
        end

    end

    sigma_readout2 = (x_hat_squared_sum / (n_steps - error_calc_start) -
                      (x_hat_sum / (n_steps - error_calc_start))^2 )

    return Dict(
    :N => N,
    :D => D,
    :V_history => V_history,
    :r_history => r_history,
    :x_history => x_history,
    :x_hat_history => x_hat_history,
    :spike_times => spike_times,
    :sigma_readout2 => sigma_readout2,
    )

end
```

References

[1] George E Uhlenbeck and Leonard S Ornstein. On the theory of the brownian motion. *Physical review*, 36(5):823, 1930.

[2] Henry C Tuckwell. *Introduction to theoretical neurobiology: volume 2, nonlinear and stochastic theories*, volume 8. Cambridge University Press, 1988.