Supplementary Materials

Robust Power-Law Computing for Edge Intelligence Using Spin-Orbit Torque Magnetic Tunnel Junctions

Shilong Xiong[1,2], Caihua Wan[1,2]*, Xiaohan Li[1], Yingqian Xu[1,2], Dehao Kong[1,2], Shengkai Xia[1,2], Tianyi Zhang[1,2], Aleksandr V. Davydenko[3], Guanxiang Du[4], Alexander S. Samardak[3]*, Guoqiang Yu[1,2,3], Xiufeng Han[1,2]*

[1]Beijing National Laboratory for Condensed Matter Physics, Institute of Physics, Chinese Academy of Sciences, Beijing 100190, China

[2]Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

[3]Laboratory of Spin-Orbitronics, Institute of High Technologies and Advanced Materials, Far Eastern Federal University, Vladivostok 690922, Russia

[4]Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China

Email: wancaihua@iphy.ac.cn; samardak.as@dvfu.ru; xfhan@iphy.ac.cn

Supplementary Note 1: Device Fabrication, Measurement Setup, and Extended Stochastic Characterization

1.1 Device fabrication and physical configuration

The SOT-MTJ entropy source uses a bottom W channel to generate spin-orbit torque. The stack, W(3)/CoFeB(1.4)/MgO(1.5)/CoFeB(3)/W(0.4)/Co(2.7)/IrMn(10)/Ru(4), was deposited at room temperature by magnetron sputtering; thicknesses are in nanometers. The films were then annealed under high vacuum at a base pressure below 4×10^{-4} Pa to improve crystallinity and tunneling magnetoresistance. Nanopillars were patterned by electron-beam lithography and ion-beam etching.

Each pillar has an approximately 300 nm × 100 nm elliptical footprint, with its long axis along y defining the free-layer easy axis through shape anisotropy. Write current flows along x through the W channel, generating a spin current along z with polarization along $-y$. The collinearity of the spin polarization and the easy axis enables field-free switching in this Type-Y configuration.

1.2 Electrical measurement setup and basic magneto-transport characterization

The device has separate write and read paths. For resistance readout, the top electrode is split into two pads so that the current-bias and voltage-sensing leads form a four-terminal Kelvin connection, reducing contributions from contact and lead resistances. An Agilent 81104A pulse generator applies 50 ns write pulses to the SOT channel. A Keithley 2400 source meter supplies a 0.1 μA sensing current, and a Keithley 2182 nanovoltmeter records the readout voltage.

At room temperature, the parallel and antiparallel resistances, R_P and R_{AP}, are approximately 36 kΩ and 100 kΩ, giving a TMR ratio of about 175%. The magnetic hysteresis loop has a coercive field of approximately 75 Oe and a shift of approximately −25 Oe, attributed mainly to the stray field of the synthetic antiferromagnetic reference layer. Under zero applied magnetic field, 50 ns pulses produce deterministic P-to-AP and AP-to-P switching at approximately −1.8 V and +2.4 V, respectively.

The TMR loop, deterministic switching, and 25 P–V curves in Figure 1 are direct electrical measurements. Independent sigmoid fits give V_c=1.881±0.151 V and k_s=7.543±1.883 V^{-1} (mean ± sample standard deviation). The observed V_c range is 1.603–2.236 V, or −14.78% to +18.86% relative to the measured mean; the observed k_s range is 3.595–11.818 V^{-1}, or −52.34% to +56.68%. The joint ±90% domain in Supplementary Note 5 is a prescribed sensitivity range centered on the measured means; it contains all 25 fitted parameter pairs and extends beyond their observed spread.

Supplementary Note 2: Event-Driven Feedback Algorithm and Operating Conditions

2.1 Pseudocode and execution flow

The algorithm first regulates the switching probability and then decodes power-law outputs from the resulting bitstream. For a normalized interior target $b \in (0,1)$, each feedback block contains n physical Bernoulli trials at the same voltage. The all-1 indicator A_k determines the next voltage update. At an accessible zero-drift fixed point on an increasing P–V branch, the model gives $p=b^{1/n}$; grouping the output into m-bit all-1 events then targets $b^{m/n}$. Finite-step feedback produces fluctuations about this ideal operating point, so decoding uses a declared post-lock interval.

The Figure 3 trajectories start at $V_0=0$ V with a fixed learning step $l=50$ mV. Algorithm 1 specifies acquisition and decoding separately. Voltage limits, endpoint handling, and possible state reuse are discussed in Section 2.4.

Algorithm 1. Event-driven acquisition and power-law decoding

Inputs: normalized interior target b; positive integers n, m, and K; voltage step $l>0$; initial voltage V_0; burn-in length B_{burn} in raw bits; and positive integer decoder stride s. Choose the retained interval so that at least m raw bits remain. The acquisition cost is $N_{\mathrm{bit,fb}}=nK$ physical trials.

Output: empirical estimate $\hat{Y}_{m/n}$ from the retained raw-bit interval.

```
V ← V0
Bitstream ← []
Step_Up ← l × b
Step_Down ← l × (1 − b)

// Phase I: feedback acquisition
For k = 1 to K:
    Block ← []
```

```
    For i = 1 to n:
        x ← Reset_Write_Read(V)      // One physical trial; returns 0 or 1
        Append x to Block
    Append all bits of Block to Bitstream
    A_k ← 1 if all bits in Block are 1, otherwise 0
    If A_k = 1:
        V ← V − Step_Down
    Else:
        V ← V + Step_Up


// Phase II: decoding (one-based indexing)
Target_Stream ← Bitstream[B_burn + 1 … End]
N_bit ← Length(Target_Stream)
Require N_bit ≥ m
G ← 0
All_Ones_Count ← 0
For j = 1, 1 + s, 1 + 2s, … while j + m − 1 ≤ N_bit:
    If all bits in Target_Stream[j … j + m − 1] are 1:
        All_Ones_Count ← All_Ones_Count + 1
    G ← G + 1
Ŷ_m/n ← All_Ones_Count / G
Return Ŷ_m/n
```

The decoder visits G=floor((N_{bit}−m)/s)+1 valid groups. With s=1, adjacent groups overlap; with s=m, they share no raw bits. Multiple values of m can be decoded from the same stream when the outputs share the same base and denominator. $\hat{Y}_{m/n}$ is a finite-sequence estimate, rather than an identity with $b^{m/n}$. The outer blocks used to summarize experimental uncertainty are a separate partition of the raw stream, defined in Supplementary Note 3.

2.2 Derivation of the expected drift and fixed-point condition

Let $p = \sigma(V)$ be the switching probability on a monotonically increasing operating branch. In feedback block k, A_k=1 when all n trials return 1, and A_k=0 otherwise. Conditional independence of the trials at fixed V_k gives

$$P(A_k = 1|V_k) = [\sigma(V_k)]^n.$$

Correspondingly,

$$P(A_k = 0 \mid V_k) = 1 - [\sigma(V_k)]^n.$$

Under the update rule in Eq. (1), $\Delta V_k = V_{k+1} - V_k$ is

$$\Delta V_k = \begin{cases} -l \cdot (1-b), & A_k = 1, \\ l \cdot b, & A_k = 0. \end{cases}$$

The conditional expectation of the voltage increment is therefore

$$E[\Delta V_k \mid V_k] = [-l \cdot (1-b)]P(A_k = 1 \mid V_k) + (l \cdot b)P(A_k = 0 \mid V_k).$$

Substituting the probabilities above gives

$$E[\Delta V_k \mid V_k] = [-l \cdot (1-b)][\sigma(V_k)]^n + (l \cdot b)(1 - [\sigma(V_k)]^n),$$

which simplifies to

$$E[\,\Delta V_k \mid V_k\,] = l(b - [\sigma(V_k)]^n). \qquad (S1)$$

The zero-drift fixed point $V_{\text{stable}}=V^*$ is defined by setting this conditional mean update to zero:

$$E[\Delta V_k \mid V_k] = 0 \Rightarrow [\sigma(V_{\text{stable}})]^n = b. \qquad (S2)$$

Thus $\sigma(V^*)=b^{1/n}$ when the target probability is reachable within the safe voltage range. This is a fixed-point relation for the conditional mean dynamics.

2.3 Local stability, learning-rate selection, and steady-state fluctuation

Local stability determines whether the mean dynamics return toward the fixed point after a small perturbation. Define the feedback-event probability as

$$q(V) = [\sigma(V)]^n.$$

Linearization about V^*, where $q(V^*)=b$, gives

$$\delta V_{k+1} \simeq [1 - lq'(V^*)]\delta V_k + \text{stochastic fluctuation.}$$

The dimensionless gain $g_{\text{eff}}=lq'(V^*)$ must satisfy $0<g_{\text{eff}}<2$ for local stability of the linearized mean dynamics. The stricter condition $g_{\text{eff}}<1$ avoids alternating mean convergence. For the sigmoid model with slope parameter k_s,

$$g_{\text{eff}} = nlk_s b(1 - b^{1/n}).$$

Because $nb(1-b^{1/n})<1/e$, the gain satisfies $g_{\text{eff}}<lk_s/e$. A technology-level upper slope bound $k_{s,\max}$ therefore gives the conservative design rule $l<e/k_{s,\max}$. This uses an operating envelope without requiring a stored, device-specific voltage-to-probability table. In the present parameterized study, the mean fitted slope of the 25 measured devices is $k_s=7.543\ \text{V}^{-1}$. The prescribed ±90% range centered on this mean spans 0.754–14.332 V^{-1}. Using its upper edge gives $l<189.7$ mV. The experimental step of 50 mV lies within this model envelope; the calculation does not establish a population-wide device bound.

The learning rate leaves the ideal fixed point unchanged but affects both convergence and precision. A larger step can shorten the cold-start transient while increasing steady-state fluctuations. In the local stochastic model, the voltage variance is approximately

$$\text{Var}(\delta V) \simeq \frac{lb(1-b)}{q'(V^*)[2 - lq'(V^*)]}.$$

For small effective gain, this variance is proportional to l. A two-stage or decaying learning rate could therefore balance acquisition time and steady-state precision.

2.4 Operational limits, normalization, and controller state

Probability encoding restricts b to [0,1]. Extending the operation to other input ranges requires a specified normalization and a corresponding output transformation that preserves the intended power law. For an interior target, the required fixed point must also lie on a monotonic branch within the accessible voltage range. A practical controller can enforce that range through

$$V_{k+1} = \mathrm{clip}_{[V_{\min}, V_{\max}]}\{V_k + l(b - A_k)\},$$

where A_k is the all-1 indicator and $\mathrm{clip}_{[l,\,u]}(x)$ constrains x to the interval $[l, u]$. The limits define a safe operating envelope; they do not constitute a stored probability calibration. The reported protocol does not specify implemented voltage bounds or a saturation policy. At b=0 or b=1, deterministic output requires a safely reachable no-switching or switching regime; otherwise, digital endpoint handling is needed.

The locked voltage is the internal state of the active feedback loop. Retaining it could provide a warm start for a nearby target while allowing continued drift compensation. Such a warm start can perform as a possible engineering strategy to reduce the number of iterations required for convergence, thereby lowering both computational latency and the energy consumed during convergence.

2.5 Temporal dependence, error propagation, and estimator uncertainty

Holding the voltage fixed within a feedback block does not itself establish independence of its raw bits. Supplementary Note 3 tests short-range dependence in the measured post-lock streams and distinguishes it from the dependence introduced by overlapping decoder windows. Both effects matter when relating the ideal model to experimental uncertainty.

Under the conditional sampling model, the update $V_{k+1}=V_k+l(b-A_k)$ has variance

$$\mathrm{Var}(\Delta V_k | V_k) = l^2 q(V_k)[1 - q(V_k)].$$

At the fixed point, the variance becomes $l^2 b(1-b)$, supplying the stochastic forcing in the local fluctuation model. For a fixed switching probability p, an output estimate from G independent, nonoverlapping m-bit groups has target $y=p^m$ and

$$E[\hat{y}] = y, \mathrm{Var}(\hat{y}) = \frac{y(1-y)}{G}.$$

The root-mean-square sampling error then scales as $G^{-1/2}$. Table S1 reports uncertainty from the experimental block analysis, while Supplementary Note 7 relates the ideal scaling to physical sampling cost.

2.6 Relation to classical PID control

Like classical feedback control, the method uses a set point and negative feedback to reject disturbances. Its update accumulates the binary event error $b-A_k$ in the voltage state. The probability error $b-[\sigma(V_k)]^n$ appears in the conditional expectation of that update, without being directly measured.

This structure is analogous to stochastic integral control: it has no explicit proportional or derivative term and therefore is not a full PID law. In the 5 Oe perturbation experiment, the controller adjusts the operating voltage after the device response changes and recovers the target event probability, illustrating disturbance rejection for a probabilistic output.

2.7 Scope beyond direct power-law functions

The demonstrated primitive produces power-law terms directly from event statistics. With n=1, parallel decoders could estimate x, x^2, …, x^M from one stream, providing a basis for a finite approximation $f(x)\approx\Sigma c_m x^m$. Exponential or logarithmic functions could be approximated in this way over a restricted range.

Such extensions require coefficient storage, weighting and accumulation, as well as signed processing and appropriate scaling when coefficients or outputs leave [0,1]. Logarithms also require an input range bounded away from zero or a suitable range-reduction strategy. These backend operations have not been demonstrated here; the experimental scope remains direct power-law computation.

Supplementary Note 3: Extended Experimental Evaluation of Power-Law Computing

3.1 Output estimates, uncertainty, and temporal dependence

The power-law outputs in Figure 3 were evaluated by digital analysis of experimentally acquired SOT-MTJ bitstreams. Each representative trajectory comprised 1500 raw bits acquired from an initial voltage of 0 V with a fixed feedback step parameter of 50 mV. Sliding readout windows of $L = 50$ bits were used for $0.5^{3/2}$ and $0.5^{1/3}$, and $L = 100$ bits for 0.5^{3}. For statistical analysis, the first 20% of the sliding-window positions were excluded, with the cutoff rounded down to an integer. The retained windows therefore spanned the final 1210, 1210, and 1220 raw bits of the respective trajectories.

For a retained sequence of N bits, the output estimate in the window starting at position j was calculated as

$$\hat{P}_{m,j} = \frac{1}{L-m+1} \sum_{i=0}^{L-m} \prod_{r=0}^{m-1} b_{j+i+r}, j = 1, \dots, N-L+1,$$

where $b_i \in \{0,1\}$ denotes a measured bit and m is the output-group length. This estimator reduces to the fraction of ones for $m = 1$ and to the fraction of consecutive all-one triples for $m = 3$. Readout windows were advanced by one bit, and the reported point estimate was obtained by averaging all retained window estimates. The resulting values were 0.351, 0.793, and 0.127, respectively, in agreement with the corresponding theoretical targets (Table S1). For stationary independent raw bits with one-probability p, the estimator has expectation p^m. Although overlapping groups preserve this expectation, shared bits introduce dependence between neighboring estimates.

To account for this overlap in the uncertainty analysis, each retained sequence was partitioned into nonoverlapping outer blocks of the same length as the readout window. An output estimate Q_j was calculated for each block using the same decoder, with all-one groups confined to that block. The block mean and sample standard deviation were

$$\bar{Q} = \frac{1}{B}\sum_{j=1}^{B} Q_j \, , s_Q = [\frac{1}{B-1}\sum_{j=1}^{B} (Q_j - \bar{Q})^2]^{1/2},$$

where B is the number of complete blocks. Approximate two-sided 95% confidence intervals were constructed using Student's t distribution,

$$\mathrm{CI}_{95\%} = \bar{Q} \pm t_{0.975,B-1}\frac{s_Q}{\sqrt{B}}.$$

These intervals quantify uncertainty in the block mean within each measured trajectory, assuming sufficiently weak dependence between blocks. Their centers may differ slightly from the reported point estimates because the latter average all overlapping readout windows. The block-level standard deviations and confidence intervals are summarized in Table S1.

Temporal correlations were assessed directly from the retained raw-bit sequences. At each lag $k = 1, \ldots, 10$, the autocorrelation was calculated as the Pearson correlation between the two sequences $(b_1, \ldots, b_{N-k})$ and $(b_{1+k}, \ldots, b_N)$. The lag-1 coefficients were +0.037, −0.033, and +0.019 for $0.5^{3/2}$, $0.5^{1/3}$, and 0.5^3, respectively. Across the examined lags, the absolute coefficients remained below 0.058, comparable to the approximate pointwise 95% white-noise reference bounds of $\pm 1.96/\sqrt{N} \approx \pm 0.056$.

The sensitivity of the $m = 3$ estimates to bit ordering was further examined using a within-block permutation test. Bits were randomly reordered within each outer block while preserving its number of ones, and the mean block-level $P(111)$ was recalculated for each of 10,000 permutations using a fixed random seed of 20260825. With T_{obs} denoting the observed statistic, T_r the statistic from permutation r, and $\bar{T}_{\mathrm{rand}}$ the mean over all permutations, the two-sided p value was calculated as

$$p_{\mathrm{perm}} = \frac{1 + \sum_{r=1}^{R} 1\,[|T_r - \bar{T}_{\mathrm{rand}}| \geq |T_{\mathrm{obs}} - \bar{T}_{\mathrm{rand}}|]}{R+1}, R = 10000.$$

The resulting p values were 0.17 for $0.5^{3/2}$ and 0.57 for 0.5^3, indicating no significant excess or deficit of all-one triples relative to randomized ordering with the same block compositions. The $m = 1$ estimates are invariant to within-block reordering. Together, the autocorrelation and permutation analyses characterize short-

range temporal dependence and its effect on the all-one statistic, while retaining the distinction between overlapping estimates and independent observations.

Supplementary Table S1. Statistical summary of the measured power-law outputs. Point estimates are averages of the retained sliding-window estimates. Standard deviations and approximate 95% t confidence intervals are calculated from nonoverlapping outer-block estimates and describe within-trajectory sampling uncertainty. The intervals are centered on the block means.

Alt text: Statistical summary for three power-law tasks. Each measured point estimate is close to its theoretical target. The table also reports window lengths, independent block counts, block-level standard deviations, and confidence intervals used to characterize sampling uncertainty.

Task	**$0.5^{3/2}$**	**$0.5^{1/3}$**	**0.5^{3}**
Decoder (n, m)	(2, 3)	(3, 1)	(1, 3)
Window L (raw bits)	50	50	100
Point estimate	0.351	0.793	0.127
Theoretical target	0.354	0.794	0.125
Nonoverlapping outer blocks	24 × 50 bits	24 × 50 bits	12 × 100 bits
Block-level SD	0.070	0.046	0.057
95% CI for block mean	[0.323, 0.382]	[0.774, 0.813]	[0.093, 0.165]

3.2 Decoding multiple outputs from one bitstream

Outputs with a common base and denominator can share the same source and feedback loop. Once the mean dynamics approach the operating point associated with $b^{1/n}$,

parallel grouping and counting modules can estimate $b^{m/n}$ for different values of m. Each output retains its own counting and storage requirements, while the physical source and controller are shared.

In the demonstration with $n = 3$, a single SOT-MTJ stream was processed to obtain $0.5^{1/3}$, $0.5^{2/3}$, 0.5^{1}, $0.5^{4/3}$, and $0.5^{5/3}$. This demonstrates reuse of measured samples across power-law outputs. Its hardware benefit depends on how many outputs are needed concurrently, their precision, and the counter and peripheral implementations. Supplementary Note 4 sets out the corresponding resource accounting.

Supplementary Note 4: Derivation of Hardware-Resource and Energy Expressions

4.1 General derivation framework

Resource costs are estimated by decomposing each implementation into primitive circuit blocks. To compare the dominant contributions consistently, we use the vector

$$\mathbf{C} = [S, A, P, T],$$

where S denotes storage, A denotes deterministic arithmetic and digital datapath complexity, P denotes peripheral macro overhead, and T denotes runtime repetitions per function evaluation.

The storage term S counts bits in function tables, coefficient memories, ROMs, SRAMs, and register files:

$$S = \sum_i B_i,$$

where B_i is the capacity of memory component i.

The arithmetic term A describes digital datapath complexity through asymptotic gate-equivalent scaling. A W-bit adder, comparator, counter incrementer, or simple event detector has approximate complexity

$$O(W),$$

whereas a general fixed-point multiplier with word length W scales approximately as

$$O(W^2).$$

This distinction separates general multiplier-based approximations from an MTJ datapath built mainly from event detectors, shift registers, counters, and update logic.

The peripheral term P includes memory-access circuits, coefficient selection, MTJ drivers, sensing and reference circuits, programmable sources, and bias generation:

$$P = \sum_k P_k,$$

Each P_k remains an explicit macro contribution because its area and energy depend on the technology and circuit design.

The runtime term T counts operations per function evaluation. Its primitive unit depends on the architecture, such as a memory access, a multiply-add cycle, or a physical stochastic trial. Thus, a single-access LUT has

$$T_{\mathrm{LUT}} = 1,$$

whereas for the MTJ online stochastic implementation, if N_{bit} denotes the total number of individual physical Bernoulli samples included in one evaluation,

$$T_{\mathrm{MTJ}} = N_{\mathrm{bit}}.$$

The energy of each implementation follows the same operation-count principle,

$$E_{\mathrm{scheme}} = \sum_i N_i \, E_i,$$

where N_i and E_i are the count and unit energy of operation i. The same accounting applies to memory access, arithmetic, MTJ sampling, sensing, event detection, counting, and feedback.

4.2 Derivation of CMOS baseline resource expressions

4.2.1 LUT-based implementation

A LUT stores the target function at every supported input code. An input width of W_{in} bits gives

$$N_{\mathrm{addr}} = 2^{W_{\mathrm{in}}}.$$

Each stored output has W_{out} bits. For one fixed function setting, the total storage is

$$S_{\mathrm{LUT}}^{(1)} = 2^{W_{\mathrm{in}}} W_{\mathrm{out}}.$$

If K function-parameter settings are stored independently, either in separate tables or in one table with an additional parameter address, the storage becomes

$$S_{\mathrm{LUT}} = K2^{W_{\mathrm{in}}}W_{\mathrm{out}}.$$

Address generation, output buffering, and optional formatting give the approximate arithmetic term

$$A_{\mathrm{LUT}} \sim O(W_{\mathrm{out}}).$$

Address decoders, word-line and bit-line circuits, sensing, and output registers are grouped into

$$P_{\mathrm{LUT}} = P_{\mathrm{mem}}.$$

For a single table access, the runtime term is

$$T_{\mathrm{LUT}} = 1.$$

Combining these terms gives

$$\mathbf{C}_{\mathrm{LUT}} \sim [K2^{W_{\mathrm{in}}}W_{\mathrm{out}},\ O(W_{\mathrm{out}}),\ P_{\mathrm{mem}},\ 1].$$

Each additional input bit doubles this storage requirement. The scaling matters most at higher input precision or with multiple independently selectable mappings.

4.2.2 Piecewise-linear implementation

A piecewise-linear implementation divides the input range into N_{seg} segments. For the j-th segment, the function is approximated by

$$y \approx a_j x + b_j,$$

where a_j and b_j are the stored slope and intercept. If the slope and intercept word lengths are W_{slope} and $W_{\mathrm{intercept}}$, the coefficient storage is

$$S_{\mathrm{PWL}} = N_{\mathrm{seg}}\left(W_{\mathrm{slope}} + W_{\mathrm{intercept}}\right).$$

The datapath first selects the segment, fetches the corresponding coefficients, and evaluates one multiply-add operation. For a general fixed-point multiplier with word length W,

$$A_{\text{mult}} \sim O(W^2).$$

With a general multiplier, addition and segment selection contribute lower-order terms, giving

$$A_{\text{PWL}} \sim O(W^2).$$

Coefficient memory, segment-selection logic, and associated registers are collected as

$$P_{\text{PWL}} = P_{\text{coef}}.$$

For a fully combinational or pipelined single-evaluation datapath, the runtime term is expressed as

$$T_{\text{PWL}} = 1.$$

The resulting resource vector is

$$\mathbf{C}_{\text{PWL}} \sim \left[N_{\text{seg}}\left(W_{\text{slope}} + W_{\text{intercept}}\right),\ O(W^2),\ P_{\text{coef}},\ 1\right].$$

Constraining slopes a_j to powers of two or short sums of powers of two allows shift-add implementations. This reduces arithmetic cost at the expense of the available coefficient set.

4.2.3 Polynomial implementation

A polynomial implementation approximates the nonlinear function as

$$f(x) \approx \sum_{i=0}^{d} c_i\, x^i,$$

where d is the polynomial degree and c_i are stored coefficients. The coefficient storage requirement is

$$S_{\text{poly}} = (d+1)W_{\text{coef}},$$

where W_{coef} is the coefficient word length.

A Horner structure evaluates the polynomial as

$$f(x) \approx c_0 + x\big(c_1 + x(c_2 + \cdots + xc_d)\big).$$

Horner evaluation uses approximately d multiply-add stages. A fully unrolled datapath therefore has

$$A_{\text{poly}} \sim O(dW^2),$$

because each stage contains a general W-bit multiplier. Control logic, registers, coefficient access, and multiplexing are collected into

$$P_{\text{poly}} = P_{\text{ctrl}}.$$

For an unrolled or deeply pipelined datapath, the normalized evaluation count is

$$T_{\text{poly}} = 1.$$

Thus,

$$\mathbf{C}_{\text{poly}} \sim [(d+1)W_{\text{coef}},\ O(dW^2),\ P_{\text{ctrl}},\ 1].$$

If a serial architecture reuses one multiplier and one adder, the arithmetic hardware can be reduced to

$$A_{\text{poly,serial}} \sim O(W^2),$$

while the runtime increases to approximately

$$T_{\text{poly,serial}} \sim d.$$

Serial reuse reduces hardware at the cost of additional evaluation cycles.

4.3 Derivation of the MTJ resource and energy expressions

The MTJ implementation represents the nonlinear operation through physical Bernoulli sampling and event-driven feedback. Its deterministic datapath detects events, counts their occurrences, and updates the applied voltage.

An n-bit shift register holds the feedback-block history. The corresponding event detector scales as

$$A_{\text{event}} \sim O(n).$$

The number of detected events is accumulated by a counter with word length W_c, giving

$$A_{\text{counter}} \sim O(W_c).$$

The feedback-update circuit adjusts the control variable using a finite word length W_u. Its add/subtract and saturation logic scale as

$$A_{\text{update}} \sim O(W_u).$$

The deterministic arithmetic and logic term is therefore

$$A_{\text{MTJ}} \sim O(n + W_c + W_u).$$

Direct power-law computation needs neither a complete input-output table nor a coefficient table, so its function-storage term is

$$S_{\text{MTJ}}^{\text{function}} \approx 0.$$

Controller state, counters, shift registers, and configuration storage are still required. They are included in the digital and peripheral terms; the zero entry refers only to function-table storage.

The peripheral macros are dominated by write-pulse generation, resistance-state readout, sensing, and programmable biasing. These contributions are written as

$$P_{\text{MTJ}} = P_{\text{driver}} + P_{\text{SA}} + P_{\text{bias}}.$$

If N_{bit} physical samples are required for one function evaluation, the runtime term is

$$T_{\mathrm{MTJ}} = N_{\mathrm{bit}}.$$

Combining the storage, arithmetic, peripheral, and runtime terms yields

$$\mathbf{C}_{\mathrm{MTJ}} \sim [0, O(n + W_c + W_u), P_{\mathrm{driver}} + P_{\mathrm{SA}} + P_{\mathrm{bias}}, N_{\mathrm{bit}}].$$

For energy accounting, each physical sample incurs write, read, sensing, and logic costs. The expression below also allocates a feedback-update contribution per sample; if the controller updates once per block, that contribution must be averaged over the block or counted separately:

$$E_{\mathrm{MTJ}} \approx N_{\mathrm{bit}}(E_{\mathrm{write}} + E_{\mathrm{read}} + E_{\mathrm{sense}} + E_{\mathrm{logic}} + E_{\mathrm{update}}) + E_{\mathrm{control}}.$$

The write energy can be expressed from the pulse voltage, current, and pulse duration as

$$E_{\mathrm{write}} \approx V_{\mathrm{write}} I_{\mathrm{write}} \tau_{\mathrm{write}},$$

or, for a resistive load approximation,

$$E_{\mathrm{write}} \approx \frac{V_{\mathrm{write}}^2}{R_{\mathrm{eff}}} \tau_{\mathrm{write}}.$$

The read energy can be written similarly as

$$E_{\mathrm{read}} \approx V_{\mathrm{read}} I_{\mathrm{read}} \tau_{\mathrm{read}}.$$

The sensing, logic, and update terms, E_{sense}, E_{logic}, and E_{update}, require circuit models or measurements. Any reset pulse must also be included in the write budget. Keeping these terms explicit allows the estimate to reflect the actual acquisition and control sequence.

4.4 Derivation of the summary relations in the main text

The first summary relation compares the storage terms of the MTJ and LUT implementations. From the LUT derivation,

$$S_{\mathrm{LUT}} = K 2^{W_{\mathrm{in}}} W_{\mathrm{out}},$$

whereas the complete-function-table storage term of the MTJ scheme is

$$S_{\mathrm{MTJ}}^{\mathrm{function}} \approx 0.$$

For the complete-function storage term, this gives

$$S_{\mathrm{MTJ}}^{\mathrm{function}} \ll S_{\mathrm{LUT}}$$

when the LUT stores mappings at nontrivial input precision or multiple parameter settings. This compares function-table storage only; controller state and peripherals remain part of the total implementation.

The second relation compares the deterministic arithmetic terms. For PWL and polynomial approximations,

$$A_{\mathrm{PWL}} \sim O(W^2),$$

and

$$A_{\mathrm{poly}} \sim O(dW^2),$$

while the MTJ implementation requires

$$A_{\mathrm{MTJ}} \sim O(n + W_c + W_u).$$

For short event lengths and moderate counter and update widths, the MTJ datapath uses primarily linear-scaling digital blocks. In regimes where these terms are small relative to the general multiplier datapath, the qualitative relation is

$$A_{\mathrm{MTJ}}^{\mathrm{digital}} \ll A_{\mathrm{PWL/poly}}^{\mathrm{digital}}.$$

The third relation states the principal peripheral contribution of the MTJ implementation,

$$P_{\mathrm{MTJ}} = P_{\mathrm{driver}} + P_{\mathrm{SA}} + P_{\mathrm{bias}}.$$

These mixed-signal contributions must be included in a complete hardware comparison. The digital-logic relation alone does not establish an advantage in total area, static power, energy, or latency.

The fourth relation follows directly from the sampling-based operation,

$$T_{\mathrm{MTJ}} = N_{\mathrm{bit}}.$$

Physical sampling therefore links accuracy to runtime and energy, even when function-table storage and deterministic arithmetic are reduced.

4.5 Boundary of the estimation

This primitive-block model compares scaling terms, rather than synthesized circuits in a common technology. Absolute costs require memory and multiplier implementations, transistor sizing, switching current, pulse duration, sensing margins, bias and reference circuits, control logic, and interconnect parasitics.

Implementation choices also change the comparison: shift-friendly slopes reduce PWL arithmetic, serial reuse trades polynomial datapath area for time, and pulse and feedback settings affect MTJ sampling cost. Such choices must be specified before evaluating a particular workload, especially a small fixed 4-bit mapping.

With those choices defined, measured, simulated, or literature-based unit energies can be combined with the actual operation counts through

$$E_{\mathrm{scheme}} = \sum_{i} N_i \, E_i.$$

The architecture thus moves much of the function-storage and multiplier burden into physical sampling and MTJ peripherals. The following calculations illustrate the resulting endurance and timing considerations; they are workload estimates, rather than measurements of an integrated prototype.

4.6 Endurance context for repeated sampling

Repeated sampling makes write endurance part of the workload budget. Related W-based high-barrier SOT-MTJs have sustained more than 1.6×10^{12} write cycles at approximately 2.4 times the critical switching voltage.[23] This literature result provides context; endurance was not measured for the present computing sequence.

Consider a serial workload in which one device generates $N_{\mathrm{bit}} = 10^3$ stochastic bits for each pixel of a 224×224 grayscale image. The physical sampling count per image is

$$224 \times 224 \times 10^3 \approx 5.0 \times 10^7.$$

Counting one write per sampling event gives approximately 3.2×10^4 image workloads at the cited endurance. Counting reset and probabilistic write separately doubles the per-image write count to approximately 1.0×10^8 and gives approximately 1.6×10^4 workloads. These are illustrative per-pixel sampling scenarios.

Distributing the sampling workload across devices can reduce cycling per device. The usable endurance still depends on duty cycle, write amplitude and waveform, temperature, reset protocol, and the specific integrated device.

4.7 Latency and throughput context

The 50 ns write pulse is only one part of a physical sampling cycle. Including reset, readout, sensing, and control gives

$$\tau_{\mathrm{cycle}} = \tau_{\mathrm{reset}} + \tau_{\mathrm{write}} + \tau_{\mathrm{read}} + \tau_{\mathrm{sense}} + \tau_{\mathrm{control}},$$

the first-order time associated with N_{bit} physical stochastic bits is

$$T_{\mathrm{power}} \gtrsim N_{\mathrm{bit}} \tau_{\mathrm{cycle}}.$$

The equivalent estimate $T_{\mathrm{power}} \approx N_{\mathrm{bit}}/R_{\mathrm{bit}}$ requires a full-cycle bit-generation rate R_{bit}. The sample count must also state whether it includes cold-start feedback and post-lock decoding, as discussed in Supplementary Note 7.

A different SOT-MTJ true-random-number generator and its post-processing circuit achieved $R_{\mathrm{bit}} = 16.7$ Mb/s, or approximately 60 ns per output bit.[30] Using that external rate solely as an illustrative full-bit rate gives, for 10^3 bits,

$$\frac{10^3}{16.7 \times 10^6} \approx 60\mu\text{s}.$$

The device, pulse sequence, readout, and probability control differ from the present architecture, so this value does not measure its response time or closed-loop bandwidth. Independent parallel samplers could shorten post-lock output estimation when their samples are statistically independent. Sequential voltage updates in one feedback loop would require a specifically parallel control design to obtain the same speedup.

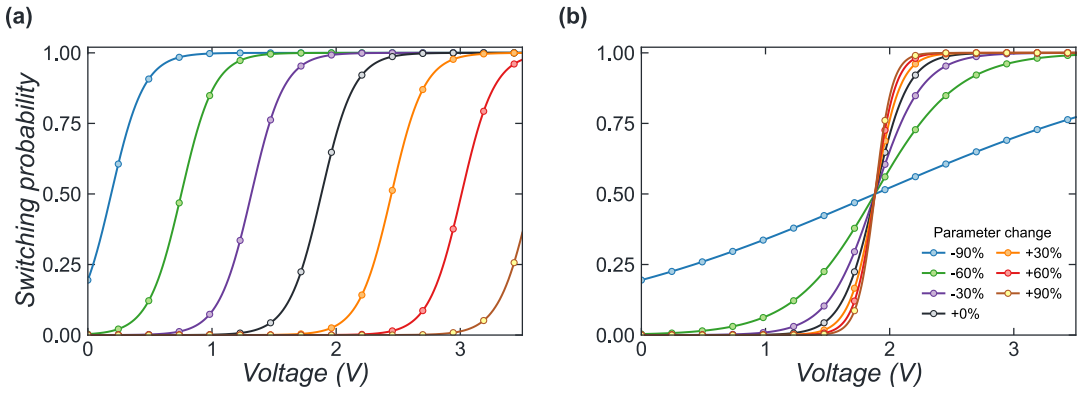
Supplementary Note 5: Simulated Response to Device-Parameter Variation

The simulations examine closed-loop feedback under simultaneous changes in the sigmoid P–V threshold and slope. The parameter domain is centered on the fitted means of 25 measured SOT-MTJ devices. Figure 4e reports the two-dimensional feedback result over ±90% changes in both parameters and overlays the 25 fitted device pairs.

5.1 Parameter-variation model

The switching model is $P(V)=1/[1+\exp(-k(V-V_c))]$, where k is the slope parameter denoted k_s in Supplementary Note 2. Critical-voltage and slope changes are varied independently and simultaneously as $V_c' = V_c \times (1 + \delta Vc)$ and $k' = k \times (1 + \delta ks)$. The two perturbations represent shifts in the operating voltage and changes in switching sensitivity.

The independent variation coefficients δV_c and δk_s span −90% to +90% in 5% increments. The baseline values are the measured means V_c=1.881 V and k_s=7.543 V^{-1}, giving simulated intervals of 0.188–3.575 V and 0.754–14.332 V^{-1}, respectively. Figure S1 illustrates the corresponding P–V curves at representative parameter changes of −90%, −60%, −30%, 0%, +30%, +60%, and +90%.



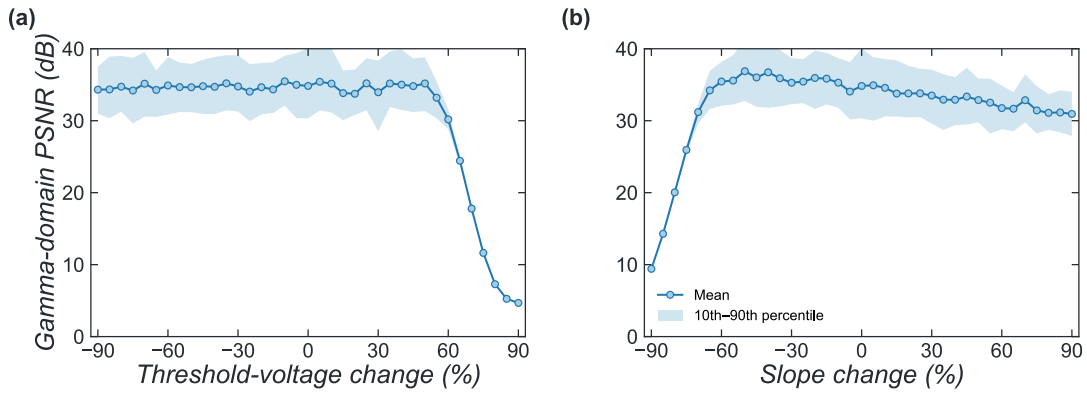
Supplementary Figure S1. Modeled switching-probability curves over the expanded parameter domain. (a) Threshold-voltage variation from −90% to +90% with k_s fixed at its measured mean. (b) Slope variation from −90% to +90% with V_c fixed at its measured mean. The black curves denote the measured-mean baseline.

Alt text: Two sets of modeled switching-probability curves show how device-parameter variation changes voltage response. Changing threshold voltage shifts the sigmoid curve horizontally, while changing its slope alters how sharply switching probability rises. The measured-mean curve provides the reference in each plot.

5.2 Image-quality response and feedback dynamics

At each point in the joint grid, the controller starts from V_0=0 V and uses the same feedback rule with a voltage clamp of ±3.5 V. The image metric is the gamma-domain PSNR relative to the ideal output for the same quantized input codes. The full two-dimensional result is shown in Figure 4e without interpolation or smoothing.

Figure S2 shows cross-sections through the joint grid with the other parameter fixed at its measured mean. The baseline gives 34.83 dB. PSNR remains near the baseline across most threshold-voltage changes before falling sharply above approximately +55%, and it decreases strongly for slope reductions approaching −90%. These trends correspond to the high-V_c and low-k_s boundaries of the low-PSNR region in Figure 4e.



Supplementary Figure S2. Cross-sections of the joint feedback PSNR map. (a) Threshold-voltage variation with k_s fixed at its measured mean and (b) slope variation with V_c fixed at its measured mean. Curves show the mean gamma-domain PSNR, and shading denotes the 10th–90th percentiles across repeated feedback simulations. The voltage is limited to ±3.5 V.

Alt text: Line plots of gamma-domain image fidelity under separate threshold-voltage and slope changes, with shaded simulation ranges. Fidelity stays near its baseline across

most threshold changes but falls sharply at high thresholds; it also drops when the switching-probability slope is greatly reduced.

The pronounced accuracy loss at high V_c or low k_s is caused by the safe-voltage boundary. In these regions, some target probabilities require operating voltages above +3.5 V, so the controller saturates and cannot establish the complete target range. Finite sampling and feedback fluctuations account for residual variation within the voltage-accessible region. The calculations use the fitted sigmoid model; results outside the measured 0.9–3.0 V interval involve extrapolation.

Supplementary Note 6: Image-Quality Metrics and Reconstruction Analysis

The image evaluation addresses two stages of the processing chain. PSNR and SSIM assess how the stochastic gamma output approaches the theoretical gamma target. MAE, RMSE, and error heatmaps assess reconstruction after 4-bit coding. Defining the reference images separately keeps functional accuracy distinct from end-to-end distortion.

6.1 Definition of image quality evaluation metrics

Let I_0 denote the original 8-bit grayscale image normalized to $[0, 1]$, and let $I_{\gamma,\mathrm{th}} = I_0^{5/11}$ denote the corresponding theoretical gamma-transformed image. The hardware gamma output, $I_{\gamma,\mathrm{hw}}$, is generated by mapping each input pixel value through the measured SOT-MTJ code-to-output characteristic. The fidelity of the hardware gamma transformation is evaluated directly in the normalized gamma domain using PSNR and SSIM, with $I_{\gamma,\mathrm{th}}$ as the reference and $I_{\gamma,\mathrm{hw}}$ as the hardware output.

To evaluate the impact on image reconstruction, the gamma-domain outputs are further quantized to 4 bits, decoded, and inverse-gamma transformed to obtain the reconstructed image I_{rec}. Reconstruction accuracy is then assessed in the original intensity domain by comparing I_{rec} with I_0.

For an image with *M* rows and *N* columns, the mean squared error is

$$\mathrm{MSE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij} - Y_{ij}\right)^2$$

With normalized gamma-domain intensities and a maximum value of 1, PSNR is

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{1}{\mathrm{MSE}}\right)$$

Higher PSNR indicates closer agreement with the theoretical gamma target. SSIM additionally measures agreement in luminance, contrast, and structure:

$$\mathrm{SSIM}(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)}$$

Here μ_X and μ_Y are means, σ_X^2 and σ_Y^2 are variances, and σ_{XY} is covariance. The stabilization constants are $C_1 = (K_1 L)^2$ and $C_2 = (K_2 L)^2$, conventionally with $K_1 = 0.01$ and $K_2 = 0.03$; the normalized intensity range gives $L = 1$. Values closer to 1 indicate greater similarity.

Reconstruction error is summarized by the mean absolute error,

$$\mathrm{MAE} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|X_{ij} - Y_{ij}\right|$$

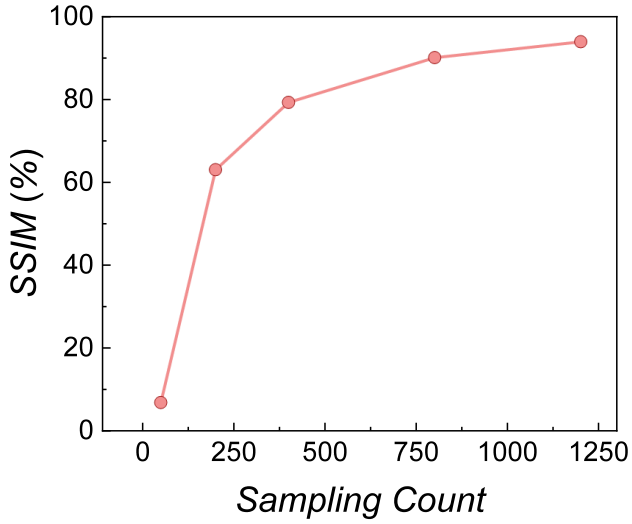
and the root mean square error,

$$\mathrm{RMSE} = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{ij} - Y_{ij}\right)^2}$$

Both use the reconstructed and original images on the 8-bit scale. MAE measures the average absolute deviation, whereas RMSE gives greater weight to large local errors. Their reference image differs from that used for gamma-domain PSNR and SSIM.

6.2 Convergence with raw-bit sampling length

Figure S3 complements the PSNR convergence in Figure 4b by evaluating SSIM against the same theoretical gamma target. Here sampling length counts individual physical trials in the raw SOT-MTJ bitstream.



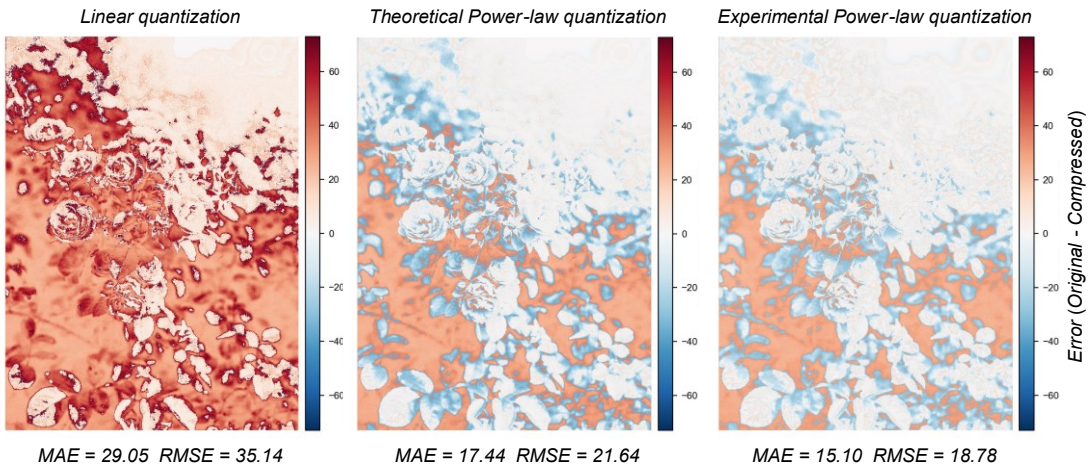
Supplementary Figure S3. Gamma-domain SSIM versus raw-bit sampling length. SSIM compares the normalized output derived from SOT-MTJ bitstreams with the theoretical gamma image. It reaches 0.95 at approximately 1200 samples.

Alt text: Line graph showing structural similarity between images reconstructed from magnetic tunnel junction bitstreams and the theoretical gamma-corrected image. Similarity increases with the number of raw samples and reaches approximately 0.95 at about 1,200 samples.

Together, the SSIM trend and the PSNR approaching 35 dB in Figure 4b indicate improving agreement with the target power-law transform as sampling increases. The observed PSNR is specific to this comparison, rather than a universal perceptual threshold. Reconstruction after 4-bit coding is evaluated separately below.

6.3 Reconstruction errors after 4-bit coding

Figure S4 compares linear quantization, theoretical gamma-corrected quantization, and quantization using the SOT-MTJ-derived gamma relation for the same test image. Each map shows reconstructed intensity minus original intensity on the 8-bit scale.



Supplementary Figure S4. Reconstruction-error maps for three 4-bit quantization methods. Left to right: linear quantization, theoretical gamma correction, and SOT-MTJ-derived gamma correction. All errors are evaluated in the original intensity domain; the gamma paths include an inverse transformation after decoding. Red indicates a darker reconstruction, blue a brighter reconstruction, and white nearly zero error. MAE values are 29.05, 17.44, and 15.10, and RMSE values are 35.14, 21.64, and 18.78, respectively, on the 0–255 scale.

Alt text: Three reconstruction-error maps compare linear quantization, theoretical gamma correction, and gamma correction derived from magnetic tunnel junction data. Both gamma methods reduce errors in the flower image relative to linear quantization, and the device-derived method has the lowest mean absolute and root-mean-square errors.

The linear-quantization map shows larger negative errors in dark regions. Both gamma-based mappings reduce these errors by assigning more quantization resolution to low intensities, consistent with their lower MAE and RMSE in Figure S4. For this image and quantization setting, the SOT-MTJ-derived mapping also gives slightly lower reconstruction errors than the discretized theoretical gamma mapping.

This ordering may reflect the interaction between finite-bit quantization and the image's gray-level distribution. Once mapped values are assigned to 16 output levels, reconstruction error depends on where those values fall within the quantization intervals as well as on their agreement with the continuous gamma curve. The result supports

gamma-corrected reconstruction for this example; a broader image dataset is needed to assess whether the observed ordering generalizes.

Supplementary Note 7: Sampling Cost and Statistical Precision

Sampling cost involves several distinct counts: the exponent denominator n, feedback-block count K, physical-trial count N_{bit}, and output-group count G. Digital input bit depth B is a separate representation parameter. A feedback block contains n trials, giving $N_{\mathrm{bit,fb}}=nK$. A decoder using G nonoverlapping m-bit groups consumes $N_{\mathrm{bit,out}}=mG$ raw bits. Total-cost accounting must state whether feedback and output estimation reuse the same trials and whether the cold-start interval is included.

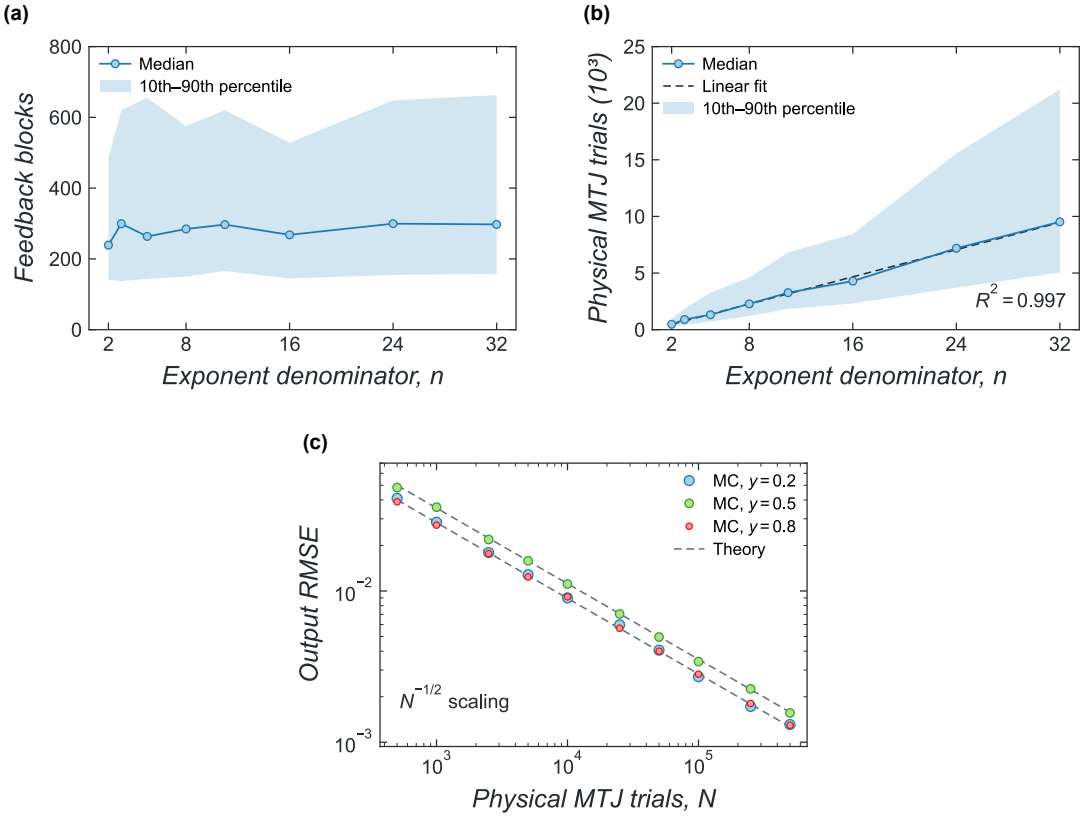
At the ideal interior fixed point, the feedback all-1 event has probability $[b^{1/n}]^n=b$. Increasing n therefore increases the physical trials per update without making that equilibrium event rarer. The number of updates required for convergence can still depend on the target, initial voltage, response curve, learning rate, voltage limits, and stopping rule. A trend with n under one parameterization is consequently not a universal convergence law.

Under a fixed-probability, independent-group approximation, an m-bit output event has probability $y=p^m$ and

$$\mathrm{Var}(\hat{y}) = \frac{y(1-y)}{G} = \frac{my(1-y)}{N_{\mathrm{bit,out}}}.$$

For fixed m, the root-mean-square sampling error scales as $G^{-1/2}$, equivalently $N_{\mathrm{bit,out}}{}^{-1/2}$. As in Section 2.5, this independent-group model describes an accuracy–sampling trade-off; finite-step feedback and temporal dependence can add uncertainty.

Figure S5 illustrates these relations with parameterized Monte Carlo results. Panels (a) and (b) use V_c=2 V, k_s=8 V^{-1}, l=50 mV, V_0=0 V, b=0.5, and n=2–32. The median feedback-block count is approximately constant over this range, while physical-trial count follows a linear fit with R^2=0.997. Panel (c) gives RMSE slopes of −0.503, −0.499, and −0.493 for y=0.2, 0.5, and 0.8, respectively, close to the independent-sampling reference.



Supplementary Figure S5. Parameterized Monte Carlo analysis of sampling cost and precision. (a) Median feedback-block count versus denominator n at b=0.5; shading denotes the 10th–90th percentiles. (b) Corresponding physical-trial count, with n trials per feedback block and a linear fit of R^2=0.997. (c) Output RMSE versus the physical sample count used in the supplied simulation for y=0.2, 0.5, and 0.8. The respective log–log slopes are −0.503, −0.499, and −0.493; dashed curves give the independent-Bernoulli reference. The stated sigmoid-model settings are V_c=2 V, k_s=8 V^{-1}, l=50 mV, and cold start V_0=0 V.

Alt text: Plots of simulated feedback sampling cost and output precision. The number of feedback blocks required for convergence remains roughly constant as the exponent denominator grows, while total device samples rise approximately linearly. Output error decreases with sample count at about the inverse square-root rate.