SPIKING LEARNED ISTA FOR INVERSE PROBLEMS WITH FINITE-TIME CONVERGENCE ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep unfolding offers an interpretable framework for learned sparse recovery. Learned Iterative Shrinkage-Thresholding Algorithm (LISTA) unfolds sparse-recovery iterations into a trainable reconstruction network. However, it represents the recovered coefficients as real-valued activations processed by dense multiply-accumulates. By contrast, a spiking implementation can represent them through sparse signed spike trains and replace dense coefficient multiplications with event-triggered accumulations. We therefore propose Spiking LISTA (S-LISTA), which replaces LISTA soft-thresholding with a Signed Soft-Threshold Integrate-and-Fire (SST-IF) unit that uses paired non-leaky integrate-and-fire neurons for the positive and negative branches and coordinate- and stage-specific scales to balance firing-rate resolution and saturation. We prove a finite-time approximation bound for SST-IF, with error proportional to the scale divided by the number of timesteps, and propagate this error through a fixed-depth unfolded network. The resulting bound shows that the spiking error of S-LISTA decreases as the number of timesteps increases. We evaluate S-LISTA on synthetic support recovery, natural-image compressive sensing, compressed-sensing MRI, and low-precision deployment. S-LISTA preserves the support-recovery behavior of its non-spiking counterpart, and after fine-tuning, it closely matches the corresponding backbone on Set11, BSD68, and MRI while remaining competitive with unfolded baselines. Under low-precision deployment, S-LISTA remains stable under reduced weight precision and avoids the sharp failures observed in its non-spiking counterpart under aggressive post-training quantization. Its measured spike activity further suggests that event-driven affine execution can reduce synaptic arithmetic.

1 INTRODUCTION

Compressed sensing recovers sparse signals from a small number of linear measurements. The Iterative Shrinkage-Thresholding Algorithm (ISTA) solves this problem by alternating affine updates with soft-thresholding. Deep unfolding turns such iterative solvers into trainable fixed-depth networks, and Learned ISTA (LISTA) follows this idea by learning the affine maps and thresholds from data (Gregor & LeCun, 2010; Zhang & Ghanem, 2018). This reduces the number of reconstruction stages, but the affine layers still rely on dense multiply–accumulate operations.

Spiking neural networks (SNNs) can reduce synaptic arithmetic through sparse, event-driven computation (Davies et al., 2018; Davidson & Furber, 2021). With binary presynaptic spikes, a synapse either contributes its weight or stays silent, so suitable hardware can replace many multiplications with event-triggered accumulations (Voelker et al., 2020). Standard artificial neural network (ANN)-to-SNN conversion, however, is built around non-negative ReLU activations (Rueckauer et al., 2017). LISTA uses soft-thresholding, which is signed, has a dead zone, and has an unbounded output range (Gregor & LeCun, 2010; Chen et al., 2018). Existing spiking sparse-coding dynamics address related optimization problems, but not finite-depth learned LISTA conversion (Tang et al., 2017).

To convert LISTA into a spiking version, we need a spiking operator for signed soft-thresholding rather than only a spiking replacement for ReLU. We introduce the Signed Soft-Threshold Integrate-and-Fire (SST-IF) unit, which uses paired integrate-and-fire (IF) neurons to encode the positive and negative branches of soft-thresholding. Applied to a trained LISTA network, SST-IF gives Spiking

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

LISTA (S-LISTA), with coordinate- and stage-specific scales setting the representable range and rate resolution.

Our contributions are as follows. First, we introduce S-LISTA as an unfolded sparse-recovery model with signed spiking thresholding. Second, we prove a finite-depth error bound that separates LISTA recovery error, measurement noise, and finite-timestep spiking error. Third, we evaluate S-LISTA on synthetic support recovery, Set11 and BSD68 compressive sensing, and compressed-sensing MRI. Finally, we study low-precision deployment and use spike activity to compare event-driven accumulations with dense multiply–accumulates.

2 Related Work

Deep unfolding for sparse recovery. Classical sparse-recovery solvers such as ISTA and FISTA provide the algorithmic template for unfolding: each iteration combines a linear update with soft-thresholding (Daubechies et al., 2004; Beck & Teboulle, 2009). LISTA turns this template into a learned reconstruction network (Gregor & LeCun, 2010), and later work gives structured conditions under which unfolded ISTA achieves linear convergence (Chen et al., 2018). For image compressive sensing, ISTA-Net extends this idea to learned reconstruction from block measurements (Zhang & Ghanem, 2018). Subsequent models improve reconstruction quality with learned sampling and convolutional denoising or deblocking stages, including OPINE-Net+ (Zhang et al., 2020), AMP-Net (Zhang et al., 2021), and ISTA-Net++ (You et al., 2021), or with Transformer blocks that add non-local context, as in TransCS (Shen et al., 2022) and CSformer (Ye et al., 2023). These architectures form stronger image-reconstruction backbones than LISTA. Our work instead keeps the explicit LISTA soft-thresholding step because it admits a direct signed spiking realization.

Spiking neural networks for inverse problems and low precision. SNNs have been applied to denoising, video restoration, and super-resolution, motivated by their sparse event-driven computation on neuromorphic hardware. SPIDEN trains a deep SNN for Gaussian and speckle denoising (Castagnetti et al., 2023); later work combines spiking and convolutional modules for event-driven video restoration (Cao et al., 2025), remote-sensing super-resolution (Xiao et al., 2025), and ANN-to-SNN conversion for image restoration (Ouyang et al., 2026). These methods target low-level vision architectures, while S-LISTA targets the soft-thresholding operator in unfolded sparse recovery. Spiking sparse-recovery theory usually studies recurrent dynamics, including links between integrate-and-fire firing rates and the locally competitive algorithm (Shapero et al., 2014), convergence to sparse-coding solutions (Tang et al., 2017), and non-convex regularized sparse recovery (Zhang et al., 2022). Low-precision SNN work quantizes synaptic weights for neuromorphic constraints (Gonzalez et al., 2025; Vogginger et al., 2025), while dense accelerator work studies weight and activation quantization under activation outliers (Lin et al., 2024a;b). We analyze finite-rate signed soft-thresholding inside a trained unfolded network and evaluate both weight-only precision and end-to-end NVFP4 quantization.

3 Method

Notation. We denote by $\mathbb{R}$ and $\mathbb{N}$ the sets of real numbers and positive integers, respectively. For any positive integer N, we denote $[N] := \{1, 2, \dots, N\}$. Bold lowercase letters (e.g., $\mathbf{x}$) represent vectors, and bold uppercase letters (e.g., $\mathbf{A}$) represent matrices. For a vector $\mathbf{x} := (x_i)_{i=1}^d$, we define its ℓ_p-norm as $\|\mathbf{x}\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$ for $1 \le p < \infty$, and its ℓ_0 (pseudo-)norm as $\|\mathbf{x}\|_0 := |\{i : x_i \neq 0\}|$, which counts the number of nonzero entries. For any positive integer n, let $\mathbb{R}^n_{++} := \{\mathbf{x} \in \mathbb{R}^n : x_i > 0 \text{ for every } i \in [n]\}$. All vector-valued functions and operations are applied element-wise unless explicitly stated otherwise.

3.1 Motivation

We consider the ill-posed inverse problem of recovering a high-dimensional signal $\mathbf{x}^\star \in \mathbb{R}^d$ from m linear measurements, where $m \ll d$. The acquisition process is modeled as $\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \boldsymbol{\varepsilon}$, with $\mathbf{A} \in \mathbb{R}^{m \times d}$ the sensing matrix and $\boldsymbol{\varepsilon}$ the measurement noise. To overcome underdetermination, a standard prior is sparsity, leading to the LASSO estimator $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x} \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1$, where $\|\cdot\|_1$ serves as the convex relaxation of the ℓ_0-norm and $\lambda > 0$ balances fidelity and sparsity.

ISTA is a classical solver for LASSO. With step size $1/L$ where $L \geq \|\mathbf{A}^\top \mathbf{A}\|_2$ and $\mathbf{x}^{(0)} = \mathbf{0}$, the k-th iteration reads $\mathbf{x}^{(k+1)} = \mathcal{T}_{\lambda/L}\left(\mathbf{x}^{(k)} + \frac{1}{L}\mathbf{A}^\top(\mathbf{y} - \mathbf{A}\mathbf{x}^{(k)})\right)$, where $\mathcal{T}_\alpha(z) = \mathrm{sign}(z)(|z| - \alpha)_+$ is the soft-thresholding operator. By defining $\mathbf{W}_1^{(k)} = \mathbf{I} - \frac{1}{L}\mathbf{A}^\top\mathbf{A}$ and $\mathbf{W}_2^{(k)} = \frac{1}{L}\mathbf{A}^\top$, ISTA admits the compact affine-nonlinear form

$$\mathbf{x}^{(k+1)} = \mathcal{T}_{\lambda/L}\left(\mathbf{W}_1^{(k)}\mathbf{x}^{(k)} + \mathbf{W}_2^{(k)}\mathbf{y}\right). \tag{1}$$

LISTA unrolls K iterations from (1) to K layers and treats $\mathbf{W}_1^{(k)}$, $\mathbf{W}_2^{(k)}$, and $\theta^{(k)} := \lambda/L$ as learnable parameters, which adapts the reconstruction dynamics to data. Algorithm unrolling can improve reconstruction quality with far fewer iterations than the original solver. However, it does not generally inherit the convergence guarantees of its underlying iterative algorithm once the algorithmic parameters are replaced by learned weights (Monga et al., 2021; Chen et al., 2018; Zheng et al., 2023). SNNs, particularly when deployed on neuromorphic hardware, offer the potential for energy-efficient, event-driven computation (Davies et al., 2018; Furber et al., 2014). Directly training SNNs is complicated by the non-differentiability of the spike-generation function, which is commonly handled with surrogate-gradient methods (Neftci et al., 2019; Zenke & Vogels, 2021). To connect these directions, we propose an unfolded spiking architecture in which each iteration is implemented with spiking neurons.

3.2 SPIKING LISTA

Dictionary-domain sparsity. Natural image patches are generally not sparse in the pixel domain. We therefore represent the signal through an (over)complete synthesis dictionary $\mathbf{D} \in \mathbb{R}^{d\times D}$, writing $\mathbf{x}^\star = \mathbf{D}\mathbf{c}^\star$, where $\mathbf{c}^\star \in \mathbb{R}^D$ is sparse in the transform domain. Substituting this representation into $\mathbf{y} = \mathbf{A}\mathbf{x}^\star + \boldsymbol{\varepsilon}$ gives $\mathbf{y} = \mathbf{A}\mathbf{D}\mathbf{c}^\star + \boldsymbol{\varepsilon}$. We parameterize $\mathbf{D}$ by $\mathbf{D} := \mathbf{U}\mathbf{V}^\top$ where $\mathbf{U} \in \mathbb{R}^{d\times r}$ and $\mathbf{V} \in \mathbb{R}^{D\times r}$.

Spike-based thresholding. The leaky integrate-and-fire (LIF) neuron model is defined as

$$u_{\mathrm{pre}}(t) = \beta_{\mathrm{IF}}u(t-1) + x(t), \quad s(t) = H(u_{\mathrm{pre}}(t) - \vartheta), \quad u(t) = u_{\mathrm{pre}}(t) - \vartheta s(t), \tag{2}$$

where $x(t)$ is the input sequence, $s(t)$ is the binary spike output, $H(a) = 1$ for $a > 0$ and $H(a) = 0$ for $a \leq 0$, β_{IF} is the leak factor, and ϑ is the firing threshold. We replace the soft-thresholding operator with a rate-coded approximation in the non-leaky, unit-threshold case $\beta_{\mathrm{IF}} = \vartheta = 1$, i.e., the IF unit. For static inverse-problem inputs, each scalar drive to an IF unit is held constant for T timesteps, as in rate-coded ANN-to-SNN conversion (Rathi & Roy, 2023; Li et al., 2021; Wang et al., 2023). Since the spike $s(t)$ takes value in $\{0, 1\}$, synaptic matrix–vector multiplications can be reduced to weight accumulations, potentially enabling energy-efficient deployment on neuromorphic hardware (Voelker et al., 2020; Rathi & Roy, 2023).

For a constant input $x(t) \equiv x$, the average firing rate of an IF neuron, $\bar{s}_T := \frac{1}{T}\sum_{t=1}^{T} s(t)$, converges to the clipped ReLU function $\min(\mathrm{ReLU}(x), 1)$ as the number of timesteps $T \to \infty$. We denote this finite-time firing rate by $\rho_T(x) := \bar{s}_T$, which provides a T-step approximation of that limiting value. Exploiting the decomposition of the standard soft-thresholding operator as $\mathcal{T}_\alpha(z) = \mathrm{ReLU}(z - \alpha) - \mathrm{ReLU}(-z - \alpha)$, we define its spike-based counterpart by replacing each ReLU term with its rate-coded analogue

$$\mathcal{T}^{\mathrm{snn}}_{\alpha,M,T}(x) := M\rho_T\left(\frac{x-\alpha}{M}\right) - M\rho_T\left(-\frac{x+\alpha}{M}\right). \tag{3}$$

We call the unit defined by (3) the **Signed Soft-Threshold Integrate-and-Fire (SST-IF)** unit, since each term is one IF neuron encoding one branch of the soft-threshold, and we refer to $\mathcal{T}^{\mathrm{snn}}_{\alpha,M,T}$ as the SST-IF operator throughout. The scale M sets both the representable range and the rate resolution: larger M reduces saturation but makes the decoded grid coarser, with step size M/T. In the experiments, the coordinate scales are calibrated from the pre-activations of the corresponding trained non-spiking network and are then fine-tuned together with the spiking model. For $\mathbf{z} \in \mathbb{R}^D$ and a positive scale vector $\mathbf{M} \in \mathbb{R}^D_{++}$, the operator is applied coordinatewise:

$$\left[\mathcal{T}^{\mathrm{snn}}_{\alpha,\mathbf{M},T}(\mathbf{z})\right]_i := \mathcal{T}^{\mathrm{snn}}_{\alpha,M_i,T}(z_i), \qquad i \in [D]. \tag{4}$$

The scale vector extends the non-saturated approximation interval separately for each coordinate. This avoids setting every coordinate to the largest scale required anywhere in the layer.

Spiking LISTA. Let $\mathbf{G} := \mathbf{AD} = \mathbf{AUV}^\top \in \mathbb{R}^{m\times D}$ denote the effective sensing dictionary. Then $\mathbf{y} = \mathbf{ADc}^\star + \boldsymbol{\varepsilon} = \mathbf{Gc}^\star + \boldsymbol{\varepsilon}$. Our proposed Spiking LISTA (S-LISTA) iteration, for $k = 0, \dots, K-1$, is

$$\mathbf{c}^{(k+1)} := \mathcal{T}^{\text{snn}}_{\theta^{(k)},\mathbf{M}^{(k)},T}\left(\mathbf{c}^{(k)} + (\mathbf{W}^{(k)})^\top(\mathbf{y} - \mathbf{Gc}^{(k)})\right), \tag{5}$$

where $\mathbf{c}^{(0)} := \mathbf{0}$, $\mathbf{W}^{(k)} \in \mathbb{R}^{m\times D}$, $\theta^{(k)} \geq 0$, and $\mathbf{M}^{(k)} \in \mathbb{R}^D_{++}$ are layer parameters. The reconstructed signal after K layers is $\mathbf{Dc}^{(K)}$. The next section analyzes this architecture as a spiking approximation of LISTA and quantifies how the finite number of timesteps T affects reconstruction accuracy.

3.3 Theoretical analysis

In the following, we analyze the convergence rate of S-LISTA and relate its spike activity to the operation count of dense LISTA evaluation. All proofs and the parameter construction are provided in Appendix A.

S-LISTA convergence. We assume the ground truth $\mathbf{c}^\star$ and noise $\boldsymbol{\varepsilon}$ belong to the set

$$\mathbb{X}(B, s, \delta) := \{(\mathbf{c}^\star, \boldsymbol{\varepsilon}) : \|\mathbf{c}^\star\|_\infty \leq B,\ \|\mathbf{c}^\star\|_0 \leq s,\ \|\boldsymbol{\varepsilon}\|_1 \leq \delta\}. \tag{6}$$

Define the layer stability factor $q_k := \|\mathbf{I} - (\mathbf{W}^{(k)})^\top\mathbf{G}\|_2$. Since $(\mathbf{W}^{(k)})^\top\mathbf{G}$ has rank at most $m < D$, the matrix $\mathbf{I} - (\mathbf{W}^{(k)})^\top\mathbf{G}$ has a unit eigenvalue and hence $q_k \geq 1$ at every layer, so the spiking term accumulates with depth and the guarantee below is a fixed-depth statement rather than a depth-uniform one. The following result propagates the finite-time thresholding error through the network while retaining the stability factor of every layer.

Theorem 3.1 (Finite-depth ϵ-approximation)**.** *Fix $K \geq 1$, $\epsilon > 0$, $B > 0$, $s \in \mathbb{N}$, and $\delta \geq 0$. Consider a linear inverse problem $\mathbf{y} = \mathbf{Gc}^\star + \boldsymbol{\varepsilon}$ where the columns of $\mathbf{G}$ are normalized and $(\mathbf{c}^\star, \boldsymbol{\varepsilon}) \in \mathbb{X}(B, s, \delta)$. Let $\widetilde{\mu} := \widetilde{\mu}(\mathbf{G})$ be the generalized mutual coherence satisfying $0 < \widetilde{\mu} < 1$ and let $C_{\mathbf{G}}$ be a constant that depends only on $\mathbf{G}$ as defined in Appendix A.1. Assume the sparsity level satisfies $s < (1 + 1/\widetilde{\mu})/2$, and set $\gamma := \widetilde{\mu}(2s-1)$ and $C_{\text{LISTA}} := 2sC_{\mathbf{G}}/(1-\gamma)$. Set*

$$\overline{q} := \sqrt{D(D-1)}\,\widetilde{\mu}, \qquad R_\epsilon := \max\{B\sqrt{s},\ sB + C_{\text{LISTA}}\delta + \epsilon\},$$
$$\overline{M}_\epsilon := B + \sqrt{D-1}\,\widetilde{\mu}R_\epsilon + C_{\mathbf{G}}\delta, \qquad \overline{A}_{K,\epsilon} := \sqrt{D}\,\overline{M}_\epsilon \sum_{j=0}^{K-1} \overline{q}^{\,j}.$$

Choose $M_i^{(k)} := \overline{M}_\epsilon$ for all $i \in [D]$ and $0 \leq k < K$. Then, for any $T \geq \max\left\{1, \lceil \overline{A}_{K,\epsilon}/\epsilon \rceil\right\}$, and every $1 \leq k \leq K$,

$$\|\mathbf{c}^{(k)} - \mathbf{c}^\star\|_2 \leq sB\gamma^k + C_{\text{LISTA}}\delta + \epsilon. \tag{7}$$

Theorem 3.1 turns the conditional estimate of Theorem A.2, stated and proved in Appendix A, into a constructive finite-depth approximation statement. The constants $\overline{q}$, R_ϵ, and $\overline{M}_\epsilon$ respectively control the worst-case layer amplification, the region in which the S-LISTA iterates are kept, and the scale needed to prevent saturation of the spiking threshold. The quantity $\overline{A}_{K,\epsilon}$ aggregates the largest possible propagated spiking error over the first K layers. Choosing $T \geq \overline{A}_{K,\epsilon}/\epsilon$ makes this propagated error at most ϵ, so the S-LISTA reconstruction follows the LISTA rate up to the prescribed finite-time tolerance. The theorem assumes non-saturation, whereas the empirical calibration sets $M_i^{(k)}$ to a high percentile of the observed pre-activations and therefore allows a small fraction of coordinates to saturate. These saturated coordinates introduce an additional clipping error not covered by the worst-case bound; in our image experiments this fraction is about 1% after calibration, and 2.7% after fine-tuning, which updates $\mathbf{M}$. The parameterization differs in the same way. The iteration (5) leaves $\mathbf{W}^{(k)}$ free, and Theorem 3.1 fixes it to the coherence-minimizing set $\mathcal{X}(\mathbf{G})$ of Appendix A.1. The trained image models instead use the tied choice $\mathbf{W}^{(k)} = \beta\mathbf{G}$ with a single learned step size, which generalizes better under our training protocol (Appendix C.3); the bound therefore characterizes the parameterization of Theorem 3.1 rather than the trained backbone.

The convergence result is stated for the sparse coefficient vector because S-LISTA estimates $\mathbf{c}^\star$ before applying the synthesis dictionary. The corresponding signal-domain reconstruction error is

controlled directly by the operator norm of $\mathbf{D}$: $\|\mathbf{D}\mathbf{c}^{(k)} - \mathbf{D}\mathbf{c}^{\star}\|_2 \le \|\mathbf{D}\|_2 \, \|\mathbf{c}^{(k)} - \mathbf{c}^{\star}\|_2$. Thus, when $\mathbf{D}$ is normalized or well conditioned, the coefficient-domain finite-depth guarantee transfers directly to the recovered signal.

Energy interpretation. The preceding results determine how large T must be for S-LISTA to attain a prescribed reconstruction accuracy. We now relate this accuracy guarantee to the computation of the linear part of each LISTA layer. Expanding the preactivation gives

$$\mathbf{z}^{(k)} = \mathbf{H}^{(k)}\mathbf{c}^{(k)} + \mathbf{b}^{(k)}, \qquad \mathbf{H}^{(k)} := \mathbf{I} - (\mathbf{W}^{(k)})^{\top}\mathbf{G}, \qquad \mathbf{b}^{(k)} := (\mathbf{W}^{(k)})^{\top}\mathbf{y}.$$

On a conventional processor, evaluating $\mathbf{H}^{(k)}\mathbf{c}^{(k)}$ requires multiplying entries of $\mathbf{c}^{(k)}$ by entries of $\mathbf{H}^{(k)}$ and adding the products. In S-LISTA, each coefficient is already represented by the positive and negative spike counts of the SST-IF operator. A spike can therefore add or subtract a precomputed, scaled entry of $\mathbf{H}^{(k)}$, replacing multiplication by a spike-triggered addition.

To count these operations, let $d_i^{(k)} := |\{j \in [D] : H_{ji}^{(k)} \neq 0\}|$ be the number of nonzero entries in column i of $\mathbf{H}^{(k)}$. For $k \in [K-1]$, let $S_i^{(k)} \in [T] \cup \{0\}$ be the total number of positive and negative spikes used to represent $c_i^{(k)}$. Define $N_{\mathrm{MAC}} := \sum_{k=1}^{K-1} \sum_{i=1}^{D} d_i^{(k)}$, $N_{\mathrm{AC}} := \sum_{k=1}^{K-1} \sum_{i=1}^{D} d_i^{(k)} S_i^{(k)}$. Here N_{MAC} is the number of multiply–accumulate operations used by conventional LISTA for these matrix–vector products, and N_{AC} is the number of spike-triggered additions used by the corresponding event-driven S-LISTA implementation.

Theorem 3.2 (Fixed-accuracy operation bound)**.** *Fix $K \ge 2$ and $\epsilon > 0$, and choose $\mathbf{M}^{(k)}$ and T according to Theorem 3.1. Then the products $\mathbf{H}^{(k)}\mathbf{c}^{(k)}$, $1 \le k < K$, can be evaluated exactly from the S-LISTA spike trains using N_{AC} spike-triggered additions and no multiplications by the decoded coefficients. Moreover, $N_{\mathrm{AC}} \le T N_{\mathrm{MAC}}$.*

Theorem 3.2 is an operation-count result. Converting it into an energy comparison requires implementation-specific costs. Let e_{MAC} and e_{AC} be the energy of one multiply–accumulate and one spike-triggered addition, and let E^0_{LISTA} and $E^0_{\text{S-LISTA}}$ collect all remaining costs of the two implementations, such as threshold evaluation, neuron updates, rate decoding, and formation of $\mathbf{b}^{(k)}$. Under the model $E_{\mathrm{LISTA}} = E^0_{\mathrm{LISTA}} + e_{\mathrm{MAC}} N_{\mathrm{MAC}}$ and $E_{\text{S-LISTA}} = E^0_{\text{S-LISTA}} + e_{\mathrm{AC}} N_{\mathrm{AC}}$, S-LISTA uses less energy whenever

$$E^0_{\text{S-LISTA}} + e_{\mathrm{AC}} N_{\mathrm{AC}} < E^0_{\mathrm{LISTA}} + e_{\mathrm{MAC}} N_{\mathrm{MAC}}. \tag{8}$$

Thus, Theorem 3.1 controls reconstruction accuracy, while Theorem 3.2 identifies the associated event-driven operation count. Actual energy savings follow by inserting the operation costs of the target hardware into (8). This count describes a spike-stream realization of the affine stages; our implementation decodes the firing rates between stages, so the activity reported in Section 4.3.2 is a proxy for that realization rather than a measurement of it.

4 Experiments and Results

We evaluate S-LISTA from three perspectives. First, we test support recovery on synthetic sparse-recovery problems, where the ground-truth support is known, and the effect of finite-rate spiking can be measured directly. Second, we evaluate image compressive sensing on natural images using Set11 and BSD68, and on compressed-sensing MRI, comparing against classical ISTA and representative learned unfolding baselines. Third, we study deployment-oriented effects, including low-precision quantization and spike activity under an event-driven operation-count proxy. Throughout, A-LISTA denotes the non-spiking network from which S-LISTA is converted, and FT abbreviates fine-tuning.

4.1 Support recovery

Support recovery gives a controlled test of the spiking soft-threshold operator because the true sparse support is known. We draw s-sparse signals $\mathbf{x}^{\star} \in \mathbb{R}^d$ with uniformly random support $\Omega^{\star}$ and observe $\mathbf{y} = \mathbf{A}\mathbf{x}^{\star} + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim \sigma\mathcal{N}(\mathbf{0}, \mathbf{I}_m)$. Nonzero magnitudes are sampled as $|g_i|$, $g_i \sim \mathcal{N}(0,1)$, floored at 0.1, and the signal is ℓ_2-normalized. The signal is sparse in the pixel domain here, so no dictionary is used: $D = d$, $\mathbf{D} = \mathbf{I}$, and $\mathbf{c}^{\star} = \mathbf{x}^{\star}$ in (5). The sensing matrix has i.i.d.

Gaussian entries scaled by $1/\sqrt{m}$ and row-normalized to unit norm; its mutual coherence is 0.449, giving a correlated, $2\times$-undersampled setting. The sparsity condition of Theorem 3.1 is sufficient, not necessary: at this coherence it admits only $s = 1$, whereas we evaluate at $s = 3$ and sweep to $s = 20$, so the experiments also probe the regime beyond the range covered by the bound. We use $d = 128$, $m = 64$, $\sigma = 0.01$, and 16 unfolding stages, train on $10{,}000$ signals, and evaluate on $1{,}000$ held-out signals. The main setting is $s = 3$; Appendix B sweeps the sparsity level.

We evaluate A-LISTA and three S-LISTA variants. A-LISTA is trained with Adam and serves as the non-spiking reference. S-LISTA w/o FT is the direct post-conversion model, using the trained A-LISTA weights, thresholds, and calibrated scales at a finite number of timesteps T. S-LISTA ($T\to\infty$) is the infinite-timestep limit of the same converted model: by Lemma 1, $\rho_T(x)$ converges to $\mathrm{clip}(x, 0, 1)$, so we evaluate this limit by replacing each ρ_T in (3) with $\mathrm{clip}(\cdot, 0, 1)$ without spike simulation or parameter changes. S-LISTA w/ FT is initialized from A-LISTA and fine-tuned separately at each finite T. For evaluation, $\widehat{\Omega}$ is the set of the s largest-magnitude entries of $\widehat{\mathbf{x}}$. We report $\mathrm{NMSE} = \|\widehat{\mathbf{x}} - \mathbf{x}^\star\|_2^2/\|\mathbf{x}^\star\|_2^2$, support-F1 $= |\Omega^\star \cap \widehat{\Omega}|/s$, and exact-support recovery $\mathbf{1}\{\Omega^\star = \widehat{\Omega}\}$. The sparsity level s is used only for evaluation and is not provided to the recovery networks.

Figure 1 shows how support recovery changes with the number of timesteps. Without fine-tuning, S-LISTA improves steadily as T increases and approaches the $T\to\infty$ reference, consistent with the finite-rate approximation analysis. Fine-tuning gives a stronger finite-timestep model: by $T = 50$, S-LISTA w/ FT is already close to A-LISTA and clearly improves over direct post-conversion. We therefore do not fine-tune with larger T values, since longer spike sequences make backpropagation substantially more expensive while the curve suggests little additional gain. Additional seed runs confirm the stability of the post-conversion curve and the $T = 10, 20$ fine-tuned models. The $T = 50$ fine-tuned point is more learning-rate sensitive, so we treat it as an attainable high-performing setting rather than the default operating point.

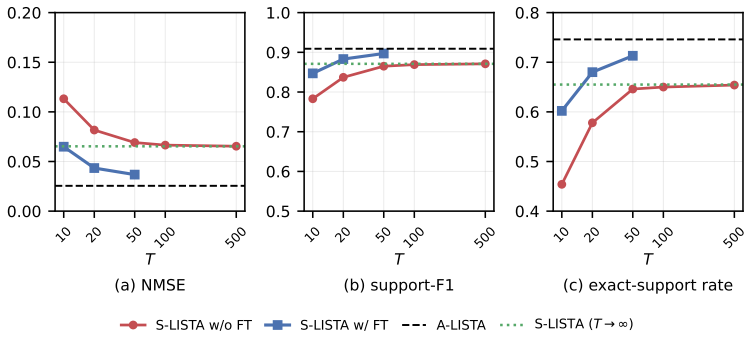


Figure 1: Synthetic recovery at $s = 3$ in `fp32` (d=128, m=64) versus the number of timesteps T. S-LISTA w/o FT and S-LISTA w/ FT (fine-tuned at each T) are both T-dependent and drawn as curves; A-LISTA (the ANN) and S-LISTA ($T\to\infty$, the clip limit) are T-independent and drawn as horizontal dashed lines.

Figure 2 in Appendix B compares A-LISTA, S-LISTA ($T\to\infty$), and S-LISTA w/ FT at $T = 20$ across sparsity levels. The three methods follow the same recovery pattern: exact-support recovery drops once the sparsity becomes too high, while support-F1 degrades more gradually. Before this transition, S-LISTA w/ FT closely tracks A-LISTA and improves over the post-conversion limit on the support metrics; after the transition, it remains competitive in NMSE. Overall, finite-timestep S-LISTA preserves the support-recovery behavior of A-LISTA across the sparsity sweep.

4.2 Image Recovery

We follow the ISTA-Net Set11 protocol: models are trained on 88,912 grayscale 33×33 patches from the 91-image ReconNet training set (Kulkarni et al., 2016; Zhang & Ghanem, 2018). Each patch has dimension $d = 1089$, and the measurement dimension m is set by the sampling ratio, for example $m = 272$ at 25% sampling. We evaluate the same trained models on Set11 and BSD68. All reported image models use the unreduced factorization $r = D = 1361$ and the non-leaky, unit-threshold IF neuron covered by Lemma 1. Appendix C.1 details calibration and backbone con-

vergence. Appendix C.3 reports the rank and leak ablations behind these settings, and Appendix G reports the compressed-sensing MRI setup and results.

For image recovery, A-LISTA uses the dictionary-domain formulation of Section 3, with $\mathbf{G} = \mathbf{AD}$ in the unfolded update. The model has $K = 10$ stages, a jointly learned row-normalized sampling matrix, and the tied update $\mathbf{W}^{(k)} = \beta\mathbf{G}$ with one shared step size β. A-LISTA is trained for 200 epochs with Adam, batch size 64, and learning rate 10^{-4} halved every 50 epochs. S-LISTA is initialized from the corresponding A-LISTA checkpoint, with coordinate scales $M_i^{(k)}$ calibrated per layer as the 99th percentile of $|z_i^{(k)}|$ over $8{,}192$ randomly drawn training blocks. We then fine-tune S-LISTA at $T = 20$ for 200 epochs with Adam, initial learning rate 5×10^{-5} halved every 50 epochs, gradient-norm clipping at 1.0, and a fast-sigmoid surrogate gradient with slope $\kappa = 25$. Fine-tuning updates all model parameters: the sampling matrix, $\mathbf{U}$, $\mathbf{V}$, β, and the per-stage $\theta^{(k)}$ and $\mathbf{M}^{(k)}$. All models are trained with final signal-domain MSE, and PSNR is averaged over each test set.

4.2.1 SET11 AND BSD68 RESULTS

Table 1 reports PSNR at 4%, 10%, and 25% sampling on Set11 and on the 68-image BSD68 test set, which is larger and covers more varied natural scenes. All methods are evaluated under the same protocol, and the image models are the ones trained for Set11, with no retraining for BSD68. ISTA (Daubechies et al., 2004) is the classical baseline, using a DCT sparsifying basis and a tuned regularization parameter. ISTA-Net, ISTA-Net+ (Zhang & Ghanem, 2018), and OPINE-Net+ (Zhang et al., 2020) are evaluated from their official checkpoints, together with two more recent unfolding networks, ISTA-Net++ (You et al., 2021) and AMP-Net-BM (Zhang et al., 2021). The ISTA-Net baselines use fixed sampling matrices, while OPINE-Net+ and our models learn the sampling matrix.

Table 1: Reconstruction PSNR (dB) on Set11 and BSD68. Baselines use released checkpoints under our evaluation protocol; ISTA-Net++ has no released 4% checkpoint. S-LISTA ($T\to\infty$) is the post-conversion limit, and S-LISTA w/ FT is fine-tuned at T=20. Bold marks S-LISTA w/ FT.

	Set11			BSD68		
Model	4%	10%	25%	4%	10%	25%
ISTA (Daubechies et al., 2004)	16.92	19.56	23.90	18.74	20.88	23.82
ISTA-Net (Zhang & Ghanem, 2018)	21.15	26.26	31.85	22.06	25.23	29.07
ISTA-Net+ (Zhang & Ghanem, 2018)	21.56	26.49	32.44	22.34	25.30	29.31
ISTA-Net++ (You et al., 2021)	—	28.36	33.74	—	26.35	30.20
OPINE-Net+ (Zhang et al., 2020)	25.52	29.81	34.81	25.16	27.81	31.50
AMP-Net-BM (Zhang et al., 2021)	25.26	29.40	34.63	25.26	27.86	31.74
A-LISTA	24.02	27.38	30.97	24.39	26.53	29.55
S-LISTA ($T\to\infty$)	23.94	26.76	30.06	24.32	26.42	29.17
S-LISTA w/ FT (T=20)	**24.09**	**27.24**	**30.90**	**24.37**	**26.59**	**29.41**

S-LISTA w/ FT closely matches A-LISTA on both test sets and improves over the infinite-timestep post-conversion model with only 20 timesteps. The same trend appears in structural similarity (SSIM; Appendix E) and across three additional fine-tuning seeds; the observed seed variability does not change the comparisons discussed here (Appendix C.2). Compared with external baselines, S-LISTA w/ FT outperforms ISTA-Net and ISTA-Net+ in the low-measurement regimes and remains close to its own non-spiking backbone. Stronger convolutional unfolding models such as OPINE-Net+ and AMP-Net-BM remain ahead, consistent with their larger nonlinear reconstruction blocks and cross-block processing. Our goal here is to preserve the accuracy of the LISTA-style backbone under spiking execution rather than to set the highest PSNR among image-compressive-sensing models.

Appendix C.3 shows that an untied variant lowers training loss but generalizes worse under this training protocol. Appendix D gives qualitative reconstructions of the Set11 Parrots image; the visual comparison mirrors Table 1, with S-LISTA w/ FT at $T = 20$ closely matching A-LISTA across sampling ratios.

4.3 QUANTIZATION AND OPERATION-COUNT RESULTS

We study deployment efficiency through low-precision arithmetic and spike activity. The quantization experiments cover weight-only quantization for image recovery and MRI, including NVFP4, the 4-bit floating-point format of NVIDIA Blackwell GPUs (`e2m1` values with one `e4m3` scale per block of 16), and end-to-end NVFP4 quantization in synthetic support recovery. The final experiment reports spike activity as an operation-count proxy for event-driven affine execution.

4.3.1 WEIGHT QUANTIZATION

In the image weight-precision sweep of Table 2 we quantize the weights only. For each matrix parameter $\mathbf{P} \in \{\mathbf{U}, \mathbf{V}, \mathbf{A}\}$, inference uses the fake-quantized copy $\widetilde{\mathbf{P}} = \mathrm{dequant}(\mathrm{quant}(\mathbf{P}))$, while activations, decoded spike rates, and scalar parameters remain in `fp32`. This sweep therefore measures weight-format error, not a fully low-precision implementation. For `fp16`, `bf16`, and the two 8-bit formats `e4m3`/`e5m2`, quantization is a round-trip cast through the target format; for `fp4`, we use `e2m1` with per-tensor absolute-maximum scaling.

Table 2 reports Set11 PSNR for A-LISTA and S-LISTA w/ FT at $T = 20$, using one row per bit width. The 16-bit and 8-bit rows summarize the range over the formats tested at that precision; the full per-format Set11 and BSD68 sweep is in Appendix F. Both models are nearly lossless at 8 bits. At `fp4`, S-LISTA is more stable: it keeps PSNR monotone as the sampling ratio increases and avoids the sharp high-ratio drop observed for A-LISTA. The MRI quantization results in Appendix G.4 show the same pattern. This robustness is consistent with the bounded SST-IF output in (3), which limits how far weight perturbations can propagate through the unfolded network.

Table 2: Weight-precision sweep on Set11 (PSNR, dB); only the matrix parameters $\mathbf{U}, \mathbf{V}, \mathbf{A}$ are quantized. Ranges combine formats with the same bit width. S-LISTA denotes S-LISTA w/ FT at T=20. Bold marks the `fp4` S-LISTA values.

		4%		10%		25%	
Format	bits	A-LISTA	S-LISTA	A-LISTA	S-LISTA	A-LISTA	S-LISTA
`fp32`	32	24.02	24.09	27.38	27.24	30.97	30.90
`fp16/bf16`	16	24.02	24.08–24.09	27.38	27.24	30.97	30.90
`fp8`	8	23.92–23.99	23.96–24.07	27.25–27.35	27.10–27.22	30.80–30.92	30.74–30.85
`fp4 (e2m1)`	4	23.23	**23.46**	26.59	**26.25**	25.57	**29.21**

Table 3 compares `fp32` and NVFP4 performance in the synthetic support-recovery setting. The table lists five models. Two are non-spiking: A-LISTA trained in `fp32`, and A-LISTA trained with the NVFP4 fake quantizer in the forward pass. Three are spiking: S-LISTA obtained by direct post-training conversion of the `fp32` A-LISTA (w/o FT), S-LISTA fine-tuned at each number of timesteps in `fp32` (w/ FT), and S-LISTA w/ FT trained with the NVFP4 quantizer in the forward pass. For the models trained in `fp32`, the `fp32` row is the unquantized result and the NVFP4 row is the same checkpoint quantized afterwards; the two models trained in NVFP4 have no `fp32` entry.

In the post-training rows, affine operands and decoded intermediates are fake-quantized and dequantized back to float, while thresholds, membrane states, spikes, and scales remain high precision. S-LISTA is more tolerant to this post-training NVFP4 quantization than A-LISTA, retaining more of its `fp32` accuracy across the tested timesteps. NVFP4-aware training recovers much of the loss for both models, so S-LISTA's advantage is clearest when an existing `fp32` checkpoint is quantized without retraining.

4.3.2 SPIKE ACTIVITY AND OPERATION-COUNT PROXY

We report spike activity to estimate the arithmetic cost of a spike-stream affine implementation. For an affine layer with $N_{\mathrm{syn}} = n_{\mathrm{in}} n_{\mathrm{out}}$ synapses, mean activity f spikes per input coefficient per timestep gives approximately fTN_{syn} spike-triggered accumulations. Relative to dense ANN evaluation, the synaptic arithmetic cost is

$$\frac{E^{\mathrm{syn}}_{\text{S-LISTA}}}{E^{\mathrm{syn}}_{\text{LISTA}}} \approx fT \frac{e_{\mathrm{AC}}}{e_{\mathrm{MAC}}}. \tag{9}$$

Table 3: Synthetic recovery under `fp32` and NVFP4 at $s = 3$ (d=128, m=64). Entries with T use T=10/20/50. Unless marked trained in NVFP4, NVFP4 results are post-training quantization of `fp32` checkpoints. Bold marks post-training NVFP4 S-LISTA entries.

Model	Format	NMSE ↓	support-F1 ↑	exact ↑
A-LISTA trained in `fp32`	`fp32`	0.0255	0.909	0.746
	NVFP4	0.1459	0.831	0.539
A-LISTA trained in NVFP4	NVFP4	0.0432	0.843	0.568
S-LISTA w/o FT (T=10/20/50)	`fp32`	0.1128/0.0816/0.0690	0.783/0.837/0.864	0.454/0.576/0.642
	NVFP4	**0.1220/0.0915/0.0950**	**0.782/0.817/0.835**	**0.448/0.519/0.556**
S-LISTA w/ FT (T=10/20/50)	`fp32`	0.0652/0.0436/0.0370	0.847/0.880/0.895	0.602/0.673/0.707
	NVFP4	**0.0845/0.0706/0.0716**	**0.816/0.838/0.828**	**0.517/0.562/0.536**
S-LISTA w/ FT trained in NVFP4 (T=10/20/50)	NVFP4	0.0715/0.0523/0.0475	0.820/0.839/0.845	0.531/0.574/0.586

Using 45 nm estimates $e_{\mathrm{MAC}} = 4.6\,\mathrm{pJ}$ and $e_{\mathrm{AC}} = 0.9\,\mathrm{pJ}$ (Horowitz, 2014), the proxy favors the spiking layer when $fT < e_{\mathrm{MAC}}/e_{\mathrm{AC}} \approx 5.1$. This is an operation-count proxy rather than an end-to-end energy measurement: it excludes memory access, neuron-state updates, comparisons, routing, static power, and control overhead. Our current implementation also decodes firing rates between affine stages, so (9) estimates the arithmetic cost of a future spike-stream affine implementation rather than the measured runtime energy of the present hybrid model.

Table 4 reports activity on Set11 patches at 25% sampling. The useful operating point is the fine-tuned short-timestep model: S-LISTA w/ FT reaches the best PSNR at $T = 20$ while remaining below the dense-affine proxy. Post-training conversion needs many more timesteps to approach the same reconstruction quality, and its proxy ratio then exceeds dense evaluation. Fine-tuning increases activity and saturation relative to direct conversion at the same T, mainly by lowering the threshold and scale so that more coefficients fire within the short timestep budget.

Table 4: Measured spike activity and synaptic-arithmetic proxy on Set11 at 25% sampling. f counts spikes per coefficient and timestep over both SST-IF branches. "Saturated" is the fraction of coordinates outside the calibrated representable range. Bold marks the lowest proxy ratio and the highest PSNR.

Model	T	f	fT	Proxy ratio	Saturated	PSNR
S-LISTA w/o FT	20	0.0512	1.02	**0.200**	1.0%	26.49
S-LISTA w/o FT	100	0.0546	5.46	1.067	1.0%	29.78
S-LISTA w/o FT	500	0.0553	27.66	5.411	1.0%	30.03
S-LISTA w/ FT	20	0.0819	1.64	0.321	2.7%	**30.90**

5 Conclusion

We introduced SST-IF, a signed rate-coded approximation of soft-thresholding, and used it to convert LISTA-style unfolded sparse-recovery models into S-LISTA. SST-IF uses paired non-leaky integrate-and-fire neurons for the positive and negative branches, with coordinate- and stage-specific scales controlling range and rate resolution. Our analysis bounds the finite-timestep approximation error and propagates it through a fixed-depth unfolded network. Experiments show that S-LISTA preserves the behavior of its non-spiking backbone: it tracks A-LISTA in synthetic support recovery, stays close to A-LISTA on Set11 and BSD68 after fine-tuning, and transfers to compressed-sensing MRI. The deployment results show tolerance to post-training NVFP4 quantization, near-lossless 8-bit image weights, and low spike activity under the synaptic-arithmetic proxy. The current implementation decodes rates between affine stages; a spike-stream affine implementation would test the predicted arithmetic savings directly. SST-IF could also be combined with stronger unfolded reconstruction backbones.

REPRODUCIBILITY STATEMENT

The synthetic pipeline of Section 4.1 specifies the signal model, sensing matrix, noise level, depth, and train/test splits, and the image protocol of Section 4.2 specifies the dataset, patch count, architecture size, optimizer, schedule, calibration rule, surrogate gradient, and the parameters updated during fine-tuning. Appendix C.1 adds the calibration sample size and the convergence check of the non-spiking backbone, Appendix C.2 reports the seed-to-seed variation of the fine-tuned image models, and Appendix G.1 gives the corresponding MRI setup. The quantization procedures, including which tensors are quantized and which remain in `fp32`, are stated in Section 4.3.1 and Appendix G.4. All datasets are public: the 91-image ReconNet training set, Set11, BSD68, and the brain MRI benchmark of Zhang & Ghanem (2018); every external baseline is evaluated from its released checkpoint under the protocol of Section 4.2. The assumptions of Theorems 3.1 and 3.2 are given in full in their statements, and the complete proofs, together with the LISTA-CP parameter construction they rely on, are in Appendix A.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript writing and editing, including improving readability, restructuring experiment descriptions, translating author-written notes into English, and drafting manuscript text from author-provided notes, tables, figures, and experimental outputs. We also used generative AI tools to assist with research code workflows, including writing or editing parts of the code, running author-specified commands, summarizing command outputs, and summarizing reported experimental results for presentation in the manuscript. We did not use generative AI tools to generate datasets, fabricate experimental results or figures, formulate mathematical claims, provide proof ingredients, or make autonomous research decisions. All figures are produced by author-written plotting scripts from the recorded experimental outputs. All AI-assisted text, code, command outputs, and suggestions were reviewed and revised by the authors against the source code, experimental outputs, tables, figures, and proofs. AI-assisted code and command executions were inspected and verified by the authors before being used in the reported experiments. The authors take responsibility for the final content of this work, including text, claims, code, results, and artifacts produced with the aid of generative AI.

REFERENCES

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009. doi: 10.1137/080716542.

Chengzhi Cao, Xueyang Fu, Yurui Zhu, Zhijing Sun, and Zheng-Jun Zha. Event-Driven Video Restoration With Spiking-Convolutional Architecture. *IEEE Transactions on Neural Networks and Learning Systems*, 36(1):866–880, January 2025. ISSN 2162-2388. doi: 10.1109/TNNLS.2023.3329741. URL `https://ieeexplore.ieee.org/document/10314003/`.

Andrea Castagnetti, Alain Pegatoquet, and Benoît Miramond. SPIDEN: deep Spiking Neural Networks for efficient image denoising. *Frontiers in Neuroscience*, 17, August 2023. ISSN 1662-453X. doi: 10.3389/fnins.2023.1224457. URL `https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2023.1224457/full`.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004. doi: 10.1002/cpa.20042.

Simon Davidson and Steve B. Furber. Comparison of artificial and spiking neural networks on digital hardware. *Frontiers in Neuroscience*, 15:651141, 2021. doi: 10.3389/fnins.2021.651141.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic

manycore processor with on-chip learning. *IEEE Micro*, 38(1):82–99, 2018. doi: 10.1109/MM.2018.112130359.

Steve B. Furber, Francesco Galluppi, Steve Temple, and Luis A. Plana. The SpiNNaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014. doi: 10.1109/JPROC.2014.2304638.

Ivan Gonzalez et al. SpikeFit: Towards optimal deployment of spiking networks on neuromorphic hardware. *arXiv preprint arXiv:2510.15542*, 2025. URL https://arxiv.org/abs/2510.15542.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, pp. 399–406, 2010.

Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 10–14. IEEE, 2014. doi: 10.1109/ISSCC.2014.6757323.

Kuldeep Kulkarni, Suhas Lohit, Pavan Turaga, Ronan Kerviche, and Amit Ashok. ReconNet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 449–458, 2016.

Yuhang Li, Shikuang Deng, Xin Dong, Ruihao Gong, and Shi Gu. A free lunch from ANN: Towards efficient, accurate spiking neural networks calibration. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6316–6325, 2021.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024a. URL https://arxiv.org/abs/2306.00978.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. QServe: W4A8KV4 quantization and system co-design for efficient LLM serving. *arXiv preprint arXiv:2405.04532*, 2024b. URL https://arxiv.org/abs/2405.04532.

Vishal Monga, Yuelong Li, and Yonina C. Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *IEEE Signal Processing Magazine*, 38(2):18–44, 2021. doi: 10.1109/MSP.2020.3016905.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63, 2019. doi: 10.1109/MSP.2019.2931595.

Yang Ouyang, Zihan Cheng, Xiaotong Luo, Guoqi Li, and Yanyun Qu. SpikingIR: A Novel Converted Spiking Neural Network for Efficient Image Restoration. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(10):8215–8223, March 2026. ISSN 2374-3468. doi: 10.1609/aaai.v40i10.37769. URL https://ojs.aaai.org/index.php/AAAI/article/view/37769.

Nitin Rathi and Kaushik Roy. DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 34(6):2924–2937, 2023. doi: 10.1109/TNNLS.2021.3111897.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11:682, 2017.

Samuel Shapero, Mengchen Zhu, Jennifer Hasler, and Christopher J. Rozell. Optimal sparse approximation with integrate and fire neurons. *International Journal of Neural Systems*, 24(5):1440001, 2014. doi: 10.1142/S0129065714400012.

Minghe Shen, Hongping Gan, Chao Ning, Yi Hua, and Tao Zhang. TransCS: A transformer-based hybrid architecture for image compressed sensing. *IEEE Transactions on Image Processing*, 31:6991–7005, 2022. doi: 10.1109/TIP.2022.3217365.

Ping Tak Peter Tang, Tsung-Han Lin, and Mike Davies. Sparse coding by spiking neural networks: Convergence theory and computational results. *arXiv preprint arXiv:1705.05475*, 2017.

Aaron R. Voelker, Daniel Rasmussen, and Chris Eliasmith. A spike in performance: Training hybrid-spiking neural networks with quantized activation functions. *arXiv preprint arXiv:2002.03553*, 2020.

Bernhard Vogginger et al. Hardware-aware fine-tuning of spiking q-networks on the SpiNNaker2 neuromorphic platform. *arXiv preprint arXiv:2507.23562*, 2025. URL https://arxiv.org/abs/2507.23562.

Ziqing Wang, Yuetong Fang, Jiahang Cao, Qiang Wang, Zhongrui Xu, and Renjing Xu. SSF: Accelerating training of spiking neural networks with stabilized spiking flow. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Yi Xiao, Qiangqiang Yuan, Kui Jiang, Wenke Huang, Qiang Zhang, Tingting Zheng, Chia-Wen Lin, and Liangpei Zhang. Spiking Meets Attention: Efficient Remote Sensing Image Super-Resolution with Attention Spiking Neural Networks, 2025. URL https://arxiv.org/abs/2503.04223. Version Number: 3.

Dongjie Ye, Zhangkai Ni, Hanli Wang, Jian Zhang, Shiqi Wang, and Sam Kwong. CSformer: Bridging convolution and transformer for compressive sensing. *IEEE Transactions on Image Processing*, 32:2827–2842, 2023. doi: 10.1109/TIP.2023.3274988.

Di You, Jingfen Xie, and Jian Zhang. ISTA-Net++: Flexible deep unfolding network for compressive sensing. In *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2021.

Friedemann Zenke and Tim P. Vogels. The remarkable robustness of surrogate gradient learning for instilling complex function in spiking neural networks. *Neural Computation*, 33(4):899–925, 2021. doi: 10.1162/neco_a_01367.

Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1828–1837, 2018.

Jian Zhang, Chen Zhao, and Wen Gao. Optimization-inspired compact deep compressive sensing. *IEEE Journal of Selected Topics in Signal Processing*, 14(4):765–774, 2020.

Xiang Zhang, Lei Yu, Gang Zheng, and Yonina C. Eldar. Spiking sparse recovery with non-convex penalties. *IEEE Transactions on Signal Processing*, 70:6272–6285, 2022. doi: 10.1109/TSP.2023.3234460.

Zhonghao Zhang, Yipeng Liu, Jiani Liu, Fei Wen, and Ce Zhu. AMP-Net: Denoising-based deep unfolding for compressive image sensing. *IEEE Transactions on Image Processing*, 30:1487–1500, 2021. doi: 10.1109/TIP.2020.3044472.

Ziyang Zheng, Wenrui Dai, Duoduo Xue, Chenglin Li, Junni Zou, and Hongkai Xiong. Hybrid ISTA: Unfolding ISTA with convergence guarantees using free-form deep neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3226–3244, 2023. doi: 10.1109/TPAMI.2022.3172214.

A PROOFS FOR THE THEORETICAL ANALYSIS

The proof of the following convergence rate result follows the approach in (Rueckauer et al., 2017).

Lemma 1 (Firing-rate approximation)**.** *Let $\rho_T(x)$ be the average firing rate of* (2) *in the non-leaky, unit-threshold case $\beta_{\mathrm{IF}} = \vartheta = 1$, with $u(0) = 0$ and constant input x. For* $\mathrm{ReLU1}\,(x) := \min\{\mathrm{ReLU}(x), 1\}$*,*

$$\sup_{x\in\mathbb{R}} |\rho_T(x) - \mathrm{ReLU1}\,(x)\,| = \frac{1}{T}. \tag{10}$$

In addition, the error equals zero if $x \in (-\infty, 0) \cup (1, \infty)$.

Proof of Lemma 1. For $\beta_{\mathrm{IF}} = \vartheta = 1$, the membrane update is

$$u(t) = u(t-1) + x - s(t). \tag{11}$$

Suppose first that $0 \le x \le 1$. We claim that $0 \le u(t) \le 1$ for every t. If $0 \le u(t-1) \le 1$, then $u_{\mathrm{pre}}(t) \in [0, 2]$. When $u_{\mathrm{pre}}(t) \le 1$, the strict threshold gives $s(t) = 0$ and $u(t) = u_{\mathrm{pre}}(t) \in [0, 1]$. When $u_{\mathrm{pre}}(t) > 1$, it gives $s(t) = 1$ and $u(t) = u_{\mathrm{pre}}(t) - 1 \in (0, 1]$. The claim follows by induction from $u(0) = 0$.

Summing (11) over $t = 1, \ldots, T$ yields

$$\rho_T(x) = \frac{1}{T}\sum_{t=1}^{T} s(t) = x - \frac{u(T)}{T}.$$

Hence $|\rho_T(x) - x| \le 1/T$ for $x \in [0, 1]$.

If $x < 0$, we prove by induction that $u(t) \le 0$ for all t. The base $u(0) = 0$ holds. If $u(t-1) \le 0$, then

$$u_{\mathrm{pre}}(t) = u(t-1) + x \le x < 0,$$

so $s(t) = H(u_{\mathrm{pre}}(t) - 1) = 0$ and $u(t) = u_{\mathrm{pre}}(t) < 0$. Hence $\rho_T(x) = 0$ and $\mathrm{ReLU1}\,(x) = 0$, so the error is 0.

If $x > 1$, we prove by induction that $u(t) \ge 0$ for all t. The base $u(0) = 0$ holds. If $u(t-1) \ge 0$, then

$$u_{\mathrm{pre}}(t) = u(t-1) + x > 1,$$

so $s(t) = H(u_{\mathrm{pre}}(t) - 1) = 1$ and $u(t) = u_{\mathrm{pre}}(t) - 1 > 0$. Thus every step emits a spike, i.e., $\rho_T(x) = 1$, and $\mathrm{ReLU1}\,(x) = 1$, so the error is 0.

At $x = 1$, the first step has $u_{\mathrm{pre}}(1) = 1$ and emits no spike because $H(0) = 0$. Each subsequent step emits one spike, giving $\rho_T(1) = (T-1)/T$. Thus the upper bound is attained. □

Lemma 2 (Spiking-threshold approximation)**.** *Let $\alpha \ge 0$, $M > 0$, and $T \ge 1$. Then*

$$\sup_{x\in[-M-\alpha, M+\alpha]} \left|\mathcal{T}^{\mathrm{snn}}_{\alpha,M,T}(x) - \mathcal{T}_\alpha(x)\right| = \frac{M}{T}. \tag{12}$$

For $\mathbf{M} \in \mathbb{R}^D_{++}$ and $|z_i| \le \alpha + M_i$, coordinatewise application gives

$$\left\|\mathcal{T}^{\mathrm{snn}}_{\alpha,\mathbf{M},T}(\mathbf{z}) - \mathcal{T}_\alpha(\mathbf{z})\right\|_2 \le \frac{\|\mathbf{M}\|_2}{T}. \tag{13}$$

Proof of Lemma 2. If $|x| < \alpha$, both inputs to ρ_T in the SST-IF operator (3) are negative, and both the ρ_T and the SST-IF outputs are zero according to Lemma 1. If $\alpha \le x \le \alpha + M$, only the positive branch is active and $(x - \alpha)/M \in [0, 1]$. Lemma 1 gives

$$\left| M\rho_T\left(\frac{x-\alpha}{M}\right) - M \cdot \frac{x-\alpha}{M}\right| \le \frac{M}{T}.$$

The interval $[-M - \alpha, -\alpha]$ follows by symmetry. Equality holds at $x = \alpha + M$, since $\rho_T(1) = (T-1)/T$.

For a vector input, the scalar estimate gives

$$\left\|\mathcal{T}^{\mathrm{snn}}_{\alpha,\mathbf{M},T}(\mathbf{z}) - \mathcal{T}_\alpha(\mathbf{z})\right\|_2^2 \le \sum_{i=1}^{D} \frac{M_i^2}{T^2} = \frac{\|\mathbf{M}\|_2^2}{T^2}.$$

Taking square roots proves (13). □

The equality case in Lemma 2 also proves that the uniform error of the present rate decoder is not $o(1/T)$. Improving the order in T therefore requires a different temporal code or neural parameter setting.

A.1 LISTA-CP PARAMETER CONSTRUCTION

Let $\mathbf{g}_i$ and $\mathbf{w}_i$ denote the i-th columns of $\mathbf{G}$ and $\mathbf{W}$, respectively. For a matrix with unit-norm columns, define the generalized mutual coherence

$$\widetilde{\mu}(\mathbf{G}) := \inf_{\substack{\mathbf{W}\in\mathbb{R}^{m\times D} \\ \mathbf{w}_i^\top \mathbf{g}_i = 1,\ i\in[D]}} \max_{i\neq j} |\mathbf{w}_i^\top \mathbf{g}_j|. \tag{14}$$

Among the matrices attaining $\widetilde{\mu}(\mathbf{G})$, define

$$C_{\mathbf{G}} := \min_{\mathbf{W} \text{ attaining } \widetilde{\mu}(\mathbf{G})} \max_{a,b} |W_{ab}|, \tag{15}$$

$$\mathcal{X}(\mathbf{G}) := \left\{ \mathbf{W} :\ \mathbf{W} \text{ attains } \widetilde{\mu}(\mathbf{G}),\ \max_{a,b} |W_{ab}| = C_{\mathbf{G}} \right\}. \tag{16}$$

The optimization defining $\widetilde{\mu}(\mathbf{G})$ is a feasible linear program, so these sets are nonempty (Chen et al., 2018).

Let $\widetilde{\mu} := \widetilde{\mu}(\mathbf{G}) > 0$, choose $\mathbf{W}^{(k)} \in \mathcal{X}(\mathbf{G})$, and set

$$\theta^{(k)} = \widetilde{\mu} \sup_{(\mathbf{c}^\star,\boldsymbol{\varepsilon})\in\mathbb{X}(B,s,\delta)} \|\widehat{\mathbf{c}}^{(k)} - \mathbf{c}^\star\|_1 + C_{\mathbf{G}}\delta. \tag{17}$$

Then the LISTA iteration

$$\widehat{\mathbf{c}}^{(k+1)} = \mathcal{T}_{\theta^{(k)}}\left(\widehat{\mathbf{c}}^{(k)} + (\mathbf{W}^{(k)})^\top(\mathbf{y} - \mathbf{G}\widehat{\mathbf{c}}^{(k)})\right), \tag{18}$$

where $\widehat{\mathbf{c}}^{(0)} = 0$, has the following linear convergence rate, which is derived in Chen et al. (2018).

Theorem A.1 (Linear convergence of LISTA)**.** *Consider a linear inverse problem $\mathbf{y} = \mathbf{G}\mathbf{c}^\star + \boldsymbol{\varepsilon}$ where the columns of $\mathbf{G}$ are normalized and $(\mathbf{c}^\star, \boldsymbol{\varepsilon}) \in \mathbb{X}(B, s, \delta)$. Assume the sparsity level satisfies $s < (1 + 1/\widetilde{\mu})/2$, so that $\gamma := \widetilde{\mu}(2s-1) \in (0,1)$. If $\mathbf{W}^{(k)} \in \mathcal{X}(\mathbf{G})$ and $\theta^{(k)}$ satisfies* (17)*, then the sequence $\{\widehat{\mathbf{c}}^{(k)}\}$ generated by LISTA satisfies*

$$\|\widehat{\mathbf{c}}^{(k)} - \mathbf{c}^\star\|_2 \le sB\gamma^k + \frac{2sC_{\mathbf{G}}}{1-\gamma}\delta, \qquad k = 1, 2, \ldots$$

and $\operatorname{supp}(\widehat{\mathbf{c}}^{(k)}) \subseteq \operatorname{supp}(\mathbf{c}^\star)$ *for every k.*

The constant used in the main text is therefore $C_{\mathrm{LISTA}} := 2sC_{\mathbf{G}}/(1-\gamma)$, and the rate is $\gamma = \widetilde{\mu}(2s-1)$; this is the explicit form of Theorem 2 of Chen et al. (2018), whose proof takes the same choice of $\mathbf{W}^{(k)}$ and $\theta^{(k)}$.

A.2 PROOF OF THE S-LISTA ERROR BOUND

Define the LISTA layer map

$$\mathcal{T}_k(\mathbf{u}) := \mathcal{T}_{\theta^{(k)}}\left(\mathbf{u} + (\mathbf{W}^{(k)})^\top(\mathbf{y} - \mathbf{G}\mathbf{u})\right). \tag{19}$$

Soft thresholding is nonexpansive, so

$$\|\mathcal{T}_k(\mathbf{u}) - \mathcal{T}_k(\mathbf{v})\|_2 \le q_k\|\mathbf{u} - \mathbf{v}\|_2, \tag{20}$$

with the layer stability factor q_k of Section 3.3.

Theorem A.2 (Finite-depth S-LISTA error)**.** *Fix $K \geq 1$ and consider the same inverse problem of Theorem A.1. For the S-LISTA preactivation*

$$\mathbf{z}^{(k)} := \mathbf{c}^{(k)} + (\mathbf{W}^{(k)})^\top(\mathbf{y} - \mathbf{G}\mathbf{c}^{(k)}),$$

assume that

$$|z_i^{(k)}| \leq \theta^{(k)} + M_i^{(k)}, \qquad i \in [D], \quad 0 \leq k < K. \tag{21}$$

Then, for $1 \leq k \leq K$,

$$\|\mathbf{c}^{(k)} - \mathbf{c}^\star\|_2 \leq \underbrace{sB\gamma^k}_{\text{LISTA convergence}} + \underbrace{C_{\mathrm{LISTA}}\delta}_{\text{noise floor}} + \underbrace{\frac{1}{T}\sum_{j=0}^{k-1}\|\mathbf{M}^{(j)}\|_2 \prod_{\ell=j+1}^{k-1} q_\ell}_{\text{spiking error}}. \tag{22}$$

An empty product equals one. Appendix A.3 gives a constructive choice of $\mathbf{M}^{(k)}$ and T that guarantees (21) *at any fixed depth.*

Proof of Theorem A.2. Let $\mathbf{e}^{(k)} := \mathbf{c}^{(k)} - \widehat{\mathbf{c}}^{(k)}$ and denote the S-LISTA preactivation by

$$\mathbf{z}^{(k)} := \mathbf{c}^{(k)} + (\mathbf{W}^{(k)})^\top(\mathbf{y} - \mathbf{G}\mathbf{c}^{(k)}).$$

Under (21), Lemma 2 bounds the discrepancy between spiking and soft thresholding at $\mathbf{z}^{(k)}$. Equation (20) then gives

$$\begin{aligned}\|\mathbf{e}^{(k+1)}\|_2 &\leq \left\|\mathcal{T}^{\mathrm{snn}}_{\theta^{(k)},\mathbf{M}^{(k)},T}(\mathbf{z}^{(k)}) - \mathcal{T}_{\theta^{(k)}}(\mathbf{z}^{(k)})\right\|_2 + \|\mathcal{T}_k(\mathbf{c}^{(k)}) - \mathcal{T}_k(\widehat{\mathbf{c}}^{(k)})\|_2 \\ &\leq \frac{\|\mathbf{M}^{(k)}\|_2}{T} + q_k\|\mathbf{e}^{(k)}\|_2. \end{aligned} \tag{23}$$

Because $\mathbf{e}^{(0)} = \mathbf{0}$, induction on (23) yields

$$\|\mathbf{e}^{(k)}\|_2 \leq \frac{1}{T}\sum_{j=0}^{k-1}\|\mathbf{M}^{(j)}\|_2 \prod_{\ell=j+1}^{k-1} q_\ell.$$

Combining this estimate with Theorem A.1 by the triangle inequality proves (22). □

Theorem A.2 is stated with the non-saturation condition (21) because the spiking threshold approximates soft thresholding only on the interval covered by the scale vector $\mathbf{M}^{(k)}$.

A.3 Choosing the constants in Theorem A.2

The two layerwise constants entering Theorem A.2 are the stability factor q_k and the coordinate-wise scale vector $\mathbf{M}^{(k)}$. We first give a simple uniform upper bound on the q_k's under the LISTA construction, and then give an explicit choice of $\mathbf{M}^{(k)}$ that guarantees the condition (21) at fixed depth.

For each layer k, define

$$\mathbf{H}^{(k)} := \mathbf{I} - (\mathbf{W}^{(k)})^\top\mathbf{G}. \tag{24}$$

Then $q_k = \|\mathbf{H}^{(k)}\|_2$. Let $\mathbf{e}_i$ denote the i-th standard basis vector of $\mathbb{R}^D$, and let $\mathbf{w}_i^{(k)}$ denote the i-th column of $\mathbf{W}^{(k)}$.

Lemma 3 (Uniform bound on the layer stability factors)**.** *Suppose that $\mathbf{W}^{(k)} \in \mathcal{X}(\mathbf{G})$. Then*

$$q_k = \left\|\mathbf{I} - (\mathbf{W}^{(k)})^\top\mathbf{G}\right\|_2 \leq \sqrt{D(D-1)}\,\widetilde{\mu}(\mathbf{G}). \tag{25}$$

In particular,

$$q := \sup_{k\geq 0} q_k \leq \sqrt{D(D-1)}\,\widetilde{\mu}(\mathbf{G}).$$

Proof of Lemma 3. By (24), the (i, j)-entry of $\mathbf{H}^{(k)}$ is

$$H_{ij}^{(k)} = \delta_{ij} - (\mathbf{w}_i^{(k)})^\top \mathbf{g}_j = \begin{cases} 0, & i = j, \\ -(\mathbf{w}_i^{(k)})^\top \mathbf{g}_j, & i \neq j. \end{cases}$$

Indeed, the definition of $\mathcal{X}(\mathbf{G})$ gives $(\mathbf{w}_i^{(k)})^\top \mathbf{g}_i = 1$, while every off-diagonal inner product has modulus at most $\widetilde{\mu}(\mathbf{G})$. Thus the diagonal of $\mathbf{H}^{(k)}$ vanishes and every off-diagonal entry has modulus at most $\widetilde{\mu}(\mathbf{G})$. Since the spectral norm is bounded by the Frobenius norm,

$$\begin{aligned} \|\mathbf{H}^{(k)}\|_2 \le \|\mathbf{H}^{(k)}\|_F &= \left(\sum_{i,j} |H_{ij}^{(k)}|^2 \right)^{1/2} \\ &= \left(\sum_{i \neq j} |(\mathbf{w}_i^{(k)})^\top \mathbf{g}_j|^2 \right)^{1/2} \le \left(D(D-1)\widetilde{\mu}(\mathbf{G})^2 \right)^{1/2} \\ &= \sqrt{D(D-1)}\, \widetilde{\mu}(\mathbf{G}). \end{aligned}$$

□

The next lemma gives a constructive coordinatewise scale. It bounds the preactivation $z_i^{(k)}$ by expanding the residual and applying the triangle inequality separately to the signal, operator, and noise terms.

Lemma 4 (Coordinatewise non-saturation)**.** *Fix $R > 0$, and for each $i \in [D]$ define*

$$Z_i^{(k)}(R) := B + \left\| \mathbf{e}_i^\top \mathbf{H}^{(k)} \right\|_2 R + C_{\mathbf{G}} \delta. \tag{26}$$

If $\|\mathbf{u} - \mathbf{c}^\star\|_2 \le R$, then the i-th coordinate of the preactivation

$$\mathbf{z}^{(k)}(\mathbf{u}) := \mathbf{u} + (\mathbf{W}^{(k)})^\top (\mathbf{y} - \mathbf{G}\mathbf{u})$$

satisfies

$$|z_i^{(k)}(\mathbf{u})| \le Z_i^{(k)}(R). \tag{27}$$

Consequently, if $M_i^{(k)} \ge Z_i^{(k)}(R)$ and $\theta^{(k)} \ge 0$, then

$$|z_i^{(k)}(\mathbf{u})| \le \theta^{(k)} + M_i^{(k)}. \tag{28}$$

Proof. Substituting $\mathbf{y} = \mathbf{G}\mathbf{c}^\star + \boldsymbol{\varepsilon}$ and using (24), we rewrite the preactivation as

$$\mathbf{z}^{(k)}(\mathbf{u}) = \mathbf{c}^\star + \mathbf{H}^{(k)}(\mathbf{u} - \mathbf{c}^\star) + (\mathbf{W}^{(k)})^\top \boldsymbol{\varepsilon}.$$

For the i-th coordinate, the triangle inequality gives

$$|z_i^{(k)}(\mathbf{u})| \le |c_i^\star| + \left| \mathbf{e}_i^\top \mathbf{H}^{(k)}(\mathbf{u} - \mathbf{c}^\star) \right| + \left| (\mathbf{w}_i^{(k)})^\top \boldsymbol{\varepsilon} \right|.$$

The first term is at most B because $\|\mathbf{c}^\star\|_\infty \le B$. For the second term, Cauchy–Schwarz and the assumption $\|\mathbf{u} - \mathbf{c}^\star\|_2 \le R$ yield

$$\left| \mathbf{e}_i^\top \mathbf{H}^{(k)}(\mathbf{u} - \mathbf{c}^\star) \right| \le \left\| \mathbf{e}_i^\top \mathbf{H}^{(k)} \right\|_2 \|\mathbf{u} - \mathbf{c}^\star\|_2 \le \left\| \mathbf{e}_i^\top \mathbf{H}^{(k)} \right\|_2 R.$$

For the noise term, Hölder's inequality and $\|\boldsymbol{\varepsilon}\|_1 \le \delta$ yield

$$\left| (\mathbf{w}_i^{(k)})^\top \boldsymbol{\varepsilon} \right| \le \|\mathbf{w}_i^{(k)}\|_\infty \|\boldsymbol{\varepsilon}\|_1 \le C_{\mathbf{G}} \delta.$$

Combining the three bounds gives (27). If $M_i^{(k)} \ge Z_i^{(k)}(R)$, then

$$|z_i^{(k)}(\mathbf{u})| \le Z_i^{(k)}(R) \le M_i^{(k)} \le \theta^{(k)} + M_i^{(k)},$$

where the last inequality uses $\theta^{(k)} \ge 0$. This proves (28). □

The preceding bound only works on the ball $\|\mathbf{c}^{(k)} - \mathbf{c}^\star\|_2 \le R$. Theorem 3.1 closes the argument by choosing R so that every S-LISTA iterate remains in that ball up to any prescribed finite depth.

Proof of Theorem 3.1. Let $\overline{q}$, R_ϵ, $\overline{M}_\epsilon$, and $\overline{A}_{K,\epsilon}$ be as in Theorem 3.1. By Lemma 3, $q_k \le \overline{q}$ for every layer k. The proof of Lemma 3 also gives

$$\left\|\mathbf{e}_i^\top \mathbf{H}^{(k)}\right\|_2 \le \sqrt{D-1}\,\widetilde{\mu}(\mathbf{G}), \qquad i \in [D].$$

Therefore the scale in Theorem 3.1 satisfies

$$M_i^{(k)} = \overline{M}_\epsilon \ge Z_i^{(k)}(R_\epsilon), \qquad i \in [D], \quad 0 \le k < K.$$

We prove by induction that $\|\mathbf{c}^{(k)} - \mathbf{c}^\star\|_2 \le R_\epsilon$ for every $k \le K$. The claim holds at $k = 0$, since

$$\|\mathbf{c}^{(0)} - \mathbf{c}^\star\|_2 = \|\mathbf{c}^\star\|_2 \le B\sqrt{s} \le R_\epsilon.$$

Assume it holds at layer $k < K$. Lemma 4 gives

$$|z_i^{(k)}| \le \theta^{(k)} + M_i^{(k)}, \qquad i \in [D],$$

so the non-saturation condition holds at layer k. The error recursion (23) can therefore be unrolled to obtain

$$\|\mathbf{c}^{(k+1)} - \widehat{\mathbf{c}}^{(k+1)}\|_2 \le \frac{1}{T}\sum_{j=0}^{k} \|\mathbf{M}^{(j)}\|_2 \prod_{\ell=j+1}^{k} q_\ell \le \frac{\overline{A}_{K,\epsilon}}{T} \le \epsilon.$$

Using Theorem A.1,

$$\|\mathbf{c}^{(k+1)} - \mathbf{c}^\star\|_2 \le \|\mathbf{c}^{(k+1)} - \widehat{\mathbf{c}}^{(k+1)}\|_2 + \|\widehat{\mathbf{c}}^{(k+1)} - \mathbf{c}^\star\|_2 \le \epsilon + sB\gamma^{k+1} + C_{\mathrm{LISTA}}\delta \le R_\epsilon.$$

This closes the induction and verifies non-saturation through layer $K-1$. Applying Theorem A.2 and using the same bound $\overline{A}_{K,\epsilon}/T \le \epsilon$ for the spiking term gives (7). □

A.4 Operation-count bound

Proof of Theorem 3.2. Fix $1 \le k < K$. For each input coefficient index i, the coefficient $c_i^{(k)}$ is produced by the spiking-threshold operator at layer $k-1$. Hence, by the signed-rate representation of that operator,

$$c_i^{(k)} = \frac{M_i^{(k-1)}}{T}\sum_{t=1}^{T}\big(s_{i,+}^{(k-1)}(t) - s_{i,-}^{(k-1)}(t)\big). \tag{29}$$

Here the superscript $k-1$ on the spike variables records the layer whose SST-IF operator produced $c_i^{(k)}$. Substituting (29) into output coordinate j, the j-th row of the coefficient-dependent product, gives

$$\big[\mathbf{H}^{(k)}\mathbf{c}^{(k)}\big]_j = \sum_{i=1}^{D}\sum_{t=1}^{T}\frac{H_{ji}^{(k)} M_i^{(k-1)}}{T}\big(s_{i,+}^{(k-1)}(t) - s_{i,-}^{(k-1)}(t)\big).$$

Thus, a positive spike from input coefficient i adds the precomputed value $H_{ji}^{(k)} M_i^{(k-1)}/T$ to row accumulator j, and a negative spike subtracts it. This evaluates $\mathbf{H}^{(k)}\mathbf{c}^{(k)}$ exactly without multiplying by the decoded value $c_i^{(k)}$. For this fixed layer k, a spike generated by coefficient i contributes through column i of $\mathbf{H}^{(k)}$. It therefore updates exactly those output accumulators j for which the column entry $H_{ji}^{(k)}$ is nonzero. By the definition of $d_i^{(k)}$, there are $d_i^{(k)}$ such nonzero entries in column i. Hence each spike from coefficient i triggers $d_i^{(k)}$ additions, and the $S_i^{(k)}$ spikes from that coefficient trigger $d_i^{(k)} S_i^{(k)}$ additions. Summing this count over all coefficients gives the layer-k addition count; summing these layer counts over $k = 1, \dots, K-1$ gives N_{AC}.

Moreover, $\theta^{(k-1)} \ge 0$ implies that the positive- and negative-branch inputs cannot both be positive. A nonpositive input produces no spikes, and the other branch emits at most one spike per time step. Therefore $S_i^{(k)} \le T$, and hence

$$N_{\mathrm{AC}} = \sum_{k=1}^{K-1}\sum_{i=1}^{D} d_i^{(k)} S_i^{(k)} \le T\sum_{k=1}^{K-1}\sum_{i=1}^{D} d_i^{(k)} = T N_{\mathrm{MAC}}.$$

The scales $\mathbf{M}^{(k)}$ and the number of timesteps T are those of Theorem 3.1, so this operation count is the one incurred at the accuracy guaranteed there. □

B ADDITIONAL SYNTHETIC RESULTS

This appendix reports the synthetic sparsity sweep that complements the timestep study of Section 4.1.

B.1 SUPPORT RECOVERY ACROSS SPARSITY LEVELS

Figure 2 shows the full sweep over sparsity levels $s = 2, \ldots, 20$ that Section 4.1 summarizes. The transition of the exact-support metric occurs once s passes about 10; each s uses an independent train/test draw from the same synthetic distribution, so the curves also reflect run-to-run variation.

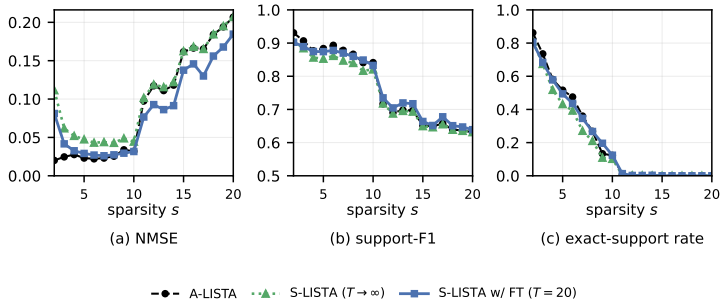


Figure 2: Support recovery versus sparsity s (d=128, m=64) for A-LISTA, S-LISTA ($T \to \infty$), and S-LISTA w/ FT at T=20.

C IMAGE-RECOVERY IMPLEMENTATION DETAILS AND ABLATIONS

C.1 CALIBRATION AND TRAINING DETAILS

Scale calibration. The amplitude scales $\mathbf{M}$ are calibrated before running the spiking network. For each coordinate i and layer k, the scale $M_i^{(k)}$ determines both the representable range of the SST-IF unit and its rate resolution: values outside $[-M_i^{(k)}, M_i^{(k)}]$ saturate, while the decoded rate has step size $M_i^{(k)}/T$. We estimate these scales from the trained A-LISTA activations by running calibration image blocks through the network and setting $M_i^{(k)}$ to the 99th percentile of $|z_i^{(k)}|$ layer by layer. This is a calibration step only and does not involve training.

The number of calibration blocks matters for post-conversion S-LISTA. With one batch of 64 blocks, the 99th percentile is close to a batch maximum, so the resulting scales depend strongly on which patches are sampled. We therefore use a fixed sample of 8,192 randomly drawn training blocks for the reported post-conversion image results, chosen simply as a sample large enough for the percentile estimate to be stable rather than tuned. Only the order of magnitude matters: calibrating on 1,024 blocks instead changes the infinite-timestep PSNR by at most 0.07 dB, so the estimate has already converged well before 8,192. Compared with the earlier 64-block calibration, this raises the infinite-timestep post-conversion PSNR on Set11 by 0.43–0.85 dB and removes the batch-to-batch variation observed with small calibration sets, which reached 0.6 dB at 25% sampling. The measured saturation rate after conversion is 0.9–1.3% across sampling ratios, consistent with the intended 99th-percentile rule. The fine-tuned checkpoints are much less sensitive to this initialization because $\mathbf{M}$ is updated during fine-tuning.

A-LISTA convergence. A-LISTA is trained for 200 epochs before conversion. The backbone is already converged at this point: on Set11, its PSNR changes by at most 0.04 dB over the last 100 epochs at 4% sampling, is unchanged from epoch 150 at 10%, and has flat training loss from epoch 100 at 25%. Thus the S-LISTA fine-tuning stage starts from a converged non-spiking backbone rather than from an undertrained A-LISTA model.

C.2 SEED VARIANCE

We repeated the S-LISTA fine-tuning with three additional seeds, which change both the data order and the calibration batch. The standard deviation across seeds is at most 0.03 dB at every sampling ratio on both test sets (24.05 ± 0.03, 27.28 ± 0.01 and 30.85 ± 0.03 dB on Set11; 24.38 ± 0.01, 26.60 ± 0.01 and 29.51 ± 0.01 dB on BSD68), and the checkpoint reported in Table 1 lies within 0.10 dB of the corresponding seed means. The differences discussed in the main text are therefore larger than seed noise.

C.3 ARCHITECTURE ABLATIONS

Tied versus untied layers. The reported image models tie the update matrix as $\mathbf{W}^{(k)} = \beta\mathbf{G}$, with one shared step size across stages. We also trained an untied variant in which each stage learns its own dictionary factors and effective operator, which raises the parameter count from 3.5M to 33.5M. Both variants use the same data, schedule, and number of epochs. Untying reaches a lower final training loss at every sampling ratio ($0.0028/0.0014/0.0006$ versus $0.0033/0.0017/0.0008$ for the tied model at $4/10/25\%$) but a lower test PSNR on both test sets, as Table 5 shows. The gap widens with the sampling ratio, from 0.70 to 1.77 dB on Set11 and from 0.28 to 0.84 dB on BSD68: the additional per-stage parameters are used to fit the training patches rather than to improve reconstruction. Under this training protocol the tied parameterization therefore acts as a useful regularizer, and the spiking model inherits the tied backbone.

Table 5: Tied versus untied image models (PSNR, dB). Both are A-LISTA backbones trained for 200 epochs under the protocol of Section 4.2; the tied row is the A-LISTA row of Table 1.

		Set11			BSD68		
Parameterization	Params	4%	10%	25%	4%	10%	25%
Tied ($\mathbf{W}^{(k)} = \beta\mathbf{G}$)	3.5M	24.02	27.38	30.97	24.39	26.53	29.55
Untied (per-stage)	33.5M	23.32	26.00	29.20	24.11	26.10	28.71

Dictionary rank. Section 3 writes the dictionary as $\mathbf{D} = \mathbf{U}\mathbf{V}^\top$ with a general rank $r \leq \min\{d, D\}$. For the 33×33 patch experiments, we trained the dictionary model at several ranks between 128 and D and did not find the low-rank factorization to improve reconstruction. All reported image models therefore keep the unreduced factorization $r = D = 1361$; note that $\mathbf{D}$ itself has rank at most $\min\{d, D\} = d$. The factorized form is retained because it can reduce storage and computation when $r \ll \min\{d, D\}$, which is more relevant for larger patch sizes.

Leak in the IF neuron. The neuron model in (2) is written with a general leak and firing threshold. The image experiments use the non-leaky, unit-threshold setting covered by Lemma 1. To check this choice, we evaluated leaky variants at inference while keeping the trained model fixed. At 25% sampling and $T = 20$, reducing the IF leak β_{IF} from 1.00 to 0.99, 0.95, 0.90, and 0.80 changes S-LISTA w/ FT PSNR from 30.90 to 30.87, 29.96, 27.76, and 24.36 dB, respectively. For the post-conversion model, the corresponding PSNR values are 26.49, 25.84, 23.41, 20.66, and 16.16 dB. The same ordering holds at 4% and 10% sampling. We therefore use the non-leaky case in all reported image experiments. The IF leak β_{IF} is distinct from the unfolded update step size β in $\mathbf{W}^{(k)} = \beta\mathbf{G}$.

D QUALITATIVE IMAGE RECONSTRUCTIONS

Figure 3 shows reconstructions of the Set11 Parrots image at 4%, 10%, and 25% sampling. The direct back-projection contains little recognizable structure, especially at low sampling ratios. A-LISTA restores the main image content, and S-LISTA w/ FT at $T = 20$ produces visually similar reconstructions across all three ratios.

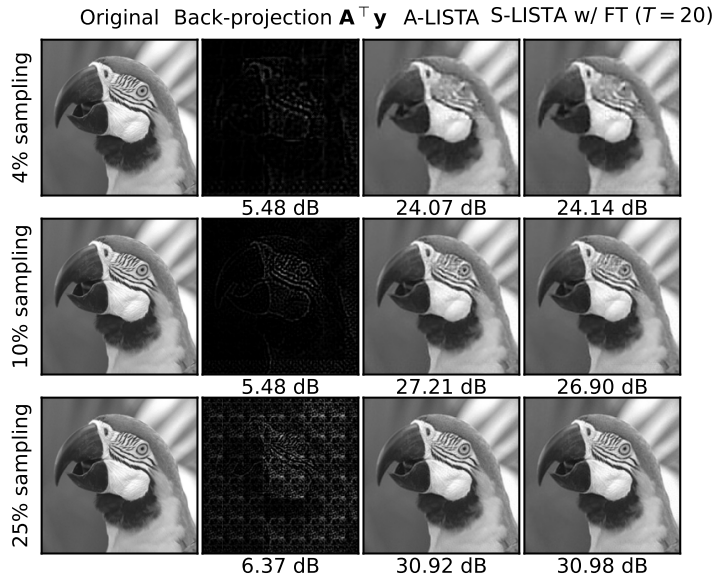


Figure 3: Set11 reconstructions of the Parrots image at 4%, 10%, and 25% sampling. Columns show the original image, compressed back-projection $\mathbf{A}^\top\mathbf{y}$, A-LISTA reconstruction, and S-LISTA w/ FT reconstruction at $T = 20$. PSNR is shown below each reconstruction.

E STRUCTURAL SIMILARITY

Table 6 reports SSIM for the models of Table 1, computed with the same call as the ISTA-Net test scripts (`structural_similarity` with a data range of 255 on the luminance channel). S-LISTA w/ FT stays within 0.004 of A-LISTA at every ratio on both test sets, so the spiking conversion preserves structural fidelity as well as PSNR.

Table 6: Reconstruction SSIM on Set11 and BSD68, for the models and protocol of Table 1.

	Set11			BSD68		
Model	4%	10%	25%	4%	10%	25%
ISTA	0.3425	0.4875	0.6969	0.3763	0.4711	0.6433
ISTA-Net	0.5961	0.7961	0.9161	0.5430	0.6958	0.8448
ISTA-Net+	0.6240	0.8036	0.9237	0.5573	0.7001	0.8507
ISTA-Net++	—	0.8520	0.9373	—	0.7400	0.8716
OPINE-Net+	0.7879	0.8904	0.9514	0.6841	0.8040	0.9062
AMP-Net-BM	0.7817	0.8876	0.9527	0.6925	0.8113	0.9148
A-LISTA	0.7245	0.8409	0.9083	0.6509	0.7751	0.8787
S-LISTA ($T\to\infty$)	0.7259	0.8301	0.8991	0.6515	0.7729	0.8756
S-LISTA w/ FT (T=20)	0.7233	0.8399	0.9107	0.6469	0.7744	0.8795

F FULL IMAGE WEIGHT PRECISION

Table 7 gives the full per-format weight-precision sweep summarized in Table 2. Only the matrix parameters are quantized; activations, decoded spike rates, and scalar parameters remain in `fp32`. The larger BSD68 test set follows the same pattern as Set11.

Table 7: Full weight-precision sweep on Set11 and BSD68 (PSNR, dB). S-LISTA is the fine-tuned model at T=20.

			4%		10%		25%	
Format	bits	Dataset	A-LISTA	S-LISTA	A-LISTA	S-LISTA	A-LISTA	S-LISTA
fp32	32	Set11	24.02	24.09	27.38	27.24	30.97	30.90
		BSD68	24.39	24.37	26.53	26.59	29.55	29.41
fp16	16	Set11	24.02	24.08	27.38	27.24	30.97	30.90
		BSD68	24.39	24.37	26.53	26.59	29.55	29.41
bf16	16	Set11	24.02	24.09	27.38	27.24	30.97	30.90
		BSD68	24.39	24.37	26.53	26.59	29.55	29.41
fp8 (e4m3)	8	Set11	23.99	24.07	27.35	27.22	30.92	30.85
		BSD68	24.36	24.36	26.51	26.57	29.51	29.38
fp8 (e5m2)	8	Set11	23.92	23.96	27.25	27.10	30.80	30.74
		BSD68	24.31	24.26	26.46	26.48	29.44	29.31
fp4 (e2m1)	4	Set11	23.23	23.46	26.59	26.25	25.57	29.21
		BSD68	23.80	23.89	26.04	25.83	25.49	28.29

G Compressed-sensing MRI

G.1 Setup

To test whether the SST-IF operator transfers to a different forward model, we apply it to compressed-sensing MRI, where the measurement is a k-space mask rather than a dense random matrix,

$$\mathbf{y} = \mathbf{\Phi} \odot \mathcal{F}(\mathbf{x}^\star), \qquad \mathbf{A}^\top \mathbf{A}\,\mathbf{x} = \mathrm{Re}\,\mathcal{F}^{-1}(\mathbf{\Phi} \odot \mathcal{F}(\mathbf{x}))\,, \tag{30}$$

with $\mathcal{F}$ the 2-D Fourier transform, $\mathbf{\Phi}$ the binary k-space sampling mask, and $\mathbf{A}^\top$ the adjoint of the measurement operator, which maps $\mathbf{y}$ to the zero-filled reconstruction. This operator acts on the whole 256×256 image, so a dense dictionary is infeasible and the analysis/synthesis dictionary is realized convolutionally; the per-coordinate scale correspondingly becomes a per-channel scale $\mathbf{M}^{(k)} \in \mathbb{R}^C$. Each stage performs a data-consistency step followed by SST-IF shrinkage of the analysis coefficients. We use 9 stages with $C = 32$ channels, train on the 800 brain images of the ISTA-Net benchmark (Zhang & Ghanem, 2018) and evaluate on its 50-image test set at 10%, 20%, and 30% k-space sampling; these ratios differ from the Set11 grid because they follow the released mask set of that benchmark, whose lowest ratio is 10%. Here A-LISTA and S-LISTA denote this convolutional variant. A-LISTA is trained for 200 epochs (Adam, 10^{-4}); S-LISTA is warm-started from it, calibrated, and fine-tuned for 60 epochs at $T = 20$.

G.2 Reconstruction results

Table 8 reports the MRI results. Post-training conversion improves as the number of timesteps increases and approaches its clip limit, while fine-tuning at $T = 20$ recovers a large part of the conversion gap. The remaining gap is larger than in the Set11 experiments, indicating that saturation matters more for this convolutional MRI variant.

ISTA-Net+ remains ahead on this benchmark. The difference is largely due to the backbone: A-LISTA already trails ISTA-Net+, while S-LISTA w/ FT stays close to its A-LISTA counterpart. The MRI experiment therefore shows that the SST-IF conversion transfers to a Fourier-mask forward model and preserves most of the accuracy of the corresponding continuous network.

G.3 Qualitative reconstructions

Figure 4 shows the corresponding reconstructions. The zero-filled inputs retain the global anatomy but contain blur and aliasing from k-space undersampling. S-LISTA w/ FT restores the main anatomical structure and is visually close to A-LISTA across the tested sampling ratios.

G.4 Weight precision

We repeat the weight-precision study of Section 4.3.1 on the MRI models, adding ISTA-Net+ as a reference. Every parameter tensor of rank ≥ 2 (the convolution kernels, i.e. the MAC operands)

Table 8: Compressed-sensing MRI on the 50-image brain test set (PSNR, dB). S-LISTA ($T\to\infty$) is the post-conversion limit, computed from the trained A-LISTA weights with calibrated per-channel scales and no SNN-specific training; S-LISTA w/ FT is fine-tuned at $T=20$. The reference implementation of Zhang & Ghanem (2018) releases only the ISTA-Net+ variant for CS-MRI, so the plain ISTA-Net row of Table 1 has no counterpart here. Best per column in bold.

Method	10%	20%	30%
ISTA-Net+ (Zhang & Ghanem, 2018)	**34.63**	**38.70**	**40.97**
A-LISTA	32.22	36.51	39.22
S-LISTA ($T\to\infty$)	30.17	34.82	37.59
S-LISTA w/ FT ($T=20$)	30.94	35.47	38.29

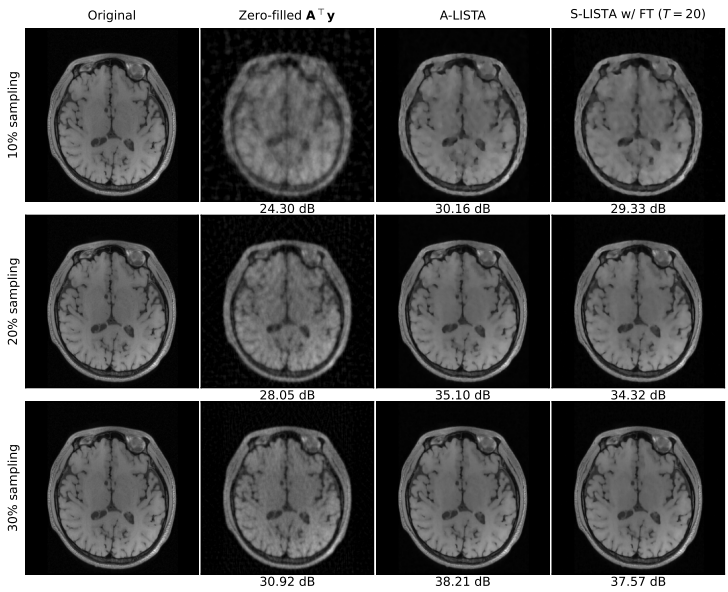


Figure 4: Compressed-sensing MRI reconstructions at 10%, 20%, and 30% k-space sampling (rows). Columns: the original slice, the zero-filled back-projection $\mathbf{A}^\top\mathbf{y}$, the A-LISTA reconstruction, and the S-LISTA w/ FT reconstruction at $T=20$, with PSNR below each reconstruction.

is fake-quantized to the target format; the per-stage scalars (step size, threshold, $\log\mathbf{M}$, and ISTA-Net+'s `soft_thr`) stay in `fp32`, so the sweep isolates weight precision. The table lists both 4-bit formats: `fp4 (e2m1)` applies one absolute-maximum scale per tensor, whereas NVFP4 uses the same `e2m1` grid with one `e4m3` scale per block of 16 values, so the two differ only in the granularity of the scaling.

Table 9: CS-MRI weight-precision sweep (PSNR, dB, 50-image brain test set). S = S-LISTA w/ FT at $T=20$, A = A-LISTA, I+ = ISTA-Net+. Best of the three per column in bold.

		10%			20%			30%		
Format	bits	S	A	I+	S	A	I+	S	A	I+
`fp32`	32	30.94	32.22	**34.63**	35.47	36.51	**38.70**	38.29	39.22	**40.97**
`fp16`	16	30.94	32.22	**34.63**	35.47	36.51	**38.70**	38.29	39.22	**40.97**
`bf16`	16	30.94	32.21	**34.63**	35.47	36.50	**38.70**	38.29	39.22	**40.97**
`fp8 (e4m3)`	8	30.84	31.99	**34.55**	35.31	36.38	**38.62**	38.20	38.30	**40.91**
`fp8 (e5m2)`	8	30.46	31.40	**34.24**	34.36	35.09	**38.37**	37.84	38.30	**40.74**
`NVFP4`	4	28.96	31.25	**33.37**	32.64	34.91	**37.73**	36.03	32.18	**40.17**
`fp4 (e2m1)`	4	28.91	29.54	**32.43**	33.41	32.76	**36.91**	36.69	32.63	**39.40**

Table 9 shows that 8-bit weights are nearly lossless for S-LISTA and ISTA-Net+, and cost A-LISTA at most 0.93 dB. At 4 bits, ISTA-Net+ remains the strongest baseline, while S-LISTA follows the same pattern observed on Set11 and BSD68: it keeps PSNR monotone with the sampling ratio, whereas A-LISTA does not. The precision sweep therefore supports the same conclusion as the image experiments: low-bit weights are usable for S-LISTA, and the spiking model avoids the large high-ratio degradation seen in A-LISTA under aggressive weight quantization.