Scaling Laws and Tradeoffs in Recurrent Networks of Expressive Neurons

Aaron Spieler[1,2] **Georg Martius**[1,3] **Anna Levina**[1,2]

[1]University of Tübingen, Germany
[2]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[3]Max Planck Institute for Intelligent Systems, Tübingen, Germany

Abstract

Cortical neurons are complex, multi-timescale processors wired into recurrent circuits, shaped by long evolutionary pressure under stringent biological constraints. Mainstream machine learning, by contrast, predominantly builds models from extremely simple units, a default inherited from early neural-network theory. We treat this as a normative architectural question. How should one split a fixed parameter budget P between the number of units N, per-unit effective complexity k_e, and per-unit connectivity k_c? What controls the optimal allocation? This calls for a model in which per-unit complexity can be tuned independently of width and connectivity. Accordingly, we introduce the ELM Network, whose recurrent layer is built from Expressive Leaky Memory (ELM) neurons, chosen to mirror functional components of cortical neurons: multi-timescale memory, structured synaptic integration, and nonlinear internal computation. The architecture allows for individually adjusting N, k_e, and k_c and trains stably across orders of magnitude in scale. We evaluate the model on two qualitatively different sequence benchmarks: the neuromorphic SHD-Adding task and Enwik8 character-level language modeling. Performance improves monotonically along each of the three axes individually. Under a fixed budget, a clear non-trivial optimum emerges in their tradeoff, and larger budgets favor both more numerous *and* more complex neurons. A closed-form information-theoretic proxy captures these tradeoffs and attributes the diminishing returns at two ends to: per-neuron signal-to-noise saturation and across-neuron redundancy. Connectivity enters as a related mechanism that helps neurons learn distinct signals. A hyperparameter sweep spanning three orders of magnitude in trainable parameters traces a near-Pareto-frontier scaling law consistent with the proxy, mapping the budget-constrained tradeoff surface between unit count, unit complexity, and connectivity. This suggests that the simple-unit default in ML is not obviously optimal once this tradeoff surface is probed, and offers a normative lens on cortex's reliance on complex spatio-temporal integrators.

1 Introduction

Cortical neurons are sophisticated spatio-temporal processors. Their dendritic morphology, active conductances, and multi-timescale internal dynamics support nonlinear integration well beyond a point-wise nonlinearity [2–4] (Fig. 1a). Mainstream machine learning, in contrast, builds models from simple units, a default inherited from early neural-network theory [5] and largely unchanged across modern recurrent, state-space, and attention-based architectures [6–8].

The two fields have increased model complexity along separate axes. Computational neuroscience has refined the single-neuron model to match the experimental observations along two strands that are rarely combined. One strand enriches the dynamical machinery at the soma, layering subthreshold currents, adaptation, spike-frequency dependence, and energy or metabolic constraints onto an

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

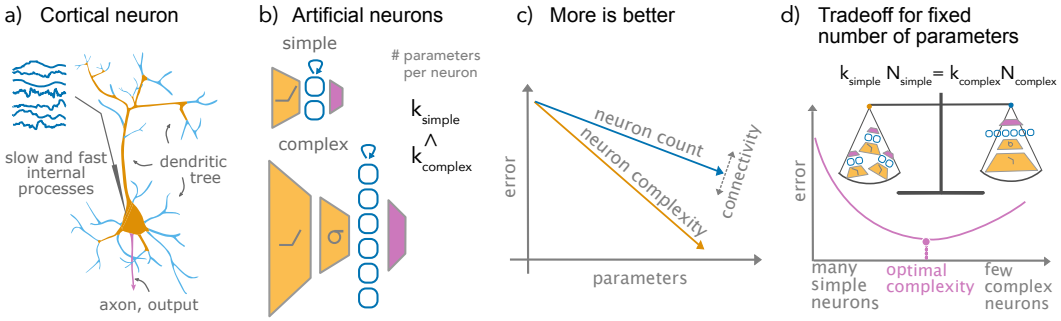


Figure 1: **Complexity of cortical neurons motivates the search for the optimal complexity of a unit in parameter-constrained recurrent networks. a)** Cortical neurons combine complex dendritic structure with rich internal dynamics, making them powerful spatio-temporal processing units. **b)** Using ELM neurons [1], from simple integrators with two memory units and a few parameters k_{simple} to large models exceeding the complexity of cortical neurons with many parameters $k_{\text{complex}} \gg k_{\text{simple}}$. **c)** Recurrent networks with more neurons or neurons of larger complexity are expected to perform better; connectivity affects the slope. **d)** Under a fixed parameter budget, there is a non-trivial tradeoff between network size and neuron complexity, which our theory explains.

otherwise point-like unit [9–13]. The other formalizes spatial integration carried out in the dendritic tree, from the canonical two-layer model of pyramidal computation to active dendritic compartments and structured input integration [2, 14–18]. Only a few trainable neuron models have attempted to flexibly capture both directions, with most efforts focusing on the single-neuron setting [19, 1].

Machine learning has mostly moved along a different axis: instead of enriching computation within units, modern architectures keep units simple and scale by structuring interactions—through convolution, gating, attention, and connectivity [20, 21, 6, 22, 7]—while stacking layers wide and deep. To assess how much additional per-unit complexity is beneficial for efficient computation and to relate these findings to the complexity of cortical neurons [14, 16, 19, 3], we require a setup in which unit complexity is a controllable parameter within an end-to-end trained network. Scaling laws provide a natural framework for this question: they describe how performance improves as resources increase along specific axes and can inform how to allocate a fixed budget across design choices [23, 24]. Prior work has primarily studied scaling with model size, data, and compute, often observing power-law trends with parameter-specific exponents. By contrast, scaling with per-unit complexity remains largely unexplored, despite biological neurons suggesting it may be a relevant axis.

To address this gap, we introduce the ELM Network, a wide recurrent model whose per-unit complexity can be adjusted along neuron-relevant dimensions and that trains stably across three orders of magnitude in trainable parameters, building on the Expressive Leaky Memory neuron [1]. To avoid scaling artifacts from hardware accelerators, we use trainable parameters under a fixed training setup as a cross-disciplinary budget. We sweep the three knobs: number of neurons N, internal neuron complexity k_{e}, and connectivity parameters k_{c} on two contrasting sequence benchmarks: the neuromorphic SHD-Adding task [25, 1] and Enwik8 character-level language modeling. Performance grows monotonically with each axis taken alone, but under a fixed budget, a clear, nontrivial optimum emerges in their joint trade-off (Fig. 1c,d), and the location of that optimum shifts toward both larger and more complex units as the budget increases. We complement these empirical results with a closed-form information-theoretic model whose interpretable parameters track distinct architectural axes in our experiments and qualitatively reproduce the observed trade-offs under a joint fit. These findings undercut the assumption that simple units are an obviously optimal default in machine learning and lend a normative reading to cortex's reliance on complex, multi-timescale units.

Our contributions are the following:

1. The ELM Network as a robust and scalable model system for recurrent computation with expressive neurons, in which neuron count, complexity, and connections vary independently.
2. Experimental scaling and tradeoff results for such networks showing monotonic gains along each axis alone, but non-trivial optima under fixed parameter budget.
3. A compact information-theoretic framework for explaining the tradeoffs in such networks in terms of single-neuron signal-to-noise and population activity redundancy.

2 The Model Architecture

We assemble Expressive Leaky Memory (ELM) neurons [1] into a recurrent architecture for sequential processing, comprising three nested hierarchies: neuron, layer, and network (Fig. 2).

The ELM Neuron. Each ELM neuron is a recurrent cell with d_m leaky memory units and a single scalar output; schematic overview in Figure 2a. At each time step, the neuron **(i)** weights its synaptic inputs $\boldsymbol{z_t} \in \mathbb{R}^{d_s}$ using $\boldsymbol{w_s} \in \mathbb{R}^{d_s}$ ($d_s = d_{\text{tree}} \cdot d_{\text{branch}}$), groups them into d_{tree} branches, and sums within each to get $\boldsymbol{b_t} \in \mathbb{R}^{d_{\text{tree}}}$; **(ii)** produces a bounded memory update proposal $\Delta\mathbf{m}_t \in \mathbb{R}^{d_m}$ by passing the branch activations and the decayed memory state through a nonlinear MLP with l_{mlp} hidden layers of width d_{mlp}; **(iii)** updates the memory units $\boldsymbol{m}_t \in \mathbb{R}^{d_m}$ via leaky integration with per-unit timescales $\boldsymbol{\tau}_m \in \mathbb{R}^{d_m}$; **(iv)** computes an exponential moving average (ema) $r_t \in \mathbb{R}$ of the linear memory readout $\mathbf{w}_r{}^\top \mathbf{m}_t$ using timescale $\tau_r \in \mathbb{R}$; and **(v)** applies a ReLU activation with a per-neuron bias to the readout with its ema subtracted, implementing a temporal high-pass filter, and yielding the neuron's activity $a_t \in \mathbb{R}$. Concretely, each neuron implements $\text{ELM}(\boldsymbol{z}_t)$ defined by:

$$
\begin{aligned}
&\text{(i)} && b_t = c \cdot \text{branch_sum}(\boldsymbol{z}_t \odot \boldsymbol{w}_s) \\
&\text{(ii)} && \Delta\boldsymbol{m}_t = \tanh\big(\text{MLP}_{\boldsymbol{w}_p}([b_t, \boldsymbol{\kappa}_m \odot \boldsymbol{m}_{t-1}])\big) \\
&\text{(iii)} && \boldsymbol{m}_t = \boldsymbol{\kappa}_m \odot \boldsymbol{m}_{t-1} + (1 - \boldsymbol{\kappa}_\lambda) \odot \Delta\boldsymbol{m}_t \\
&\text{(iv)} && r_t = \kappa_r r_{t-1} + (1 - \kappa_r)\boldsymbol{w}_r{}^\top \boldsymbol{m}_t \\
&\text{(v)} && a_t = \text{ReLU}\big(b + \boldsymbol{w}_r{}^\top \boldsymbol{m}_t - r_t\big)
\end{aligned}
\tag{1}
$$

where $\kappa_r = \exp(-1/\tau_r)$, $\boldsymbol{\kappa_m} = \exp(-1/\boldsymbol{\tau_m})$, $\boldsymbol{\kappa_\lambda} = \exp(-\lambda/\boldsymbol{\tau_m})$. Learnable: $\boldsymbol{w_s}, \boldsymbol{w_p}, \boldsymbol{w_r}, b$. Fixed: timescales $\boldsymbol{\tau_m}, \tau_r$, input scale c, and timescale ratio λ. The branch_sum sums contiguous segments of size d_{branch}. The neuron's expressivity is primarily controlled through d_m, l_{mlp}, and $\tau_{\max}$.

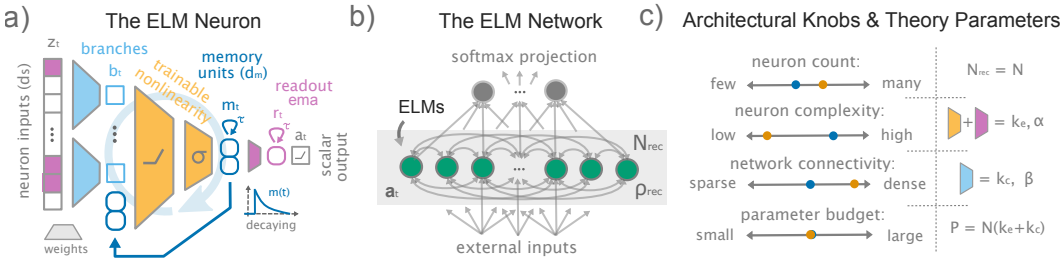


Figure 2: **A stable and flexible model system for studying scaling and tradeoffs in recurrent networks of expressive neurons. a)** The modified Expressive Leaky Memory (ELM) neuron. **b)** ELM neurons are assembled as ELM Network, a doubly recurrent sequence model whose computational core is a wide recurrent hidden layer in which each neuron itself is recurrent; followed by a smaller readout layer and output projection. **c)** (left) The main study axes as explicit architectural knobs (right) Corresponding theory parameters: neuron count to N, effective neuron complexity to (k_e, α), network connectivity structure to (k_c, β), and total budget to $P = N(k_e + k_c)$. We count $k_e = \#w_p + \#w_r$ and $k_c = \#w_s$. Theory α, β are introduced in Section 4.2. Model parameters in Appendix. Tab. 1.

The ELM Layer. An ELM layer consists of N_{rec} independently parameterized ELM neurons, each receiving $P/N_{\text{rec}} = k_e + k_c$ trainable parameters of the layer budget P. The layer input at time t is the concatenation of the feed-forward signal $\mathbf{u}_t$ (either external input or the previous layer's activity) and the layer's own previous activity $\mathbf{a}_{t-1}$. Each neuron receives a fixed, uniformly sampled subset of d_s input channels from this concatenation, with ρ_{rec} determining the probability of sampling a recurrent connection at initialization. A layer, therefore, computes:

$$
\begin{aligned}
\boldsymbol{z}_{t,i} &= \text{input_select}_i([\boldsymbol{u}_t, \boldsymbol{a}_{t-1}]), \\
a_{t,i} &= \text{ELM}_i(\boldsymbol{z}_{t,i}).
\end{aligned}
\tag{2}
$$

where $\boldsymbol{a_t} \in \mathbb{R}^{N_{\text{rec}}}$, $\boldsymbol{u_t} \in \mathbb{R}^{d_{\text{inp}}}$. input_select$_i$ gathers the d_s channels assigned to neuron i at initialization, and ELM$_i$ has neuron-i-specific parameters.

The ELM Network. The full model comprises a larger recurrent hidden ELM layer (e.g., $N_{\text{rec}} = 1024$), a smaller feed-forward readout ELM layer with simpler neurons ($l_{\text{mlp}} = 0$, $d_m = 3$) without output filtering or activation nonlinearity, and a final linear layer that maps the readout activity to the target output dimensionality. Text input sequences are embedded via a scaled one-hot encoding.

3 Related Work

Recurrent and modular architectures split the computation across interacting stateful components. Typically, **computational neuroscience** asks how feedback and population dynamics in networks of simple spiking or rate-based neurons support computation [26–30]. However, more intricate neuron models with some tunability have also been used in networks [31–34]. Generally, **deep learning** has focused on performance and found specialized modules [35–37, 8] and parallel module-like multi-head and mixture-of-experts setups [6, 38] to be beneficial; recently, also structured state-spaces [39, 7]. While most of them use vector-valued module outputs (breaking the neuron analogy) and do not separately control module connectivity, some do feature more neuron-like scalar outputs [7, 40, 41]. Uniquely, ELM Networks feature independently parameterized scalar-output neurons with flexibly adjustable internal complexity that is separately tunable from connectivity.

Performance scaling laws provide the natural language for turning architectural choices into resource-allocation questions. Prior work has shown that performance often follows predictable trends with model size, data, and compute across architectures and modalities [42, 23, 43, 24], whereas **budget-constrained tradeoff** investigations explore how finite resources should be allocated across model-shape dimensions [44–46]. Similar budgeted "module" complexity vs. counts experiments in related architectures have also found interior optima [6, 47–51]. Complementary **approximation-theoretic** work studies how expressivity changes with architectural choices such as width and depth [52–56]. In contrast to previous works scaling networks around a fixed computational unit or jointly with connectivity, here, the unit complexity itself and connectivity separately become scaling axes.

Information-theoretic perspectives on neural computation study how neural systems represent signals under noise, redundancy, and resource constraints [57–60]. Efficient coding, sparse coding, information bottleneck, and population coding provide complementary views on how signals are compressed, made task-relevant, and distributed across noisy neural ensembles [58, 61–65], with encoding tradeoffs [66–68]. Our framework follows this tradition, but casts the problem as a resource-allocation tradeoff between single-unit fidelity and population-level aggregation and coordination.

4 Results

In our experimental section, we address three core questions: **1)** How does a recurrent layer's performance scale with neuron count, neuron complexity, and network connectivity, subject to budget constraints? Section 4.1. **2)** Can the scaling tradeoffs be explained in terms of general information-theoretic principles? Section 4.2. **3)** Can we find architectural scaling recipes that trace the Pareto frontier across datasets and parameter budget? Section 4.3. For complete model hyperparameters, detailed training methods, and full derivations, see Appendix A and B.

4.1 Evaluating network scaling and tradeoffs across a wide range on complementary datasets

Our first set of experiments is conducted on the SHD-Adding benchmark [1], a challenging neuromorphic sequence benchmark with biologically plausible spatio-temporal input patterns. Each input sample is a 700×1000 binary spatio-temporal pattern, representing the concatenation of two spoken digits in German or English (see Fig. 14). Individual digits were originally spike-encoded using a bio-inspired cochlear model [25]. The prediction target is the sum of the spoken digits.

Should one invest parameters in network width or neuron complexity? Through robust initialization and stable training, ELM networks manage to reliably solve the task, and we observe roughly monotonic performance improvements for scaling along the number of neurons or neuron complexity (Fig. 3a,b), with networks of roughly 150 neurons or 10 memory units approaching the best previously reported accuracy levels (83% [1]). For a fixed parameter budget, however, these two dimensions compete, and a non-monotonic curve emerges (Fig. 3c), with a steep drop-off in performance for networks with small neurons or few neurons, and a clear in-between optimum that sharpens for neurons trained with binary activation as in spiking neural networks. Thus, the parameter budget in recurrent layers should be carefully balanced between complexity and the number of neurons.

Despite the dataset being one of the most challenging neuromorphic benchmarks, even modest-sized ELM Networks start running into dataset-related saturation effects. To explore scaling across larger network dimensions in all directions, we turn to a complementary character-level language modeling benchmark, Enwik8 [70]. It is comprised of the first 10^8 bytes from Wikipedia in 2006, containing

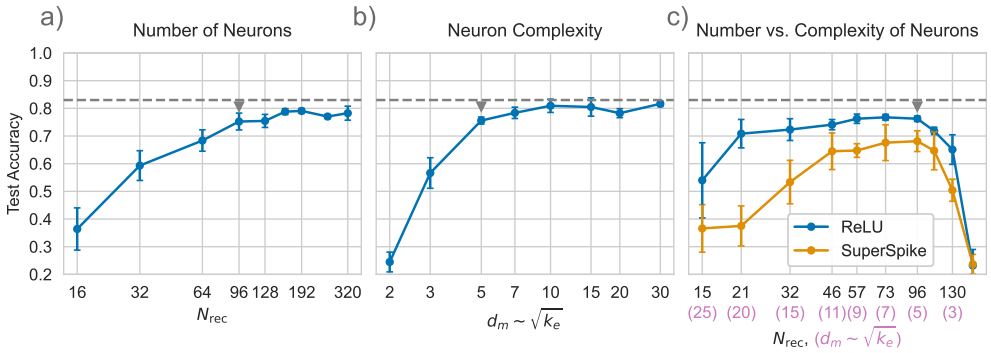


Figure 3: **More neurons or more complex neurons each improve performance on the SHD-Adding task; under a fixed budget, a clear non-trivial optimum emerges.** The reference model (triangle) was chosen to be below saturation along all dimensions. Mean and standard deviation across three seeds are displayed. Test accuracy improves with **a)** the number of neurons N_{rec} and **b)** neuron complexity $k_e \sim d_m^2$ (varying memory units d_m). **c)** Under a fixed parameter budget, a clear optimum emerges in the number-vs-complexity tradeoff; the optimum sharpens with binary-spiking neuron activation SuperSpike [69] (default ReLU). Secondary d_m axis label in pink brackets.

240k articles primarily in English. By using the standard preprocessing pipeline from Transformer XL [71], we get 204 unique bytes, which we one-hot encode as the network input.

Do these scaling tradeoffs generalize to larger dataset and network sizes? The monotonic performance improvement trends observed on SHD-Adding extend on Enwik8 over at least another order of network width and neuron complexity (Fig. 4a,b). Curiously, the individual scaling slopes can be manipulated: by increasing neuron input connections (d_s) proportional to neuron count, the network improves much faster, and if we impose simpler synaptic integration ($l_{\text{mlp}} = 0$), we see slower performance improvements, even when accounting for total parameters (see Appendix Fig. 9). Lastly, we again find a nontrivial optimal neuron complexity for a resource-constrained layer, with the optimum emerging robustly across three parameter budgets (Fig. 4c). Interestingly, we find that increased budget favors network configurations with more *and* more complex neurons; shifting from $(N_{\text{rec}}, d_m) = (512, 15)$ to $(1024, 25)$ when the layer budget is quadrupled. Importantly, this interior-optimum emerges robustly also for different levels of network recurrence or number of network hidden layers (Appendix Fig. 12), wide range of neuron high-pass-filter timescales (Appendix Fig.11), and even alternative budget types like Wall-Time or FLOPS (Appendix Fig. 28).

How does network performance scale with connectivity and recurrence? Treating connectivity as its own scaling dimension, we observe similar monotonic performance improvements (Fig. 4d). However, if we change the number of recurrent or feedforward connections individually, a more nuanced picture emerges: while having too few of either results in a stark drop-off in performance, once sufficient feedforward connections from input exist, almost all performance benefit comes from the additional recurrent connections, a trend that continues well into the regime of multiple connections between pairs of neurons ($d_s \gtrsim N_{\text{rec}} + d_{\text{inp}} = 1228$). When varying the fraction of recurrent connections to total inputs, ρ_{rec}, we find that larger networks favor higher recurrence (Fig. 4e). Lastly, keeping the number of neurons fixed but varying the split of parameters between connectivity k_c and neuron complexity k_e, we find an optimum around N_{rec} connections (Fig. 4f).

4.2 Evaluating network scaling and tradeoffs under an information-theoretic framework

Our previous experiments demonstrate the advantage of neurons with intermediate complexity empirically, yet also raise new questions: why is it better to invest additional parameter budget simultaneously in more *and* more complex neurons, or how exactly do neuron integration capabilities and network connectivity modulate scaling steepness? To get a better understanding of the underlying dependencies, we develop a theoretical framework and establish its link to experimental observations.

An information-theoretic model of a neural layer. We frame a neural layer as a collection of noisy neuron-channels (Fig. 5a,b): each neuron i produces an output $y_i = f_i(x) + n_i$, where f_i

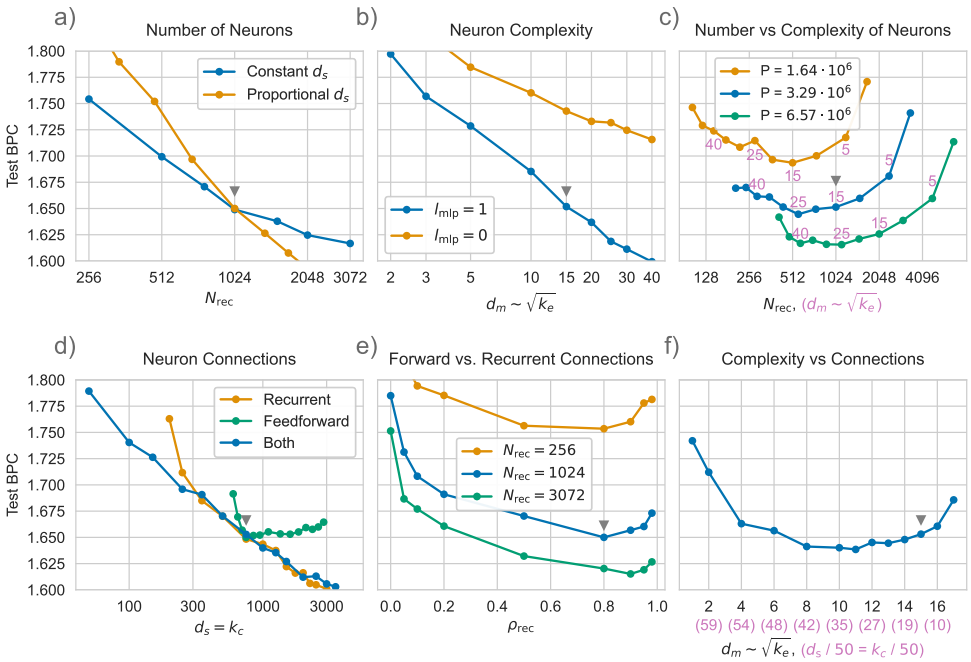


Figure 4: **Monotonic scaling gains extend across vast network sizes; larger budgets favor both more *and* more complex neurons, and connectivity introduces new tradeoff dimensions.** Enwik8: character-level language modeling (test BPC, lower is better) [70]. Reference model marked with a triangle. Exemplary network activity in Appendix Fig. 8. **a, b)** Monotonic improvement with neuron count N_{rec} and per-neuron complexity $k_e \sim d_m^2$ (varying memory units d_m) spanning over an order of magnitude each without saturating, with slope of improvement modulated by network connectivity and neuron integration hierarchy respectively. **c)** Under a fixed budget, a clear non-trivial optimum emerges. It shifts with increased budget, favoring more complex and more numerous neurons. Similar tradeoffs are observable in multi-head attention Transformers; see Appendix Fig. 23. **d)** More synapses d_s improve performance, and recurrent connections drive this improvement far beyond where feedforward connections saturate. **e)** An optimal recurrent fraction ρ_{rec} shifts with network size, roughly tracking the ratio of recurrent neurons to possible presynaptic inputs. **f)** Splitting neuron parameters between complexity and connectivity yields an optimum around N_{rec} connections.

is a task-relevant signal component, and n_i is a task-uninformative residual (approximation error), treated as phenomenological noise [72, 73]. Consequently, the latent signal component $f_i(x)$ is not directly observable, as measurements of a neuron's output y_i will also contain its task-uninformative noise component n_i. Unlike previous works, we further model per-neuron complexity k_e reducing its residual noise (approximation error) n_i, making each neuron-channel output closer to its ideal f_i. The mutual information $I_{\text{rep}} = I(f; y)$ between a layer's noisy output and its task-relevant signal, the *effective representation information*, quantifies how much usable information the layer carries, and is therefore a proxy indicative of task performance. More channels *or* more informative neuron channels increase I_{rep}, inter-neuron signal redundancy reduces it relative to independent channels.

Four assumptions together yield a closed-form expression for I_{rep}: **(A1)** Gaussian signal and noise; **(A2)** power-law signal covariance eigenvalue spectrum $\lambda_i \propto i^{-\beta}$, with smaller β corresponding to a higher effective dimensionality of the signal across channels; **(A3)** per-neuron noise decays as a power law in k_e until saturating at a floor q_∞, $\sigma_n^2(k_e) \propto \max((\gamma k_e)^{-\alpha}, q_\infty)$, and **(A4)** a fully fungible budget P, distributed among $N = P/(k_e + k_c)$ neurons each receiving k_e effective parameters under connectivity overhead k_c, concretely

$$I_{\text{rep}}(k_e) \;=\; \tfrac{1}{2} \sum_{i=1}^{P/(k_e+k_c)} \log_2\!\big(1 + s(k_e)\, i^{-\beta}\big)\,, \qquad s(k_e) \;=\; \min\!\big((\gamma k_e)^{\alpha}, q_\infty^{-1}\big)\,. \tag{3}$$

with $s(k_e)$ as the neuron's parameter dependent signal-to-noise ratio (SNR). The budget trade-off is visible directly: larger k_e raises per-channel SNR but reduces the number of channels N, creating a tension and raising an allocation question. Full derivation in Appendix B.

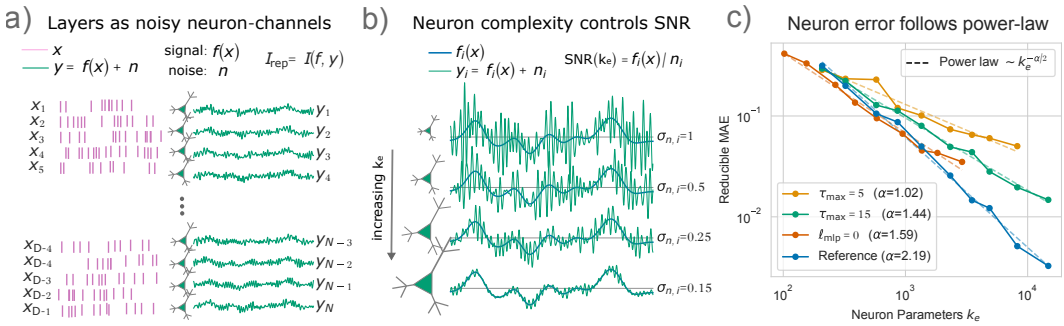


Figure 5: **The *Effective Representation Information* derived from viewing neural layers as noisy channels is motivated by residual-error scaling in single neurons. a)** Neural activations may be viewed as a noisy multi-channel representation of their inputs, with I_{rep} quantifying how much task-relevant information they transmit. **b)** Per-neuron complexity determines the residual variance $\sigma_n^2(k_e)$, a decreasing function of parameter budget k_e, that leads to increasing signal-to-noise ratio $\text{SNR}(k_e)$. **c)** We test this noise ansatz empirically by fitting increasingly complex ELM neurons to single-neuron membrane-voltage recordings from the NeuronIO dataset [19]: the average (three seeds) reducible Mean Absolute Error (MAE) on test set decreases as a power law in neuron parameters $\propto k_e^{-\alpha/2}$ across two orders of magnitude, with α capturing the neuron's spatio-temporal integration settings (e.g. l_{mlp}, $\tau_{\max}$). This new perspective on neural computation motivates the particular form of I_{rep}.

What do the theory parameters mean in ELM Networks, and do the assumptions hold? While some theory parameters are directly accessible from the architecture, the four phenomenological parameters require some deliberation and rest on assumptions that we will examine in the following. **The accessible ones:** $N = N_{\text{rec}}$ is the recurrent-layer width, $k_e = \#w_p + \#w_r$ is the per-neuron internal parameter count, and $k_c = \#w_s$ is the per-neuron connection parameter count. **The noise decay exponent** α describes an individual neuron's spatio-temporal integration expressivity, which also depends on $\tau_{\max}$ or l_{mlp}. The underlying assumption (A3) is testable on the NeuronIO single-neuron benchmark, where per-neuron noise is directly observable as the residual error in fitting an isolated ELM neuron to recorded membrane voltages: the reducible MAE follows a power law in k_e across two orders of magnitude, yielding α from the slope (Fig. 5c). Within ELM Network trained on Enwik8, the per-neuron noise reductions remain close to power law, with its slope tracking the neuron-related expressivity changes qualitatively (Fig. 6g). **The spectral decay exponent** β captures the covariance structure of the task-relevant signal component across neurons (smaller β, higher effective dimensionality, less inter-neuron redundancy), which we expect to be shaped by connectivity: more connections allow neurons to coordinate, reducing redundancy and decreasing β. We test (A2) on ELM Networks, as unlike α, β is a neural layer property and cannot be investigated in individual neurons: the empirical eigenvalue spectrum of the layer's activity covariance matrix is well approximated by a truncated power law with a low-rank cutoff (Fig. 6h, Appendix Fig. 19). (A2) is thus broadly supported in our setting, with deviations confined to the low-eigenvalue tail that contributes negligibly to I_{rep}. **The remaining parameters**, γ and q_∞, set the prefactor and floor of (A3): γ converts model parameter count into dimensionless effective parameters, a conversion efficacy that may be influenced by train steps or optimizer settings, while q_∞ is the residual variance no amount of k_e can reduce, observable as saturation in the error-vs-k_e curve (Appendix Fig. 17,25). Note that these parameters are generally **model- and dataset-specific**; for example, α directly captures neuron-model-specific expressivity, and β also depends on the underlying training data [74].

Do model predictions match the experimental scaling curves? We probe the theoretical framework and its parameter-to-architecture associations through matched theory-to-experiment comparisons on Enwik8 (Fig. 6). The experimental evaluations are paired with theoretical curves obtained from a single joint fit: the curves of Fig. 6a/d and reference curves in (b/e, c/f) share a single tuple $(\alpha, \beta, \gamma, q_\infty)$ and affine mapping $I_{\text{rep}} \to \text{BPC}$. Each set of experiments targets one of the proposed architectural-to-theory correspondences: budget P, the noise-decay exponent α, or the spectral exponent β, all other parameters are kept at reference levels.

We find: **(1)** the joint fit qualitatively reproduces the empirical scaling shapes for all three sensitivity analysis groups, including the location and shift of the optimum with budget (a/d), the wider optima

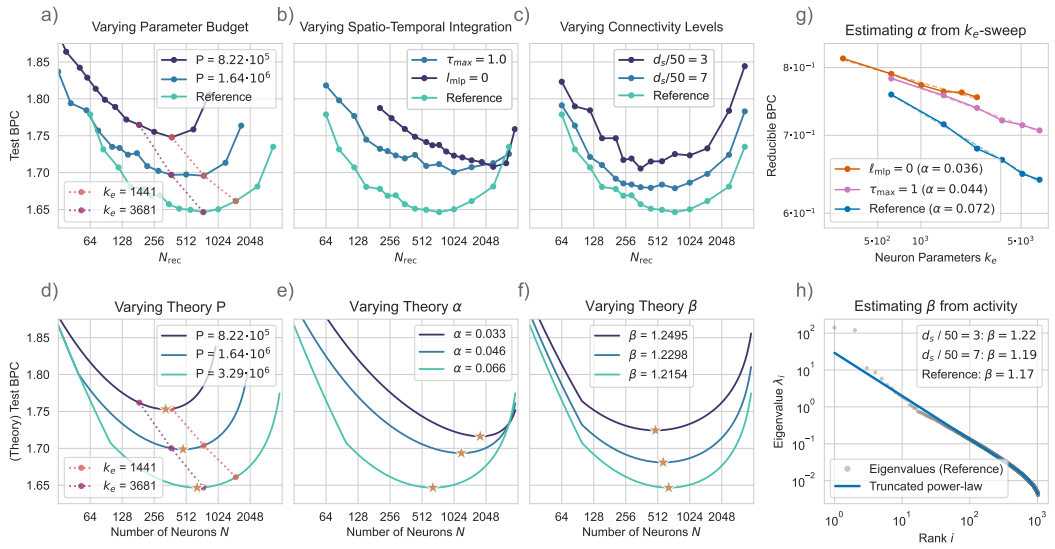


Figure 6: **The theoretical model qualitatively reproduces the empirical scaling tradeoffs and links their shifts to measurable architectural quantities. a–f)** Enwik8 experiments (top row) are paired with corresponding theory proxy (bottom row) obtained via joint fit using shared reference parameters (Pearson $r = 0.9828$ and RMSE $= 0.0092$ BPC). In all panels, the reference model has $d_s = 750$ (synapse connections), $\tau_{\max} = 100$ (longest memory timescale),$l_{\text{mlp}} = 1$ (MLP hidden layers), and theory is affine adjusted using $\text{BPC} = -0.34 \cdot I_{\text{rep}} + 2.7$. Theory optima marked with a golden star. For panels a–f overlaid see Appendix Fig. 24. **a, d)** Optimal performance for intermediate neuron complexity; increasing the budget P shifts the optimum toward more and larger neurons. Curves touch for large k_e once the neurons reach the per-neuron noise floor q_∞. **b, e)** Weakening spatio-temporal integration matches decreasing α and shifts the optimum toward wider layers. All curves cross at a signal-to-noise ratio of one with $k_e = 1/\gamma$. **c, f)** Reducing connectivity corresponds to increasing the spectral decay exponent β, and shifts the optimum toward larger neurons. **g, h)** Direct measurements of α and β on the models with $N_{\text{rec}} = 1024$ are in the same ballpark *and* change in accord with the joint theoretical fits' values. **g)** Estimating α by fitting a power law to the dependence of reducible BPC on single neuron parameter (details Appendix B.5). **h)** Estimating β from the eigenvalue spectrum of the network activity covariance matrix (at memory readout $\boldsymbol{w_r}^\top \boldsymbol{m_t}$) by fitting a truncated power law with shared cutoff parameters (i_c, ν) (details in Appendix Fig. 19). For **predictive framework** experiments, with all theory parameters independently derived, and the proxy predicting budgeted tradeoffs (Pearson $r = 0.9801$), see Appendix Fig. 22

and curve crossings under reduced α (b/e), and the vertical shifts and slight optimum shifts under reduced β (c/f); **(2)** the fitted α and β values are in the same ballpark as independent in-network measurements (Fig. 6g,h), with α biased slightly high and β uniformly biased slightly low, compatible with the α/β fitting degeneracy; and **(3)** the response of measured α and β to architectural changes agrees in direction with the joint-fit predictions, α decreasing when neuron integration is weakened ($\tau_{\max} = 1$, $l_{\text{mlp}} = 0$) and β increasing when connectivity is reduced (d_s lowered). Together, a four-parameter proxy captures the empirical scaling tradeoffs across three architectural axes under a single constrained fit, with each phenomenological exponent independently identifiable from architecture-specific measurements. Note that α, β, γ and q_∞ are defined architecture-agnostically, and may be applicable to any layer of scalar-output units with separately tunable complexity.

4.3 Search for optimal architecture scaling rule via large-scale hyperparameter exploration

The theoretical proxy can predict the optimum's location in (N_{rec}, k_e), but treats k_e and k_c as a scalar parameter count, leaving the internal parameter allocation across d_m (memory units), d_{mlp} (MLP width), d_{tree} (number of branches), d_{branch} (synapses per branch) unspecified. We address this with a large-scale, structured search, over (N_{rec}, d_m) together with configuration rules that constrain $d_{\text{mlp}}, d_{\text{tree}}$, and d_{branch} subject to $d_m \leq d_{\text{mlp}} \leq d_{\text{tree}} \leq d_{\text{branch}}$, and set the recurrence fraction $\rho_{\text{rec}} \sim {}^{N_{\text{rec}}}/_{(N_{\text{rec}}+d_{\text{inp}})}$ for both SHD-Adding and Enwik8 tasks (Fig. 7). Details in Appendix A.3.

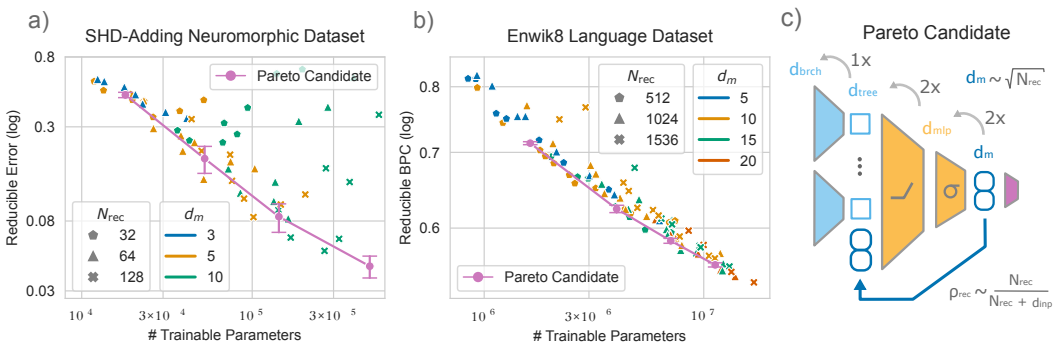


Figure 7: **A simple joint-scaling heuristic consistent with theory traces the empirical Pareto frontier across both datasets and three orders of magnitude.** We perform a structured large-scale hyperparameter search including neuron internal dimensions. The mean across three runs is reported on SHD-Adding and single-run performance on Enwik8. For the Pareto candidates, the mean and standard deviation across five runs are shown for both datasets. **a,b)** On both datasets, networks with more *and* more complex neurons become optimal as budget increases (as before), and a single joint scaling recipe displaying approximate power-law decay can trace the Pareto frontier closely. **c)** The *Pareto Candidate* only has N_{rec} as free scaling variable with all other settings derived from it.

We find a single scaling recipe to perform routinely among the best across both datasets and roughly three orders of magnitude in trainable parameters: $d_m \sim \sqrt{N_{\mathrm{rec}}}$ with $d_{\mathrm{mlp}} = 2d_m$, $d_{\mathrm{tree}} = 2d_{\mathrm{mlp}}$, and $d_{\mathrm{branch}} = d_{\mathrm{tree}}$ scaling per-neuron parameters and connectivity linearly with N_{rec}. The recipe scales per-neuron connectivity aggressively, in line with the theoretical proxies corresponding β's conceptual importance; the resulting reducible-error decay follows an approximate power law consistent with above dataset and neuron noise floor regimes. The theoretical proxy predicts the recipe to hold only until larger neurons reach their individual noise floor q_∞, after which any additional benefit must come from more and better coordinated neurons; a regime we have not yet reached in this experiment ($d_m \leq 20$ vs $d_m \approx 70$ on Enwik8), and which we leave for future work.

5 Discussion

In this work, we studied the empirical and theoretical performance tradeoff between the number, complexity, and connectivity of neurons in recurrent layers under budget constraints. Empirically, performance improves monotonically along each axis, but under a fixed budget, a non-trivial optimum emerges that shifts toward both more *and* more complex neurons as the budget grows. In our information-theoretic framework, the dependence of *Effective Representation Information* (I_{rep}) on the number and complexity of channels reproduces these tradeoff shapes and their shifts under a single constrained joint fit, with each phenomenological exponent independently identifiable from architecture-specific measurements. A joint search yields a simple scaling recipe for model hyperparameters that traces the empirical Pareto frontier across both studied datasets and three orders of magnitude in trainable parameters, consistent with the theoretical proxies' predictions. Our findings position complex per-unit machinery as a budget-efficient design choice, with implications for both architecture design in machine learning and the interpretation of biological neuron complexity.

Limitations: All results are for ELMs and ELM Networks, limiting the generality of the findings. **Experiments.** Models with the same number of parameters can vary significantly in GPU wall-time or memory consumption, due to hardware misalignment. Using a fixed training setup likely degrades performance non-uniformly across tested configurations. Both bias the exact scaling exponents and location of optima. While extensive, the structured Pareto search may have missed optimal configurations outside its search space. **Theory.** The framework abstracts away recurrence, network topology, and optimization dynamics. Its assumptions hold approximately rather than exactly (e.g. truncated vs pure power-law eigenvalue spectrum), and the architecture-to-theory mapping is empirical rather than derived. While independently measured α and β yield qualitatively good proxy predictions, their exact values are not readily identifiable due to significant variance. **Neuroscience.** The phenomenological nature of the ELM neuron, BPTT-based training, and the use of parameter

count as a budget proxy that does not directly capture metabolic cost, together limit direct biological interpretability.

Despite these caveats, our results suggest that per-unit complexity and connectivity machinery deserves systematic study as a scaling axis in its own right, with the compact Effective Representation Information formulation potentially helping to explore it effectively.

Beyond our preliminary experiments with multi-head attention Transformers, signatures of interior optima for similar budgeted module count versus module size experiments in multiple other related architectures [47–51] indicate that the presented framework might extend to, and the observed tradeoff may be present in, more mainstream architectures.

Natural next steps include probing larger network sizes to test the predicted noise-floor regime of the scaling recipe; treating training and data as explicit scaling dimensions; extending the framework to related modular architectures and structured connectivity; and finally, providing guidance for analogous measurements in biological circuits.

Acknowledgments and Disclosure of Funding

This work was funded by the Federal Ministry of Research, Technology and Space (BMFTR) through the RealmNN project under the NeuroAI initiative (grant number 01GQ2508). We acknowledge support from the BMBF through the Tübingen AI Center (FKZ: 01IS18039B). AL is a member of the Cluster of Excellence "Machine Learning: New Perspectives for Science", EXC 2064/1 (project number 39072764).

We thank the Max Planck Computing and Data Facility (MPCDF) for providing computational resources on the Raven high-performance computing system. AS thanks David Beniaguev and Antonio Orvieto for insightful discussions.

The authors declare no competing interests.

References

[1] Aaron Spieler, Nasim Rahaman, Georg Martius, Bernhard Schölkopf, and Anna Levina. The expressive leaky memory neuron: an efficient and expressive phenomenological neuron model can solve long-horizon tasks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vE1e1mLJ0U.

[2] Matthew Larkum. Are dendrites conceptually useful? *Neuroscience*, 2022.

[3] Ido Aizenbud, Daniela Yoeli, David Beniaguev, Christiaan PJ de Kock, Michael London, and Idan Segev. Dendritic morphology and synaptic nonlinearities enhance functional complexity in human cortical neurons. *Proceedings of the National Academy of Sciences*, 123(28):e2533168123, 2026.

[4] Panayiota Poirazi and Athanasia Papoutsi. Illuminating dendritic function with computational models. *Nature Reviews Neuroscience*, 21(6):303–321, 2020.

[5] Snehashish Chakraverty, Deepti Moyi Sahoo, and Nisha Rani Mahato. *McCulloch–Pitts Neural Network Model*, pages 167–173. Springer Singapore, Singapore, 2019. ISBN 978-981-13-7430-2. doi: 10.1007/978-981-13-7430-2_11. URL https://doi.org/10.1007/978-981-13-7430-2_11.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[7] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[8] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *Advances in Neural Information Processing Systems*, 37:107547–107603, 2024.

[9] Eugene M Izhikevich. Which model to use for cortical spiking neurons? *IEEE transactions on neural networks*, 15(5):1063–1070, 2004.

[10] Romain Brette and Wulfram Gerstner. Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *Journal of neurophysiology*, 94(5):3637–3642, 2005.

[11] Wulfram Gerstner, Werner M Kistler, Richard Naud, and Liam Paninski. *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[12] Guillaume Bellec, Darjan Salaj, Anand Subramoney, Robert Legenstein, and Wolfgang Maass. Long short-term memory and learning-to-learn in networks of spiking neurons. *Advances in neural information processing systems*, 31, 2018.

[13] Tanguy Fardet and Anna Levina. Simple models including energy and spike constraints reproduce complex activity patterns and metabolic disruptions. *PLoS Computational Biology*, 16(12):e1008503, 2020.

[14] Panayiota Poirazi, Terrence Brannon, and Bartlett W Mel. Pyramidal neuron as two-layer neural network. *Neuron*, 37(6):989–999, 2003.

[15] Monika P Jadi, Bardia F Behabadi, Alon Poleg-Polsky, Jackie Schiller, and Bartlett W Mel. An augmented two-layer model captures nonlinear analog spatial integration effects in pyramidal neuron dendrites. *Proceedings of the IEEE*, 102(5):782–798, 2014.

[16] Albert Gidon, Timothy Adam Zolnik, Pawel Fidzinski, Felix Bolduan, Athanasia Papoutsi, Panayiota Poirazi, Martin Holtkamp, Imre Vida, and Matthew Evan Larkum. Dendritic action potentials and computation in human layer 2/3 cortical neurons. *Science*, 367(6473):83–87, 2020.

[17] Balázs B Ujfalussy, Judit K Makara, Máté Lengyel, and Tiago Branco. Global and multiplexed dendritic computations under in vivo-like conditions. *Neuron*, 100(3):579–592, 2018.

[18] Greg J Stuart and Nelson Spruston. Dendritic integration: 60 years of progress. *Nature neuroscience*, 18 (12):1713–1721, 2015.

[19] David Beniaguev, Idan Segev, and Michael London. Single cortical neurons as deep artificial neural networks. *Neuron*, 109(17):2727–2739, 2021.

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[21] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[22] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=uYLFoz1vlAC.

[23] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. doi: 10.48550/arXiv.2001.08361.

[24] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2022.

[25] Benjamin Cramer, Yannik Stradmann, Johannes Schemmel, and Friedemann Zenke. The heidelberg spiking data sets for the systematic evaluation of spiking neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[26] Wolfgang Maass. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671, 1997.

[27] Herbert Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach*, volume 5. GMD-Forschungszentrum Informationstechnik Bonn, 2002.

[28] Dean V. Buonomano and Wolfgang Maass. State-dependent computations: spatiotemporal processing in cortical networks. *Nature Reviews Neuroscience*, 10(2):113–125, 2009. doi: 10.1038/nrn2558.

[29] Valerio Mante, David Sussillo, Krishna V. Shenoy, and William T. Newsome. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature*, 503(7474):78–84, 2013. doi: 10.1038/nature12742.

[30] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

[31] Henry Markram, Eilif Muller, Srikanth Ramaswamy, Michael W Reimann, Marwan Abdellah, Carlos Aguado Sanchez, Anastasia Ailamaki, Lidia Alonso-Nanclares, Nicolas Antille, Selim Arsever, et al. Reconstruction and simulation of neocortical microcircuitry. *Cell*, 163(2):456–492, 2015.

[32] João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in neural information processing systems*, 31, 2018.

[33] Manuel Brenner, Florian Hess, Jonas M Mikhaeil, Leonard F Bereska, Zahra Monfared, Po-Chen Kuo, and Daniel Durstewitz. Tractable dendritic rnns for reconstructing nonlinear dynamical systems. In *International conference on machine learning*, pages 2292–2320. Pmlr, 2022.

[34] Hanle Zheng, Zhong Zheng, Rui Hu, Bo Xiao, Yujie Wu, Fangwen Yu, Xue Liu, Guoqi Li, and Lei Deng. Temporal dendritic heterogeneity incorporated with spiking neural networks for learning multi-timescale dynamics. *Nature Communications*, 15(1):277, 2024.

[35] Jan Koutník, Klaus Greff, Faustino Gomez, and Jürgen Schmidhuber. A clockwork RNN. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1863–1871. PMLR, 2014. URL https://proceedings.mlr.press/v32/koutnik14.html.

[36] Adam Santoro, Ryan Faulkner, David Raposo, Jack W. Rae, Mike Chrzanowski, Theophane Weber, Daan Wierstra, Oriol Vinyals, Razvan Pascanu, and Timothy P. Lillicrap. Relational recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pages 7310–7321, 2018. URL https://papers.nips.cc/paper/7960-relational-recurrent-neural-networks.

[37] Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=mLcmdlEUxy-.

[38] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

[39] Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

[40] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *International conference on learning representations*, volume 2025, pages 70367–70413, 2025.

[41] Luke Darlow, Ciaran Regan, Sebastian Risi, Jeffrey Seely, and Llion Jones. Continuous thought machines. *Advances in Neural Information Processing Systems*, 38:21548–21594, 2026.

[42] Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md. Mostofa Ali Patwary, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017. doi: 10.48550/arXiv.1712.00409.

[43] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020. doi: 10.48550/arXiv.2010.14701.

[44] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[45] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 924–932, 2021.

[46] Ibrahim M Alabdulmohsin, Xiaohua Zhai, Alexander Kolesnikov, and Lucas Beyer. Getting vit in shape: Scaling laws for compute-optimal model design. *Advances in Neural Information Processing Systems*, 36: 16406–16425, 2023.

[47] Noah Amsel, Gilad Yehudai, and Joan Bruna. Quality over quantity in attention layers: When adding more heads hurts. In *International Conference on Learning Representations*, volume 2025, pages 62431–62483, 2025.

[48] Igor Dubinin, Antonio Orvieto, and Felix Effenberger. Improved state mixing in higher-order and block diagonal linear recurrent networks. *arXiv preprint arXiv:2602.12021*, 2026.

[49] Nadezhda Chirkova, Ekaterina Lobacheva, and Dmitry Vetrov. Deep ensembles on a fixed memory budget: One wide network or several thinner ones? *arXiv preprint arXiv:2005.07292*, 2020.

[50] Ekaterina Lobacheva, Nadezhda Chirkova, Maxim Kodryan, and Dmitry P Vetrov. On power laws in deep ensembles. *Advances In Neural Information Processing Systems*, 33:2375–2385, 2020.

[51] Jan Ludziejewski, Jakub Krajewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, et al. Scaling laws for fine-grained mixture of experts. In *Forty-first International Conference on Machine Learning*, 2024.

[52] Andrew R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, 1993. doi: 10.1109/18.256500.

[53] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314, 1989. doi: 10.1007/BF02551274.

[54] Guido F. Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems*, volume 27, 2014.

[55] Matus Telgarsky. Benefits of depth in neural networks. In *Proceedings of the 29th Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 1517–1539. PMLR, 2016. URL https://proceedings.mlr.press/v49/telgarsky16.html.

[56] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2847–2854. PMLR, 2017. URL https://proceedings.mlr.press/v70/raghu17a.html.

[57] Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3): 379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[58] Horace B. Barlow. Possible principles underlying the transformations of sensory messages. In Walter A. Rosenblith, editor, *Sensory Communication*, pages 217–234. MIT Press, Cambridge, MA, 1961.

[59] Fred Attneave. Some informational aspects of visual perception. *Psychological Review*, 61(3):183–193, 1954. doi: 10.1037/h0054663.

[60] Simon B. Laughlin. A simple coding procedure enhances a neuron's information capacity. *Zeitschrift für Naturforschung C*, 36(9–10):910–912, 1981. doi: 10.1515/znc-1981-9-1040.

[61] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. doi: 10.1038/381607a0.

[62] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

[63] Larry F. Abbott and Peter Dayan. The effect of correlated variability on the accuracy of a population code. *Neural Computation*, 11(1):91–101, 1999. doi: 10.1162/089976699300016827.

[64] Bruno B. Averbeck, Peter E. Latham, and Alexandre Pouget. Neural correlations, population coding and computation. *Nature Reviews Neuroscience*, 7(5):358–366, 2006. doi: 10.1038/nrn1888.

[65] Rubén Moreno-Bote, Jeffrey Beck, Ingmar Kanitscheider, Xaq Pitkow, Peter Latham, and Alexandre Pouget. Information-limiting correlations. *Nature Neuroscience*, 17(10):1410–1417, 2014. doi: 10.1038/nn.3807.

[66] Gašper Tkačik, Jason S Prentice, Vijay Balasubramanian, and Elad Schneidman. Optimal population coding by noisy spiking neurons. *Proceedings of the National Academy of Sciences*, 107(32):14419–14424, 2010.

[67] Eizaburo Doi and Michael S Lewicki. A simple model of optimal population coding for sensory systems. *PLoS computational biology*, 10(8):e1003761, 2014.

[68] Deep Ganguli and Eero P Simoncelli. Efficient sensory encoding and bayesian inference with heterogeneous neural populations. *Neural computation*, 26(10):2103–2134, 2014.

[69] Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

[70] Matt Mahoney. Large text compression benchmark, 2011. URL `http://www.mattmahoney.net/dc/text.html`.

[71] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 2978–2988, 2019.

[72] William Bialek, Fred Rieke, Robert van Steveninck, and David Warland. Reading a neural code. *Advances in neural information processing systems*, 2, 1989.

[73] Alexander Borst and Frédéric E Theunissen. Information theory and neural coding. *Nature neuroscience*, 2(11):947–957, 1999.

[74] Alexander Maloney, Daniel A Roberts, and James Sully. A solvable model of neural scaling laws. *arXiv preprint arXiv:2210.16859*, 2022.

[75] Julian Rossbroich, Julia Gygax, and Friedemann Zenke. Fluctuation-driven initialization for spiking neural network training. *Neuromorphic Computing and Engineering*, 2(4):044016, 2022.

[76] Alexandre Bittar and Philip N Garner. A surrogate gradient spiking baseline for speech command recognition. *Frontiers in Neuroscience*, 16:865897, 2022.

[77] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=SylKikSYDH`.

[78] Etay Hay, Sean Hill, Felix Schürmann, Henry Markram, and Idan Segev. Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS computational biology*, 7(7):e1002107, 2011.

[79] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

A Architecture, Training, Dataset and Analysis Details

Software libraries and packages. All numerical experiments were performed in Python 3.12.10 using JAX 0.4.35 with GPU-accelerated JAXlib 0.4.35, leveraging Equinox 0.11.10 for model construction, Optax 0.2.3 for optimization, and PyTorch 2.5.1 for data loading. Statistical analysis and curve fitting used NumPy 2.2.6 and SciPy 1.14.1. The accompanying code repository can be found at `https://github.com/AaronSpieler/elmnetwork`.

Compute infrastructure and runtimes. Accelerated computations were performed on up to four A100 40GB GPUs on shared compute clusters, primarily the Raven cluster at MPCDF, with individual runs taking no longer than 24h, typically taking 2-4h on SHD-Adding, 5-15h on Enwik8, and 8-12h on NeuronIO on a single GPU. Scaling the reference ELM Network on Enwik8 beyond $d_m = 35$, $N_{\text{rec}} = 3072$ or $d_{\text{branch}} = 50$ required more than one GPU to fit in VRAM and ran longer. Approximately 75k GPU hours were spent throughout the project, a vast majority of which was spent developing the architecture ($\sim 50\%$) and on exploratory or scaling experiments ($\sim 30\%$).

Table 1: Reference model configurations for all three benchmarks.

Parameter	Notation	SHD-Adding	Enwik8	NeuronIO
Neuron				
Memory units	d_m	5	15	20
MLP hidden layers	l_{mlp}	1	1	1
MLP hidden width	d_{mlp}	$2d_m$	$2d_m$	$2d_m$
Number of branches	d_{tree}	30	50	45
Synapses per branch	d_{branch}	10	15	100
Total synapses	d_s	300	750	4500
Input scale	c	10	10	1
Timescale ratio	λ	5	5	5
Memory timescales	$\tau_{\min}$, $\tau_{\max}$	1, 500	0.1, 100	1, 150
High-pass filter	τ_r	5	2	—
Layer				
Recurrent neurons	N_{rec}	96	1024	—
Recurrence fraction	ρ_{rec}	0.25	0.8	—
Readout neurons	—	19	204	—
Input embedding	—	—	one-hot	—

Modifications to the ELM neuron in [1]: this work builds on the "Branch-ELM" variant with the improved memory update, with fixed memory timescales τ_m initialized equidistant in log-space, and disabled synapse current decay using $\tau_s = 0$. The implementation adds a high pass filter on the memory readout with EMA timescale τ_r for robust and stable training, uses a ReLU neuron output activation with learnable bias b, scales synaptic inputs using c which accelerates early training, uses default weight initialization per branch for w_s too, swaps $ReLU^2$ in for MLP hidden layer activation for performance, and adds a small L2 regularization term on the MLP output for improved trainability.

A.1 Robustness of recurrent networks with high-pass filtered neurons

While successfully training ELM Networks to similar performance without a high-pass filter mechanism *is possible*, its addition to ELM neurons is sufficient to make the initialization and training of this doubly-recurrent architecture remarkably robust with respect to orders of magnitude of parameter changes in neuron complexity and network layer width, and results in stable dynamics throughout.

Typically, such neuroscience-inspired architectures are highly sensitive to training and initialization, struggling with exploding or vanishing activity, requiring careful initialization and regularization [75]. Yet with this simple high-pass filter modification, standard deep learning weight initialization becomes viable, and even grossly misconfigured architectures may become simple to train.

There are two core mechanisms why it works, one addressing exploding activity, and one addressing vanishing activity; **1)** with sufficiently small τ_r the slow oscillations that typically blow up

network activity through recurrence are completely eliminated by construction (only high frequency oscillations pass the filter), **2)** neuron and network biases that would typically keep activity levels below threshold are removed entirely, as the filtering will always result in half the signal above 0 as long as the hidden state shows *any* variance at all, which translates into 50% network activity with a 0 activity-threshold initialization by construction (any bias removed through centering) (see Fig. 8).

Choosing a good τ_r is simple, as networks remain stable for a wide range of timescales (see Fig. 11). We recommend a default of $\tau_r = 5$ and sweeping the range [2,20]. On tasks like SHD-Adding, which have particularly sparse inputs yet long-range dependencies, choosing a larger τ_r can noticeably improve performance. Note that ELM neurons can still take advantage of orders of magnitude larger memory timescales τ_m than their high-pass filter timescale τ_r, as in the case of SHD-Adding with 500 vs 5.

A.2 Datasets and corresponding training details

Dataset and code availability. SHD [25] is available from Zenke Lab at `https://zenkelab.org/resources/spiking-heidelberg-datasets-shd/` under the Creative Commons Attribution 4.0 International License (CC BY 4.0). NeuronIO [19] is available from David Beniaguev on Kaggle at `https://www.kaggle.com/datasets/selfishgene/single-neurons-as-deep-nets-nmda-train-data` under CC BY-SA 4.0. For SHD-Adding [1] and NeuronIO [19], we use the dataloaders provided by Spieler et al. at `https://github.com/AaronSpieler/elmneuron`, released under the MIT License; the SHD-Adding dataloader ingests SHD data. Enwik8 [70] is available from Matt Mahoney at `http://mattmahoney.net/dc/enwik8.zip`; it consists of the first 10^8 bytes of the March 3, 2006 English Wikipedia dump and is licensed under the GNU Free Documentation License (GFDL). We preprocess and load Enwik8 using the standard Transformer-XL pipeline [71] provided by Dai et al. at `https://github.com/kimiyoung/transformer-xl`, released under the Apache License 2.0.

Evaluations on the SHD-Adding dataset. The SHD-Adding benchmark is based on the Spiking Heidelberg Digits dataset [25], and was introduced in [1]. Each sample pairs two spike-encoded spoken digits (0–9, German or English) end-to-end, and the model must predict their sum (19 classes, chance $\approx 5.3\%$). The 700-channel binary spike trains are trimmed to one second per digit and discretized into 2 ms bins, producing $2 \times 500 = 1000$ time steps. The network is optimized with cross-entropy on its final output. SuperSpike surrogate gradient used a scale of 100. A 20% validation split guides model selection, and test accuracy (mean $\pm$ std) is reported over three seeds. Where reported, reducible error denotes the mean test error minus a floor of $1 - 0.94^2 \approx 0.116$, based on a reference single-digit SHD accuracy of 0.94 [76].

Table 2: Training configurations for all three benchmarks.

Setting	SHD-Adding	Enwik8	NeuronIO
Optimization			
Optimizer	Adamax	Adam	Adam
Learning rate	5e-3	5e-4	5e-4
LR schedule	cosine	cosine	cosine
Warmup steps	400	100	0
Gradient clip norm	1.0	1.0	1.0
Batch size	8	8	8
Epochs / turns	70	750	35
Steps per epoch / turn	2000	400	10000
Label smoothing	1e-2	1e-4	—
Regularization			
Input dropout	0.2	0.0	—
Recurrent dropout	0.2	0.0	—
MLP output L2	0.002	0.01	—
Neuron activity L1	0.2	1.0	—

Evaluations on the Enwik8 dataset. Enwik8 [70] is preprocessed following the standard Transformer-XL pipeline [71]. This yields 204 unique byte tokens, which we encode as one-hot vectors scaled by 3. The network processes sequences of length 100 and maintains context across consecutive batches through hidden state reuse: the recurrent hidden state is carried forward from one batch to the next, with a reset probability that decays from 1.0 to 0.01 over the first 40k training steps following a cosine schedule. At test time, evaluation proceeds sequentially with batch size 1 to ensure a single unbroken hidden state across the full test set. Models are trained using BPTT and cross-entropy loss; performance is reported as test bits-per-character (BPC $=$ loss $/\ln 2$), and the model with the lowest validation loss is selected for final evaluation. Where reported, reducible BPC denotes the remaining test BPC above the reference floor of 0.97 [77].

Evaluations on the NeuronIO dataset. The NeuronIO dataset [19] provides simulated input-output data of a detailed biophysical layer 5 cortical pyramidal neuron model [78], with 1278 binary pre-synaptic spike channels as input and somatic membrane voltage and output spikes as targets. Following [19], somatic voltage is capped at -55 mV, offset by -67.7 mV, and scaled by $1/10$. Each training sample spans 500 ms, with the first 150 ms excluded from the loss as burn-in. The model is a single ELM neuron trained using BPTT with equally weighted binary cross-entropy (spikes) and mean squared error (voltage). The model with the lowest validation RMSE is selected, and test voltage prediction performance is reported as reducible MAE (mean $\pm$ std) over three seeds. Where reported, reducible MAE denotes the test MAE minus a floor of 0.319 mV, the average achieved by the largest model in our sweep ($d_m = 100$, AUC $= 0.9938$).

Fixed parameter budget experiments. In case exactly matching the budget was not possible, the neuron count was rounded down for the N vs k_e sweeps, and the number of synapses per branch were rounded down for the k_e vs k_c experiments.

Activation regularization. Two regularization terms are applied to the recurrent hidden layer. A per-neuron L2 penalty on the time-averaged absolute MLP output keeps individual neurons' memory updates away from $\tanh$ saturation, promoting gradient flow; this is particularly helpful for neurons with large d_m. A population-level L1 penalty on the mean neuron activity encourages sparse firing, particularly beneficial for wide layers. An ablation of these activity regularizers can be found in Appendix Fig. 13. Strengths are listed in Table 2. Neither term is used for NeuronIO.

A.3 Structured joint hyperparameter search experiment details

Table 3: Search space of the structured joint hyperparameter experiment.

Setting	SHD-Adding	Enwik8
Swept settings		
N_{rec}	{32, 64, 128}	{512, 1024, 1536}
d_m	{3, 5, 10, 15}	{5, 10, 15, 20}
d_{mlp}	$\{2d_m, 3d_m\}$	$\{2d_m, 3d_m, \lfloor 10\sqrt{d_m} \rfloor\}$
d_{tree}	$\{d_{\text{mlp}}, 2d_{\text{mlp}}\}$	$\{d_{\text{mlp}}, 2d_{\text{mlp}}\}$
d_{branch}	$\{d_{\text{tree}}/2, d_{\text{tree}}\}$	$\{d_{\text{tree}}, 2d_{\text{tree}}\}$
Pareto candidate		
N_{rec}	{32, 64, 128, 256}	{256, 512, 768, 1024, 1280}
d_m	$\lfloor \frac{1}{2}\sqrt{d_{\text{inp}}/15 + N_{\text{rec}}} \rfloor$	$\lceil \frac{1}{2}\sqrt{d_{\text{inp}} + N_{\text{rec}}} \rceil$
$d_{\text{mlp}},\ d_{\text{tree}},\ d_{\text{branch}}$	$2d_m,\ 2d_{\text{mlp}},\ d_{\text{tree}}$	$2d_m,\ 2d_{\text{mlp}},\ d_{\text{tree}}$

Additional configuration details. The recurrence fraction is set to $\rho_{\text{rec}} = \sqrt{N_{\text{rec}}/(N_{\text{rec}} + d_{\text{inp}})}$ on SHD-Adding and $\rho_{\text{rec}} = N_{\text{rec}}/(N_{\text{rec}} + d_{\text{inp}})$ on Enwik8. On Enwik8, the readout layer neurons match the hidden layer in complexity and connectivity. On SHD-Adding, the readout layer connectivity is instead adjusted proportionally to N_{rec} with $d_{\text{tree}} = d_{\text{branch}} \in \{10, 15, 20\}$. Regularization is switched to per-neuron scaling, with strengths (MLP L2 at 10^{-5}, activity L1 at 10^{-3}) selected to match the reference configuration's per neuron regularization strength at $N_{\text{rec}} = 1024$; on Enwik8 the MLP L2 term is applied to both ELM layers. The Pareto Configurations show mean and standard deviation

across five seeds, otherwise, SHD-Adding results show mean across three seeds, whereas Enwik8 shows individual runs. Training otherwise follows the reference setup (Table 2).

A.4 Theoretical framework experiments and data analysis

Three analyses connect the theoretical proxy (Eq. 3) to experimental observations. All curve fitting is performed by minimizing squared residuals in log-space via nonlinear least squares, with model selection by corrected Akaike Information Criterion (AICc).

Measuring α. The per-neuron expressivity exponent α is estimated by fitting a power law to reducible error as a function of per-neuron parameters $k_e = \#\mathbf{w}_p + \#\mathbf{w}_r$. On NeuronIO ($d_m \in [2, 50]$), the metric is MAE $\propto \sigma_n \propto k_e^{-\alpha/2}$, where the halved exponent arises because MAE scales with the noise standard deviation rather than its variance (Fig. 5c, 17). On Enwik8 ($d_m \in [5, 35]$, $N_{\text{rec}} = 1024$), α is recovered directly as the log-log slope of reducible BPC versus k_e, since in the low-SNR regime ($s \approx 1$, $d_m \leq 35$) the spectral exponent β affects only the level of I_{rep}, not its scaling with k_e (Appendix B.5; Fig. 6g, Fig. 20). Therefore, α slopes are best measured in the low-SNR regime, and k_e sweeps should be truncated before the log bends or noise floor is reached. Measuring α beyond the low-SNR regime should be possible but would require a separate treatment and, in general, require estimating β first.

Measuring β. The spectral decay exponent β is estimated from the eigenvalue spectrum of the sample covariance of the memory readout $\mathbf{w}_r^\top \mathbf{m}_t$, centered across 50 batch trajectories of 512 time steps after discarding a 128-step burn-in, by fitting a truncated power law $\lambda_i = \sigma_f^2 \, i^{-\beta} \exp[-(i/i_c)^\nu]$ with shared cutoff (i_c, ν) across different connectivity configurations (Fig. 6h, 19). The memory readout was chosen as it represents the neuron output, which also contains the task-relevant slow components, more in line with the whole test set BPC evaluations. Measuring after the high-pass filter removes those slower components and artificially biases measurements towards significantly smaller β.

Joint theory fit. The joint theory fit maps $-I_{\text{rep}}$ to test BPC via a shared affine transformation across all seven Enwik8 parameter sweeps, using derivative-free global optimization (differential evolution) to minimize the sum of squared BPC residuals over eight free theory parameters (Fig. 6a–f), with details in Fig. 16. Some large k_e configurations for performance degrading parameter setting $\tau_{\max} = 1.0$ and d_s / $50 = 7$ displayed unusually large deviations in performance compared to their neighboring configurations, and were rerun.

B Derivation of the Effective Representation Information

We derive the formula for the effective representation information,

$$I_{\mathrm{rep}}(k_e) \;=\; \frac{1}{2} \sum_{i=1}^{P/(k_e+k_c)} \log_2\!\big(1 + s(k_e)\cdot i^{-\beta}\big)\,, \tag{4}$$

from the mutual information of a Gaussian vector channel, stating all assumptions. Under assumptions **(A1)–(A4)**, the derivation in this section yields the expression for $I_{\mathrm{rep}}(k_e)$ exactly.

B.1 Assumptions

(A1) Gaussian channel. The layer output $\mathbf{y} \in \mathbb{R}^N$ decomposes as $\mathbf{y} = \mathbf{f}(\mathbf{x}) + \mathbf{n}$, where $\mathbf{x} \in \mathbb{R}^D$ is the input, $\mathbf{f} : \mathbb{R}^D \to \mathbb{R}^N$ maps inputs to the task-relevant signal component, and $\mathbf{n} \sim \mathcal{N}(\mathbf{0},\, \sigma_n^2 \mathbf{I}_N)$ is independent additive Gaussian noise. The signal is modeled as Gaussian: $\mathbf{f}(\mathbf{x}) \sim \mathcal{N}(\mathbf{0},\, \mathbf{C}_f)$, with positive-definite signal covariance $\mathbf{C}_f \in \mathbb{R}^{N\times N}$ matching the empirical second moments of the learned representation.

(A2) Power-law signal spectrum. The signal covariance $\mathbf{C}_f$ has N ordered eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N > 0$ that follow a power law in the rank index: $\lambda_i = \sigma_f^2\, i^{-\beta}$, where the leading signal variance $\sigma_f^2 \equiv \lambda_1$ sets the scale and $\beta > 0$ is the spectral decay exponent.

(A3) Phenomenological residual noise model. The total steady-state noise variance per neuron at per-neuron effective parameter count k_e is modelled as

$$\sigma_n^2(k_e) \;=\; \sigma_f^2\, q(k_e), \qquad q(k_e) \;=\; \max\!\big((\gamma k_e)^{-\alpha},\, q_\infty\big)\,,$$

where $\gamma > 0$ is an effectivity constant, $\alpha > 0$ is the expressivity exponent, and $q_\infty > 0$ is an irreducible normalised residual floor set by the neuron's binding bottleneck — most prominently its scalar output mechanism. Equivalently, the noise-to-signal variance ratio is assumed to decay as a power law in k_e until pinned at this floor. This is adopted as a phenomenological ansatz for the effective residual mismatch of a finite-budget neuron.

(A4) Fixed parameter budget. The total budget P is divided equally among N neurons, each receiving $P/N = k_e + k_c$ parameters: $k_e > 0$ effective parameters driving expressivity and a connectivity overhead $k_c \geq 0$ that does not directly reduce approximation error.

B.2 Terminology

We evaluate the mutual information between the signal component and the layer's noisy output under the empirical data distribution. We formalise this quantity as the *effective representation information*, $I_{\mathrm{rep}}(k_e)$. Unlike Shannon channel capacity—which requires maximising over all possible input distributions—$I_{\mathrm{rep}}(k_e)$ measures the information throughput achieved by the assumed representation channel under the network's architectural and finite-budget constraints.

B.3 Derivation

Step 1 (Mutual information of a Gaussian channel). Under **(A1)**, $I(\mathbf{f};\mathbf{y}) = h(\mathbf{y}) - h(\mathbf{y}|\mathbf{f}) = h(\mathbf{y}) - h(\mathbf{n})$. For Gaussian vectors, differential entropy is $\frac{1}{2}\log_2[(2\pi e)^N \det(\mathbf{\Sigma})]$; the prefactors cancel in the difference, and the ratio of determinants gives:

$$I(\mathbf{f};\,\mathbf{y}) \;=\; \frac{1}{2}\log_2 \det\!\big(\mathbf{I}_N + \sigma_n^{-2}\,\mathbf{C}_f\big)\,. \tag{5}$$

*Exact given **(A1)**.* □

Since the Gaussian distribution maximizes differential entropy for a fixed covariance, $h(\mathbf{y}) \geq h(\mathbf{y}_{\mathrm{true}})$ at matched second moments while $h(\mathbf{n})$ remains unchanged. Equation (5) therefore serves as an *upper bound* on the mutual information of any *additive channel with the same signal-independent noise distribution and the same output covariance*.

Step 2 (MIMO mutual information under power-law spectrum). Under **(A2)**, the matrix $\sigma_n^{-2}\,\mathbf{C}_f$ has eigenvalues $s\,i^{-\beta}$ for $i = 1, \ldots, N$, where $s = \sigma_f^2/\sigma_n^2$ is the leading-mode signal-to-noise ratio. Using $\det(\mathbf{I} + \mathbf{A}) = \prod_i (1 + \lambda_i(\mathbf{A}))$, equation (5) becomes

$$I(\mathbf{f};\, \mathbf{y}) \;=\; \frac{1}{2} \sum_{i=1}^{N} \log_2\!\big(1 + s\, i^{-\beta}\big)\,. \tag{6}$$

Each eigenmode contributes independently: mode i carries $\frac{1}{2}\log_2(1 + s\,i^{-\beta})$ bits, with an effective per-mode signal-to-noise ratio of $s\,i^{-\beta}$ decreasing as a power law in the mode index.

This spectral decay provides a natural soft cutoff on the number of informative dimensions. The effective dimensionality of the representation is governed by the stable rank of the signal covariance, $N_{\text{eff}} = \sum_{i=1}^{N} i^{-\beta}$. Depending on the exponent β, N_{eff} either saturates ($\beta > 1$), grows logarithmically ($\beta = 1$), or grows sublinearly ($\beta < 1$). Consequently, high-order modes ($i \gg s^{1/\beta}$) contribute negligibly to the mutual information.

*Exact given **(A1), (A2)**.* □

Step 3 (Per-neuron SNR from the phenomenological residual law). Under **(A3)**, the total noise variance at effective budget k_e is

$$\sigma_n^2(k_e) \;=\; \sigma_f^2 \cdot \max\!\big((\gamma k_e)^{-\alpha},\; q_\infty\big)\,.$$

The leading-mode signal-to-noise ratio (SNR) is therefore

$$s(k_e) \;=\; \frac{\sigma_f^2}{\sigma_n^2(k_e)} \;=\; \min\!\big((\gamma k_e)^{\alpha},\; q_\infty^{-1}\big)\,. \tag{7}$$

This is a power law in k_e capped at a hard ceiling. It satisfies $s(k_e) = (\gamma k_e)^{\alpha}$ for small k_e (parametric regime) and $s(k_e) = q_\infty^{-1}$ above the crossover threshold $k_e^* = \gamma^{-1}\, q_\infty^{-1/\alpha}$ (floor regime), where additional effective parameters provide no further increase in per-neuron SNR. Mode i of the signal covariance thus experiences an effective SNR of $s(k_e)\,i^{-\beta}$.

*Exact given **(A3)**.* □

Step 4 (Assembly). Substituting the layer width $N = P/(k_e + k_c)$ from **(A4)** and the SNR mapping $s = s(k_e)$ from (7) into equation (6) yields:

$$\boxed{I_{\text{rep}}(k_e) \;=\; \frac{1}{2} \sum_{i=1}^{P/(k_e+k_c)} \log_2\!\big(1 \;+\; \min\!\big((\gamma k_e)^{\alpha},\; q_\infty^{-1}\big) \cdot i^{-\beta}\big)} \tag{8}$$

which completes the derivation.

*Exact given **(A1)–(A4)**.* □

B.4 Crossing of curves for varied α experiments

For a fixed total parameter budget P, varying the per-neuron budget k_e also changes the layer width according to $N = P/(k_e + k_c)$. The dependence of the I_{rep} curve on the expressivity exponent α enters through the $s(k_e) = (\gamma k_e)^\alpha$ before the neuron noise floor. Since $1^\alpha = 1$ for all α, curves obtained by varying α will intersect where $\gamma k_e = 1$ (see Fig. 6b), specifically at

$$k_{\text{cross}} = \gamma^{-1}, \qquad N_{\text{cross}} = \frac{P}{\gamma^{-1} + k_c}.$$

which permits direct estimation of γ by identification of the crossing point k_{cross}. A reduced γ would shift the crossing to larger per-neuron budgets k_{cross} and, at fixed P, to smaller layer widths N_{cross}.

B.5 Scaling of I_{rep} with per-neuron budget

We characterize how I_{rep} scales with per-neuron complexity k_e for a fixed layer size N, specifically when $s(k_e)$ is of order unity and pre noise floor saturation (the regime realized by most of our experiments). All k_e-dependence in equation (8) enters through the per-neuron SNR $s(k_e) = (\gamma k_e)^\alpha$, as $N = P/(k_e + k_c)$ is effectively k_e independent due to the likewise growing budget P.

When s is of order unity, the per-mode SNRs $s \cdot i^{-\beta}$ are small for all but the leading modes, and $\log_2(1 + x) \approx x/\ln 2$ applies to the bulk of the sum, giving:

$$I_{\text{rep}} \;\approx\; \frac{(\gamma k_e)^\alpha}{2 \ln 2} \sum_{i=1}^{N} i^{-\beta} \;\propto\; k_e^\alpha \tag{9}$$

The log-log slope is therefore $\eta \approx \alpha$. The spectral exponent β enters only through the mode sum prefactor, setting the level of I_{rep} but not its scaling with k_e. This can be used to measure α directly from network layer performance, as was done in Fig. 6g.

B.6 Single-neuron noise-floor saturation

The $I_{\text{rep}}(k_e)$ expression (Eq. (8) implies two regimes for the neuron-level SNR: an increasing regime for smaller per-neuron budgets k_e and a noise-floor-limited regime once $(\gamma k_e)^\alpha$ reaches the value $1/q_\infty$, as seen from the expression for $s(k_e)$:

$$s(k_e) = \min\left((\gamma k_e)^\alpha, \frac{1}{q_\infty}\right).$$

The transition between these regimes occurs where the two branches are equal and is empirically observable as a knee in a fixed-layer-width neuron-complexity scaling curve (see Fig. 20). Solving for q_∞ at the transition point k_{sat} gives

$$q_\infty = (\gamma k_{\text{sat}})^{-\alpha}.$$

which permits direct estimation of the single-neuron noise-floor parameter using independently determined values of α and γ.

C Additional Results and Supporting Evidence

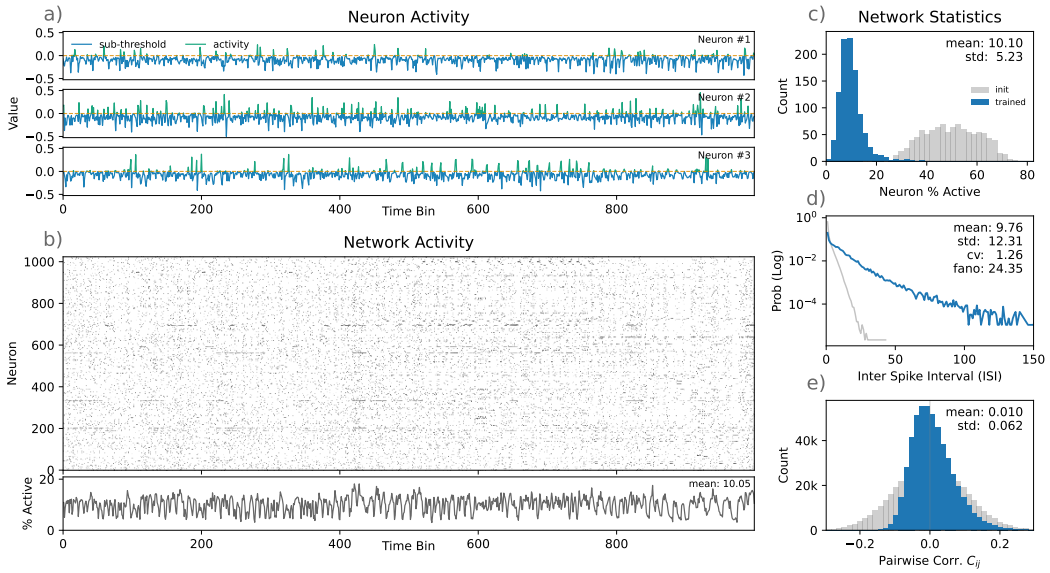


Figure 8: **Training drives ELM Networks into a sparse, mostly asynchronous, irregular activity regime.** Example inference on Enwik8 of a reference model configuration with $N = 1024$ and $d_m = 15$. **a, b)** Individual neurons' activity is characterized by brief spike-like above-threshold activations and high-frequency sub-threshold fluctuations. At the population level, $\sim$10% of neurons are active at any given time, firing asynchronously with loose global synchronization and oscillations. **c)** Training sparsifies the population activity from $\sim$50% to $\sim$10% active (with heavy tail); while a small L1 activity regularization accentuates this trend, sparsification happens regardless. **d)** Inter-spike intervals are approximately exponentially distributed (CV $\approx$ 1.3, Fano $\approx$ 24), indicating irregular bursty firing. **e)** Pairwise correlations are tightly centered near zero, and get smaller with training, indicating training decorrelating neuron activations.

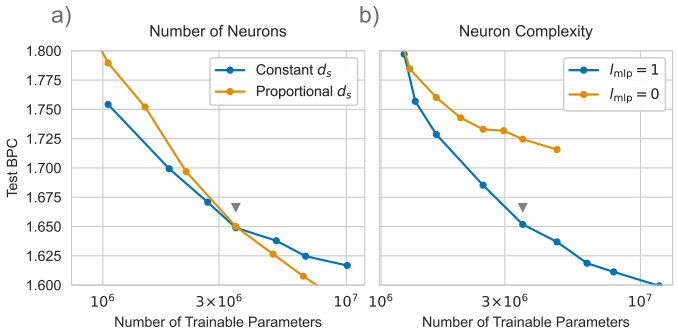


Figure 9: **Proportional connectivity and deeper dendritic integration scale better even accounting for additional parameter cost:** experimental sweep across number of neurons and neuron complexity on Enwik8 matching Figure 4a,b, with x-axis plotting in terms of total network trainable parameters. While curves rescale, the same trend emerges; **a)** scaling with number of neurons works better with proportional neuron connectivity, and **b)** scaling with neuron complexity with hierarchical synaptic integration beats simpler integration by a large margin.

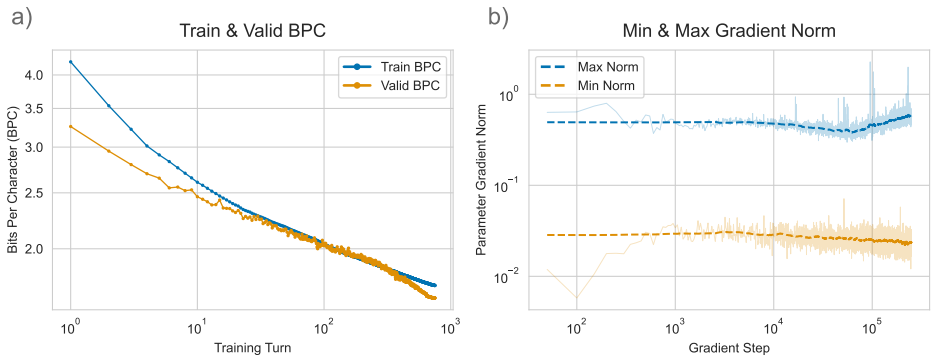


Figure 10: **ELM Network training is smooth and gradients remain stable throughout:** Training dynamics of the reference run with $N{=}1024$ and $d_m{=}15$ in Figures 6. **a)** Train and valid BPC over 750 training turns, converging near 1.644 valid BPC slightly below test BPC of 1.65. **b)** Min and max parameter gradient norms, logged every 50th gradient step for 5000 samples, remain stable throughout training with no signs of exploding or vanishing gradients.

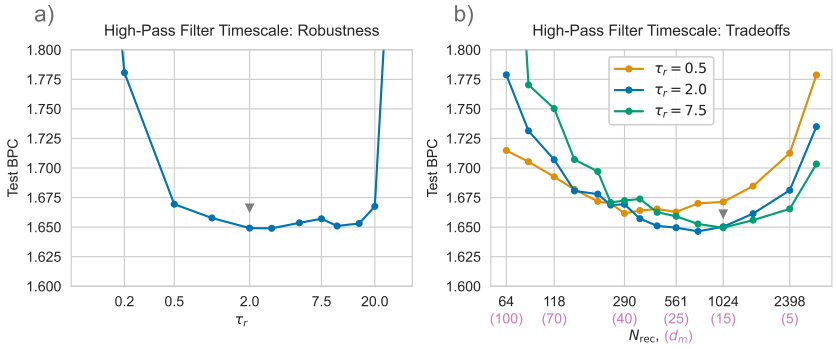


Figure 11: **A wide range of high-pass filter timescales stabilize network dynamics and result in the same qualitative tradeoffs:** Sensitivity analysis of τ_r on the Enwik8 dataset. **a)** For the reference architecture with $N{=}1024$ and $d_m{=}15$ training remains stable and yields similar performance over an order of magnitude in τ_r. Training runs with too large τ_r become unstable, while ones with too small timescale remove all signal from neuron output. **b)** The trade-off between number and complexity of neurons for the reference parameter budget of $P = 3.29 \cdot 10^6$ yields the characteristic interior optimum irrespective of the tested timescales. However, large neuron configurations benefit from fast readout timescales, whereas small neuron configurations from slower ones, thereby shifting the exact location of the optimum.

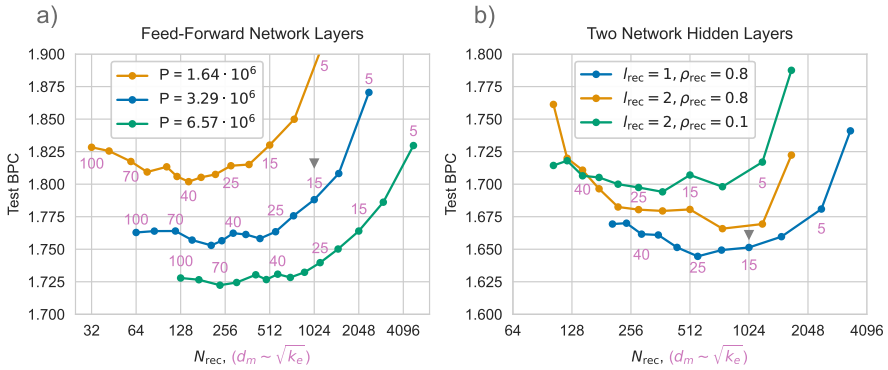


Figure 12: **The qualitative trade-off between number of neurons and complexity also persists for feed-forward and two-hidden-layered networks:** We evaluate the N vs k_e tradeoff on Enwik8 for purely feed-forward ELM-Network with $\rho_{\text{rec}} = 0.0$ and networks with an additional hidden layer but same parameter budget. Note that individual ELM Neurons remain internally recurrent irrespective of ρ_{rec}, and for parameter matched networks with twice the hidden layers the per-hidden-layer budget is halved. **a)** We likewise observe the emergence of an interior optimum, that shift towards more and more complex neurons with increasing budget like in Figure 4. Overall network performance is degrading substantially ($\sim$0.1 test BPC) and the optimal neuron complexity has increased significantly, yet the optimum is not as sharp anymore as before. The network possibly relies more on individual neurons extracting meaningful features on their own, unable to build on recurrent features. **b)** The neuron complexity vs neuron count tradeoff is preserved, with the optimum shifting towards smaller neuron configurations, as our theoretical proxy would predict for such network layers with half parameter budget. Performance also degrades overall (additional vertical shift), consistent with a reduced effectivity of the network parameters, and the optimum sharpened somewhat. Reducing recurrence results in the same qualitative change in performance as in single hidden layer networks.

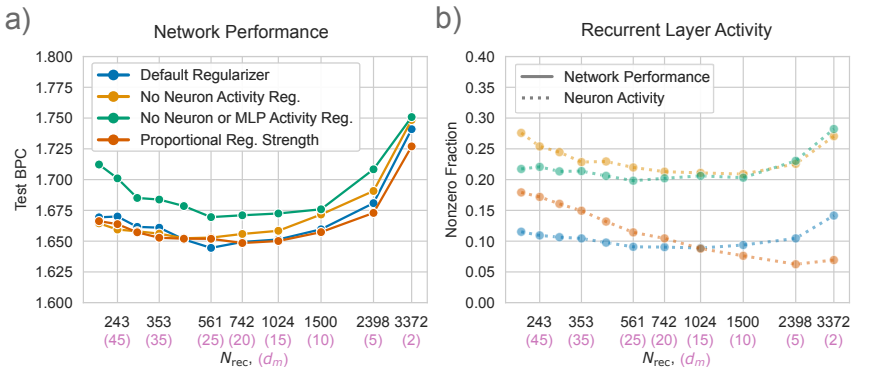


Figure 13: **The ELM-Network benefits from L2 neuron MLP output and L1 network activity regularization, without changing the qualitative tradeoffs:** Regularizer setup described in Appendix A.2. **a)** Compared to the default regularizer setup (blue curve), disabling the L1 regularizer on network activity (orange curve) slightly degrades performance for larger networks. Additionally disabling the L2 regularizer on the neurons MLP output more strongly degrades performance for particularly large neuron configurations. Scaling both regularizer strength automatically based on number of neurons, slightly improves performance at the parameter sweep extremes. **b)** The corresponding average recurrent layer activations measured across 2000 steps. Disabling the L1 neuron output regularizer roughly doubles network activity. Additionally disabling the L2 MLP output regularizer results in reduced activity levels for networks with large neuron configurations, as some neurons become unstable. Number of neuron proportional activity regularization improves performance across the board, now delivering consistent per neuron regularization strength across network configurations, further dropping network activity for wide networks while simultaneously improving performance.

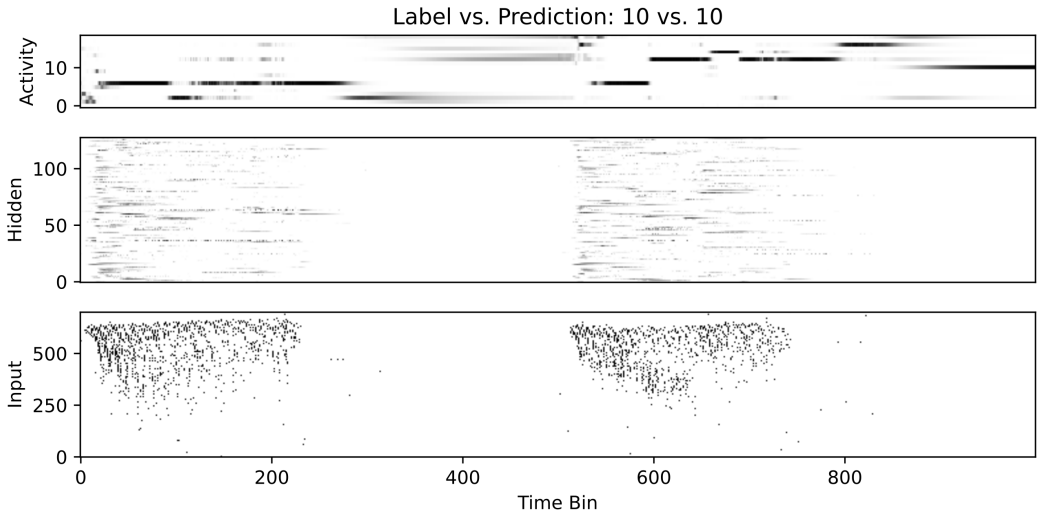


Figure 14: **The ELM-NET architecture exhibits rich temporal dynamics, characterized by periods of asynchronous irregular firing, synchronized bursts, and network silence:** An example network inference of an ELM Network with 128 neurons on SHD-Adding (see Appendix A.2). The networks' hidden layer displays short spike or burst like activity, with more active network phases visibly correlating to high input activity periods where individual digits were spoken. In between digit presentation the network falls mostly silent. The readout layer ELM neuron remains active throughout all periods, and displays frequent confident switching between different predictions; ultimately settling correctly.

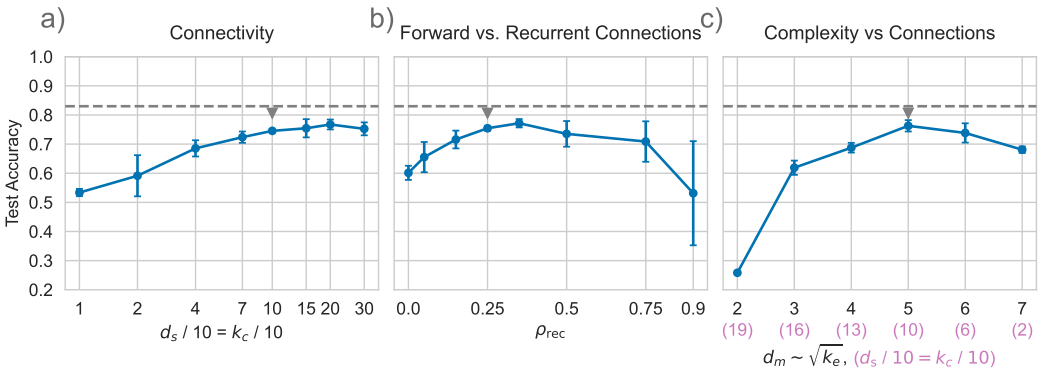


Figure 15: **Connectivity introduces the same qualitative tradeoffs on SHD-Adding as on Enwik8:** Estimated dataset noise floor at 83% marked with dashed line. **a)** Increasing synapses per branch d_{branch} improves performance with diminishing returns. **b)** An optimal recurrent fraction $\rho_{\text{rec}} \approx 0.30$ exists, which is roughly proportional to the ratio of recurrent to recurrent plus input connections. **c)** The tradeoff between neuron complexity and neuron connectivity shows an optimum for balanced configurations. These tradeoffs qualitatively mirror the Enwik8 findings (Fig. 4d–f).

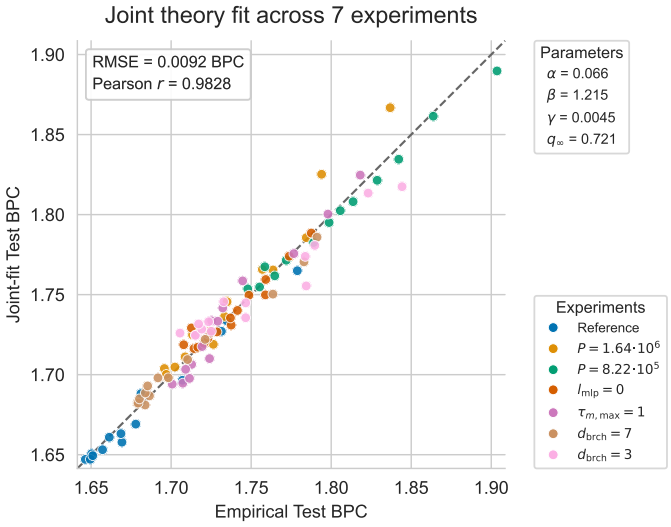


Figure 16: **Joint theory-experiment fit across seven experiments.** Each point is one (N_{rec}, d_m) configuration; the seven experiments span three parameter budgets and multiple architectural correspondences to the two key theory parameters α ($\tau_{m,\max}$, l_{mlp}) and β (d_{branch}). Eight theory parameters, the four shared reference values plus per-experiment versions for each varied quantity, and a single affine map are fit jointly to all 103 points by minimizing the sum of squared BPC using derivative-free optimization (differential evolution with L-BFGS-B polishing); reported α, β, γ, q_∞ refer to the reference architecture. A single parameter set captures both absolute BPC levels and the effects of architectural changes, supporting the view that theory parameters correspond to distinct identifiable architectural knobs. **Ablating** the core theory parameters α and β that control how individual and collective neuron performance scales and trade-off against each other degrades the quantitative fit substantially ($r = 0.7317$ and $r = 0.4542$). Removing q_{inf} or setting $\gamma = 1$ affects few points quantitatively ($r = 0.9797$ and $r = 0.9659$), but qualitatively degrade the fits by removing the convergence of curves for large k_e and crossing of curves for different α for small k_e configurations.

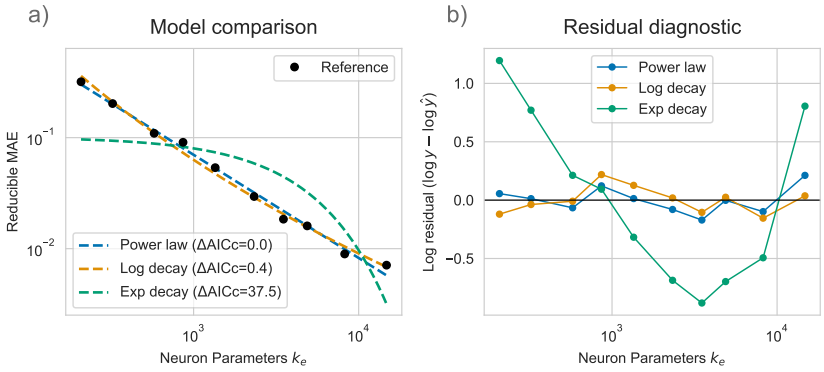


Figure 17: **A single neuron's reduction of approximation error is well described as a power law in parameter count.** Various sized ELM Neuron fit to NeuronIO with reducible MAE reported as mean over three seeds. Decay curves fitted in log-log. **a)** The reduction in error is compatible with a power law decay across two orders of magnitude in per-neuron parameters k_e, strongly preferred over exponential-decay alternatives ($\Delta\text{AICc} = 37.5$), however, logarithmic decay cannot be ruled out ($\Delta\text{AICc} = 0.4$). **b)** The unstructured residuals of the power law fit (compared to structured "U" shape of exponential fit) confirm the corresponding fit is approximately unbiased across the sweep. This finding motivates the per-neuron residual noise law $\sigma_n^2(k_e) \propto k_e^{-\alpha}$ used in the theory (Appendix B).

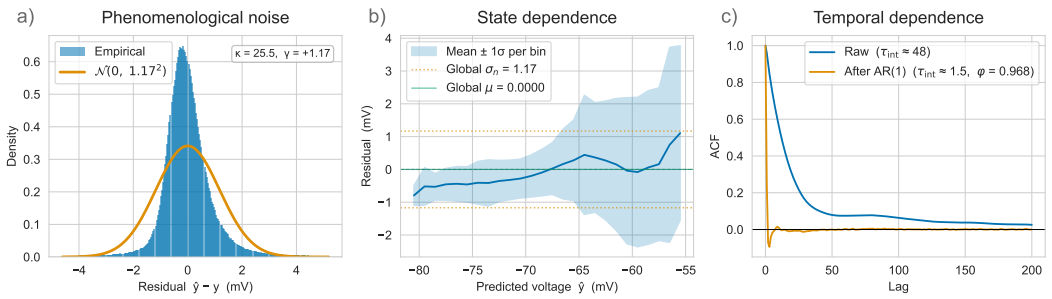


Figure 18: **The ELM neuron voltage prediction residuals on NeuronIO have some state and temporal dependence:** Residuals for a neuron model with $d_m = 3$ fitted on the NeuronIO dataset containing single neuron voltage recordings. Note that the underlying target membrane voltage data itself displays multiple operating regimes, with particularly violent dynamics towards spiking threshold ($\approx -60mV$). **a)** Voltage-prediction residuals are concentrated near zero with mild skew and a sharper-than-Gaussian peak, suggesting the additive noise model to be reasonable, with deviations that can be treated as phenomenological noise. **b)** Residual mean and one-standard-deviation band per predicted-voltage bin: the standard deviation grows mildly toward depolarized states, indicating state-dependent heteroskedasticity. **c)** Residual autocorrelation function: the raw residuals are temporally correlated ($\tau_{\text{int}} \approx 48$), but an AR(1) correction with $\varphi = 0.968$ collapses the ACF to near-zero ($\tau_{\text{int}} \approx 1.5$), showing the temporal structure is dominated by a single slow mode.

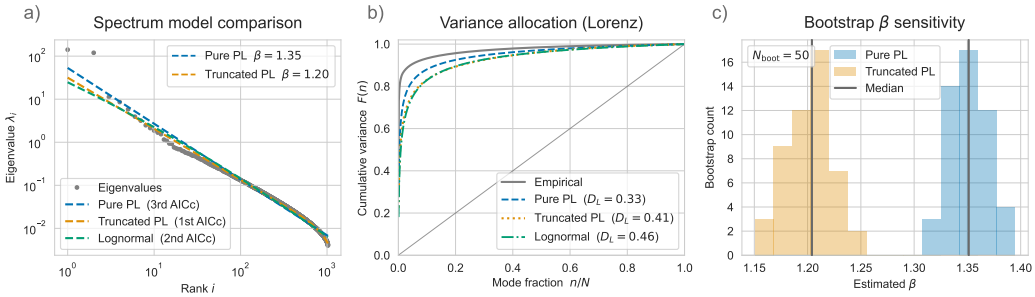


Figure 19: **Power-law structure in the ELM neurons' output.** Function fitting performed in log-log. Individual neuron output measured at memory readout $w^T m_t$ as it contains the task-relevant slow signal components. Eigenvalues computed across 50 distinct recordings of 512 steps, after discarding 128-step burn-in, that were generated using the reference network with $N_{\text{rec}} = 1024$ evaluated on Enwik8. **a)** The neuron's output covariance eigenvalue spectrum is best captured by a truncated power law $\lambda_i = \sigma_f^2\, i^{-\beta} \exp[-(i/i_c)^\nu]$ among the tested spectral models, supporting the assumed heavy-tailed spectral form. **b)** The Lorenz comparison provides an integrated check that the fitted spectrum preserves the variance allocation across modes, rather than only matching eigenvalues point-wise. **c)** Bootstrap resampling of recordings with replacement ($N_{\text{boot}} = 50$) for β estimation quantifies the sensitivity of the fitted spectral exponent, with vertical lines marking the median β values reported in panel a). **Note** that β in our theory refers to the latent signal's power-law exponent; however, we can only directly measure the signal along with residual noise, which may bias the measurement. Notably, none of the measured spectra (even after output nonlinearity) show the flat floor that an isotropic additive residual would produce, indicating a structured rather than isotropic residual. However, since only the effective per-mode signal-to-noise ratios enter I_{rep}, its form does not depend on how the residual is distributed across neurons. For more accurate estimates of β, the measurements may therefore use networks of highly expressive neurons that are assumed to have smaller approximation error (see also Fig. 27).

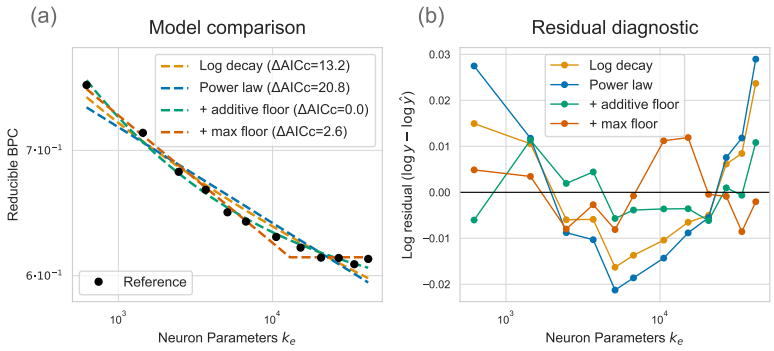


Figure 20: **Empirical neuron complexity scaling on Enwik8 independently validates the max noise floor assumption:** Reducible test BPC vs. per-neuron effective parameter count k_e, with layer width kept fixed at $N = 1024$ but increasing layer budgets. Functions fitted in log-log, with floor function slopes seeded with a pure power-law fit, and floors seeded with the last evaluation point. We compare models using the corrected Akaike Information Criterion (AICc). **a)** The additive floor and max floor $\max(ck_e^{-\alpha}, f)$ models are strongly preferred over the pure power law or log decay (see ΔAICc in the legend), consistent with Assumption **(A3)**, but are not significantly distinguishable from each other. **b)** While residuals reveal a bump for the max floor fit just around its knee (marking the transition into the noise floor), compared to the log decay and pure power law fits (and somewhat additive floor) it doesn't seem to diverge towards the noise floor.

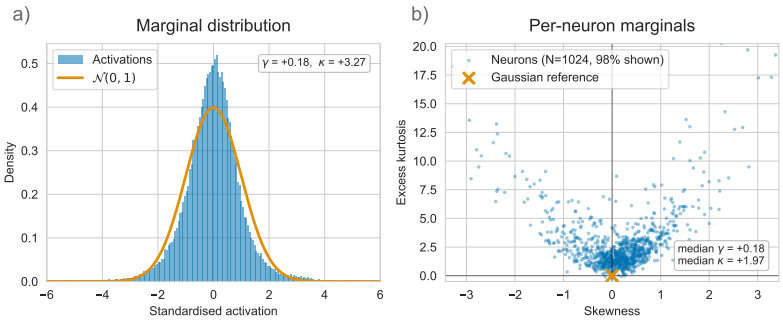


Figure 21: **Gaussianity of the readout $w_r^\top m_t$ of neurons on Enwik8.** Only $y = f + n$ is observable, so Gaussianity is tested on y directly. Data plotted from 50 distinct recordings of 512 steps, after discarding 128-step burn-in. Recordings from the reference model with $N_{\text{rec}} = 1024$ on Enwik8. **a)** Pooled marginal of per-neuron z-scored activity vs. $\mathcal{N}(0, 1)$, shape-only. **b)** Per-neuron (skew, excess kurtosis) with Gaussian reference at $(0, 0)$.

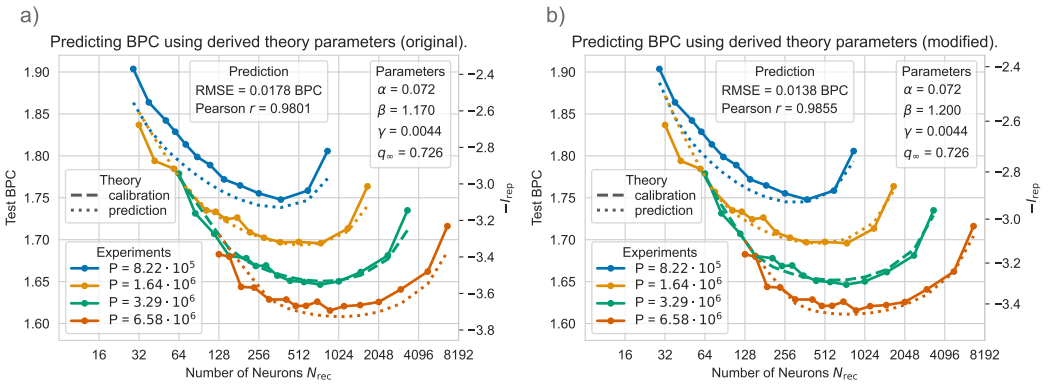


Figure 22: **Independently derived theory parameters predict how the optimal neuron-number–complexity tradeoff shifts across unseen parameter budgets.** This analysis tests whether the theoretical proxy can extrapolate across budgets when its parameters are obtained independently rather than jointly fitted to the displayed tradeoff curves. **Parameter derivation:** The neuron-expressivity exponent $\alpha = 0.072$ is measured from the pre-saturation scaling of reducible BPC with per-neuron complexity k_e at fixed layer width (Fig. 6g and Appendix Fig. 20a, explanation Appendix B.5). The spectral exponent $\beta = 1.17$ is measured from a truncated power-law fit to the recurrent-layer activity covariance spectrum (Fig. 6h and Appendix Fig. 19, details in Fig. 19). The common crossing of the changed α curves is visually identified near the final point of the reference series in Fig. 6b, at $k_e = 225$, giving $\gamma = k_{\text{cross}}^{-1} = 1/225 \approx 0.0044$ (explanation in Appendix B.4. The saturation knee is likewise identified visually in Appendix Fig. 20a: the fourth point from the right is selected because performance no longer improves beyond it, yielding $q_\infty = 0.726$ through $q_\infty = (\gamma k_{\text{sat}})^{-\alpha}$ (explanation in Appendix B.6). **Mapping calibration:** The I_{rep}-to-test-BPC mapping is calibrated using a single experiment series at the reference budget, $P = 3.29 \cdot 10^6$, shown by dashed theory curves. After calibration, the mapping is held fixed and used to predict the tradeoff curves at all other budgets, shown by dotted curves. Solid curves show empirical test BPC, and the reported metrics are pooled across the predicted budgets. **a)** With the affine mapping $\text{BPC} = b + aI_{\text{rep}}$, the locations of the optima and their shifts with budget are predicted qualitatively and quantitatively ($r = 0.9801$) quite well. Nonetheless, BPC at the quarter budget is slightly underestimated for small N, while performance at the doubled budget is somewhat overestimated for large N. **b)** With a saturating exponential mapping $\text{BPC} = B_{\min} + a\exp(-cI_{\text{rep}})$ with floor $B_{\min} = 0.97$ (which can account for diminishing performance gains caused by the fixed-size readout layer, as well as finite task-performance limits not explicitly represented by I_{rep}) and a $\beta = 1.20$ (a slightly higher measured median of the bootstrap estimates in Appendix Fig. 19c) alignment for large-N is improved and under and over predictions reduced for quarter and double budgets. More robust β estimation, e.g., by averaging measurements across multiple trained models, might improve performance further.

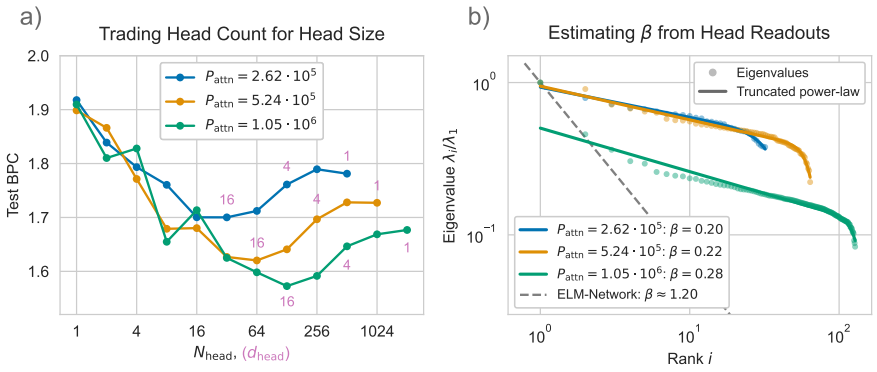


Figure 23: **Similar unit count vs complexity tradeoffs emerge in multi-head attention Transformers, with higher head-count being preferred as budget grows, consistent with a flatter activity spectrum.** Analogous to experiment Fig. 4c we train a range of Transformers on Enwik8 and sweep the number of attention heads N_{head} against the per-head dimension d_{head} (see also [6]) at three fixed attention parameter budgets P_{attn} determined by $(N_{\text{head}} \cdot d_{\text{head}}) \in \{512, 1024, 2048\}$ being held fixed respectively. We use a pre-LN decoder-only Transformer [79] of three blocks with $d_{\text{model}} = 128$ and dense feed-forward width $4d_{\text{model}}$, otherwise trained as in Appendix A.2. Secondary d_{head} axis labels in pink. **a)** A clear interior optimum emerges at every budget and shifts towards more heads with it, as in Fig. 4c. Unlike in ELM-Networks, however, the optimal unit-size remains roughly constant at $d_{\text{head}} \approx 16$ across all three budgets. **b)** We consider each head's additive write into the residual stream as its output activity. As these are vector-valued, we reduce each pair's cross-covariance to its trace, which keeps one channel per head. Eigenvalue spectra of the covariance across head outputs, measured and fitted as in Appendix A.4, yield $\beta \approx 0.20$–0.28, far below the $\beta \approx 1.20$ of the ELM Network's recurrent layer. Having a vector-valued output possibly helps different heads to occupy more distinct activity dimensions. **Interestingly**, with $N_{\text{eff}} = \sum_i i^{-\beta}$ (Appendix B.3), the measured β places the attention layer in the sublinearly growing regime. Additional heads therefore keep contributing informative dimensions, and the substantial inter-unit redundancy that drives the optimum towards more complex units as in Fig. 4c does not set in across the swept budgets and number of heads, consistent with our theoretical proxy's prediction for $\beta \ll 1$. **Furthermore**, a simple unit complexity sweep using d_{head} at fixed $N_{\text{head}} = 32$ and otherwise unchanged hyperparameters yields an estimate of $\alpha \approx 0.122$, slightly higher than the ELM-Networks of $\alpha \approx 0.072$, also indicating better expressivity scaling. **Note** that d_{head} corresponds only roughly to the per-unit complexity k_e, so part of it arguably acts as connectivity overhead k_c instead, in line with the caveat raised in Section 5. The β estimates carry the same signal-plus-noise bias discussed in Fig. 19.

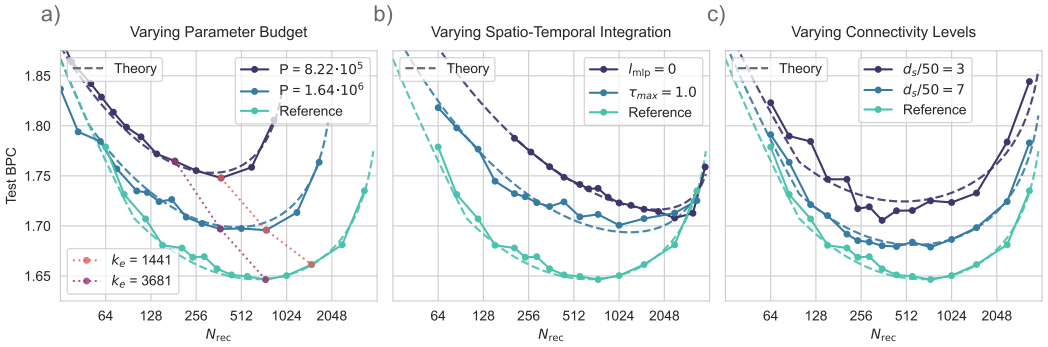


Figure 24: **The theoretical proxy model qualitatively reproduces the empirical scaling tradeoffs.** The displayed experimental tradeoff results and corresponding jointly fitted I_{rep} based theory curves (dashed lines) are the same as in Fig. 6a-f) but plotted on top of each other. Note that the theory curves extend somewhat beyond the experimentally probed range. While the joint theoretical fit tracks the empirical results across all 7 experiments quite well overall, smaller discrepancies remain. For example, the q_{inf}- caused knee and its corresponding steeper performance degradation (left side of each subplot) might happen too late and is likely too steep, indicating that possibly an earlier but softer saturation q_{inf} mechanism would be more accurate.

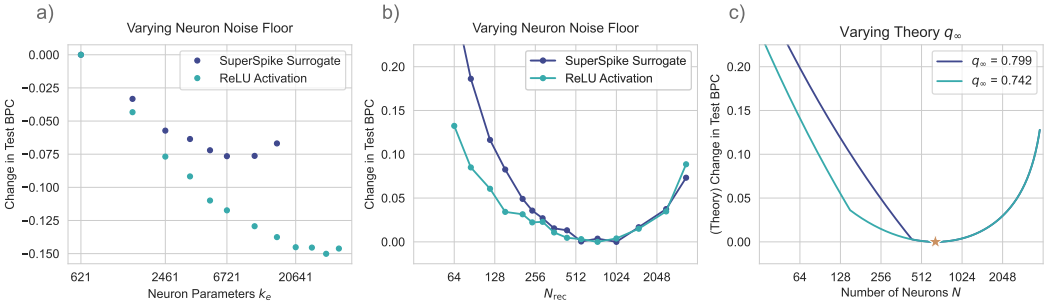


Figure 25: **Changing to binary neuron activation increases neuron noise floor q_∞ and is qualitatively matched by the theoretical proxy.** All results for ELM Networks are evaluated on Enwik8. Change in Test BPC is plotted throughout to exclude a noise-floor-unrelated effect of SuperSpike that leads to an overall degradation in performance (vertical shift, possibly related to change in γ). **a)** Sensitivity analysis with respect to neuron parameters k_e for ELM Networks with $N_{\text{rec}} = 1024$ and either default ReLU or surrogate SuperSpike neuron activation functions. The noise floor is directly visible as the point above which no meaningful improvement in performance happens, which is roughly at 20k and 6.7k parameters per neuron for ReLU and SuperSpike, respectively, indicating a higher neuron noise floor for SuperSpike networks. Particularly large neuron configurations in combination with less robust training methods can lead to training instabilities resulting in a decrease in performance (e.g., see last point SuperSpike). **b)** Empirical evaluations for the fixed budget neuron count vs neuron complexity experiment with both activation functions. Mostly, the performance for small numbers N_{rec} of complex neurons is affected (left side of the panel), conceptually consistent with large neurons reaching their noise floor, possibly with training instabilities causing a widening performance gaps for very large neuron configurations. **c)** Corresponding theory proxy curves with q_∞ values derived using the formula in Appendix B.6 and k_{sat} values 6721 and 20641 respectively from panel a). Overall, the qualitative behavior of the experiment is reproduced, although the theoretical proxy postulates a sharper transition than observed (see also the smoother noise-floor transition in panel a).

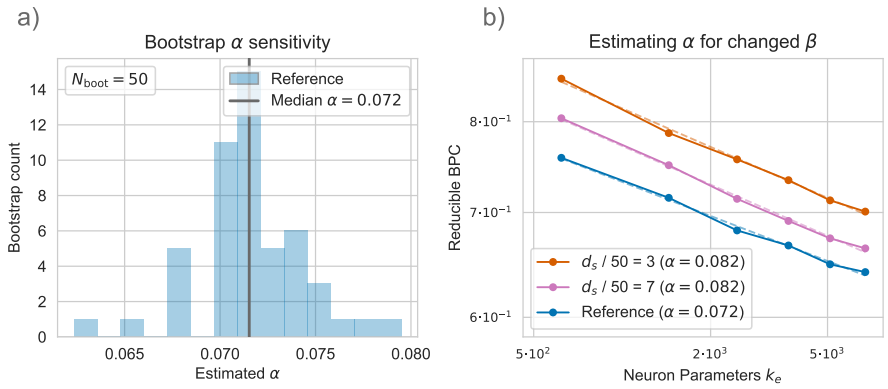


Figure 26: **The neuron-expressivity exponent α measurements shows smaller bootstrap spread, and smaller sensitivity to connectivity changes, compared to changes induced by spatio-temporal integration capabilities.** Measurements for ELM Networks with $N_{\text{rec}} = 1024$ and $d_m = 15$ for panel a) and increasing d_m for panel b). **a)** Bootstrap resampling of the observed (k_e, BPC) pairs with replacement ($N_{\text{boot}} = 50$) also yields the single reference measurement of $\alpha = 0.072$. The bootstrap spread is smaller relative to the separation between the reference and the direct spatio-temporal-integration modifying experiment measurements in Fig. 6g which yielded 0.036 and 0.044 respectively. **b)** Reducing connectivity, which should conceptually only impact the independently measured β, yields matching α estimates within the approximate bootstrap spread. The deviations are also much smaller than changes induced by directly weakening neuron integration in Fig. 6g, which yielded α estimates of 0.036 and 0.044 respectively. This further evidences a reasonably clean association between theory and architectural parameters.

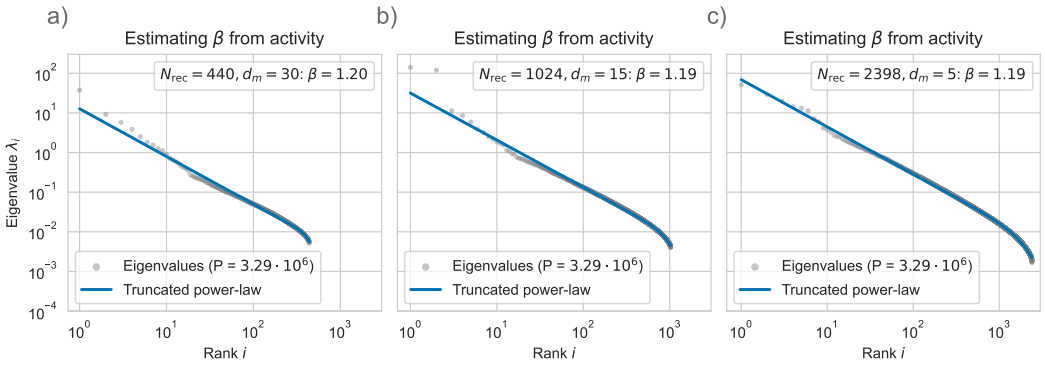


Figure 27: **The spectral exponent β is invariant, within bootstrap spread, across network width and neuron complexity at a fixed parameter budget.** Same activity recording and power-law fitting protocol as in Fig. 19 (cutoff fitted per configuration), applied to three configurations of the reference budget sweep $P = 3.29 \cdot 10^6$ of Fig. 6a. **a–c)** The three network configurations yield roughly the same $\beta = 1.20, 1.19, 1.19$, well within the bootstrap spread of Fig. 19c, despite an order-of-magnitude change in per-neuron parameters k_e and a fivefold change in width, however, show slight deviation from the shared-cutoff measurement yielding 1.17 in Fig. 6h, likewise within the spread. The framework treats β as a layer property shared along a budget sweep irrespective of neuron count or neuron complexity; the invariance observed here is consistent with this, supports the shared-β assumption behind the joint fit and the budget extrapolation of Fig. 22, and indicates that β can be measured reliably within a given neuron type. **Note**, however, that preliminary measurements of β across widely changed α (such as in Fig. 6g) were less consistent, likely due to the measured activity spectrum containing signal and residual together (see note in Fig. 19), and architectural changes also altering the residual structure. We overall recommend measuring β within networks of neurons with sufficient spatio-temporal integration capability (higher α), such as the reference configuration.

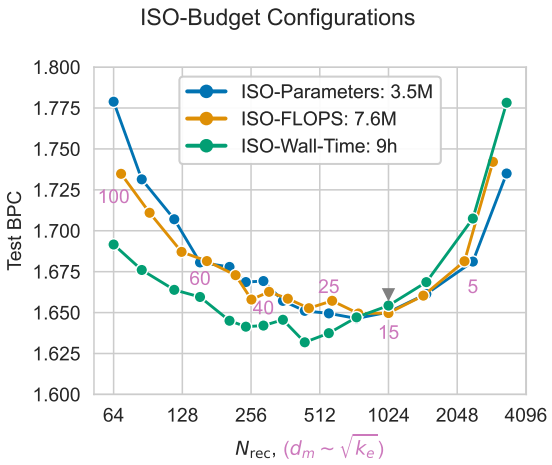


Figure 28: **The characteristic tradeoff between number of neuron and complexity of neuron persists across different budget types:** besides (trainable) parameters we additionally consider (inference) FLOPS and (training) wall-time of the whole network as budgets. For ISO-FLOPS (7.6M for single step) large neuron configurations gain width whereas small neuron configurations lose width, yet the tradeoff essentially looks qualitatively and quantitatively unchanged For ISO-Wall-Time (9h) the interior-optimum shifts towards larger neuron network configurations, which execute faster (resulting in more training steps compared to simpler neuron configurations) due to increased hardware alignment of larger but fewer matrix operations on NVIDIA GPUs. The resulting varied training length might be accounted for using varied γ in the theoretical proxy.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The three main claims (monotonic per-axis scaling with a non-trivial budget-constrained optimum, an information-theoretic framework reproducing the tradeoff shapes, and a scaling recipe tracing the Pareto frontier) are each demonstrated in Sections 4.1–4.3. The scope is ELM Networks on SHD-Adding and Enwik8; we do not claim cross-architecture or SOTA-comparison results.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We explicitly discuss the limitations of our work in section 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: A full derivation of I_{rep} including all assumption can be found in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We describe the ELM Network architecture in section 2, the experimental and theoretical setups in section 4, with all remaining methodological details in Appendix A.2.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a full implementation of the ELM Network including training and evaluation scripts for full reproducibility. All benchmark datasets that were used are already publicly available.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We describe the ELM Network architecture in section 2, the experimental and theoretical setups in section 4, with all remaining methodological details in Appendix A.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: SHD-Adding and NeuronIO sweeps report median or mean$\pm$std over three seeds (captions specify). Enwik8 uses one seed per configuration because of the compute cost of training character-level language models at these network sizes; the figures cover many configurations densely, so the trends are visible without per-point error bars. Joint theory fit quality is given as Pearson r in Fig. 6, and ΔAICc for comparing candidate functional forms is reported where it applies (e.g., Fig. 17).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources and execution times discussed in Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work used non-sensitive public datasets and performed computational experiments without human trials, and in compliance with NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

Justification: Our work is foundational research on neural architecture scaling and does not target a specific downstream application, deployment setting, or human population.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: We do not provide new datasets, and trained models are not applicable to sensitive data and therefore do not pose any risk for misuse.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: Please refer to the data and code availability paragraph in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new datasets were introduced, however, some model weights might be released with instructions after publication as part of the code release.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: No crowdsourcing or research involving human subjects was performed.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No crowdsourcing or research involving human subjects was performed.

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as part of the research methodology, but were employed as assisting tools for writing and coding.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.