LEARNING STABLE SPIKE REPRESENTATIONS FOR ROBUST ANN-TO-SNN CONVERSION

Anonymous authors
Paper under double-blind review

ABSTRACT

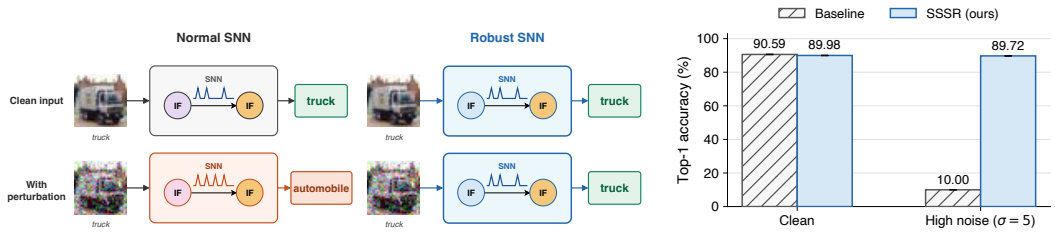
Spiking neural networks (SNNs) enable event-driven computation with the potential for energy-efficient inference, but training deep SNNs remains costly. ANN-to-SNN conversion offers a practical alternative by transferring pretrained artificial neural networks (ANNs) to spiking models. Existing work has largely focused on preserving post-conversion accuracy on clean inputs, while the robustness of converted SNNs to noises remains less understood. In this paper, we ask which properties of the source ANN determine the stability of the resulting SNN. We show that post-conversion stability is governed by the interaction between perturbation amplification and the spacing of discrete spike-count states. This analysis yields a layerwise robustness surrogate and motivates **S**pike-**S**tate **S**tability **R**egularizer (SSSR), a conversion-aware source-training regularizer that requires neither noise injection nor modification of the conversion rule. Across VGG-16 and ResNet-18 on CIFAR-10/100, SSSR improves strong-noise accuracy over the strongest standard L1/L2 baseline by up to 50.17 percentage points, while remaining within 2.16 points of clean accuracy across all four settings. These results show that post-conversion robustness can be shaped directly during the training of source-ANN models. Our code is available at `https://anonymous.4open.science/r/sssr-4E4C/`.

1 INTRODUCTION

Spiking neural networks (SNNs) communicate through discrete spike events and offer the potential for energy-efficient inference on neuromorphic hardware (Maass, 1997; Davies et al., 2018). Deep SNNs can be obtained either by training the spiking network end-to-end using surrogate gradients (Neftci et al., 2019; Wu et al., 2018; Lee et al., 2016), or by converting a pretrained artificial neural network (ANN) (Rueckauer et al., 2017; Sengupta et al., 2019; Diehl et al., 2015). End-to-end SNN training requires temporal unrolling of neuronal dynamics and approximates gradients through the non-differentiable spike function, increasing training complexity and computational cost compared with standard ANN optimization. ANN-to-SNN conversion provides an attractive alternative: it first exploits mature ANN training techniques and then maps the learned network to an SNN for inference. Recent conversion methods have substantially improved the accuracy achievable at low inference latency. For example, Bu et al. (2022b) show that a converted SNN can retain $> 90\%$ of its source ANN's top-1 accuracy using only four SNN inference timesteps.

Recent ANN-to-SNN conversion methods have largely focused on conversion fidelity: preserving the predictive behaviour of the source ANN at low inference latency by reducing quantization, calibration, and temporal mismatch (Bu et al., 2022b; Hao et al., 2023b;a; Bojkovic et al., 2025). Yet clean conversion fidelity does not reveal how stable the resulting spike-count representation is to perturbation. Figure 1 illustrates this distinction: two converted SNNs with similar clean accuracy of around 90% achieve only 10.00% and 89.72% accuracy, respectively, under the strongest tested perturbation on CIFAR-10. Recent works have started to study robustness, but from a complementary direction through perturbation-aware SNN training and adversarially robust conversion (Ding et al., 2024; Ozdenizci & Legenstein, 2024). A more fundamental question, however, remains unresolved: when the source ANN is trained only on clean data, what properties of the source model determine the stability of the converted SNN under internal perturbations, as illustrated in Fig. 1?

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.



(a) Illustration of prediction instability under perturbation. IF: Integrate-and-Fire neuron.

(b) Accuracy on clean and high noise on CIFAR-10.

Figure 1: Clean conversion accuracy can conceal large differences in internal-noise robustness. In (a), perturbation changes the normal SNN's prediction, whereas the robust SNN remains correct. In (b), baseline and SSSR have similar clean accuracy on CIFAR-10, but at high noise ($\sigma = 5$) their accuracies are 10.00% and 89.72%, respectively. Bars show mean $\pm$ standard deviation over five training seeds.

In this paper, we study the stability of finite-time spike-count representations in rate-coded ANN-to-SNN conversion. We show that local perturbation sensitivity depends jointly on two factors: the ***amplification*** induced by the transformation preceding a spiking activation and the ***spacing*** between adjacent representational states. Their interaction yields a margin-normalized sensitivity that links properties of the source model to the stability of the converted representation. By replacing the sample-dependent state margin with the characteristic activation spacing, we derive a tractable layerwise surrogate whose reduction lowers an upper envelope on the phase-averaged state-transition probability.

Guided by this analysis, we propose SSSR, a **S**pike-**S**tate **S**tability **R**egularizer (SSSR) applied to the training of source ANN model. SSSR reduces perturbation amplification relative to representational spacing by jointly adapting the relevant weights and thresholds, while retaining conventional L2 regularization for weights without a corresponding spiking activation. The method requires neither perturbation injection during training nor modification of the ANN-SNN conversion rule. Experiments on CIFAR-10/100 using VGG-16 and ResNet-18 show that SSSR remains close to its clean accuracy under strong perturbations and improves accuracy at the strongest tested perturbation by up to 50.17 percentage points over the strong baselines. Threshold-scaling controls further show that this improvement cannot be explained by uniformly increasing firing thresholds after training. Our contributions are:

- **Robustness Mechanism.** We show that post-conversion robustness is governed by the interplay between perturbation amplification and the spacing of discrete spike-count states, and derive a margin-normalized sensitivity that captures this relationship.
- **Conversion-Aware Training Objective.** We introduce SSSR, a clean-training objective that jointly controls effective weight amplification and spike-count spacing through matched-weight regularization and adaptive thresholds, without noise injection or modifying the conversion procedure.
- **Empirical Gains.** Across VGG and ResNet models on CIFAR-10/100, SSSR improves strong-noise accuracy by up to 50.17 points over standard L1/L2 regularization, with ablations showing that the gains arise from joint weight–threshold adaptation rather than threshold scaling alone.

2 Related Work

Low-latency ANN-to-SNN Conversion. ANN-to-SNN conversion seeks to preserve the functionality of a trained ANN while replacing continuous activations with spike-based computation. Early methods rely on weight normalization, threshold balancing, and activation normalization (Diehl et al., 2015; Rueckauer et al., 2017). More recent work has substantially reduced conversion error at low inference time steps through potential initialization (Bu et al., 2022a), parameter calibration (Li et al., 2021), residual-potential and offset-spike correction (Hao et al., 2023a;b), mitigation of temporal

misalignment (Bojkovic et al., 2025), and quantization-aware source training (Hu et al., 2023; Bu et al., 2022b). Other approaches explicitly reduce the mismatch between quantized ANN activations and their spiking representations (Liu et al., 2022; 2025; Yang et al., 2025). These methods have primarily targeted clean conversion accuracy and finite-time representation error.

Robustness of Spiking and Converted Networks. Robustness in SNNs has been studied from complementary perspectives. Prior work analyzes perturbation propagation and stable membrane-potential dynamics in directly trained SNNs (Ding et al., 2024), as well as stability properties of random spiking networks (Araya et al., 2026). Adversarial robustness has also been incorporated explicitly into ANN-to-SNN conversion (Ozdenizci & Legenstein, 2024). These approaches introduce robustness through the SNN training or conversion procedure. The setting of this paper is different: the source ANN is trained only on clean data, and we study the stability of its finite-time spike-count representation after conversion.

Regularization and Source-Model Robustness. The scope and effect of L2 regularization in normalized networks are known to depend on parameterization. In particular, normalization can make preceding weights scale-invariant, changing the optimization effect of weight decay (Van Laarhoven, 2017), and practical training recipes often exclude biases and BN affine parameters from regularization (He et al., 2019). The treatment of BN-γ can also depend on its architectural role (Kim et al., 2024).

Gap Addressed by This Work. In summary, prior work has advanced ANN-to-SNN conversion, robustness-aware training, and regularization in normalized networks, but leaves a key question open: when the source ANN is trained only on clean data, what determines the stability of the converted spike-count representation? We show that this stability can be traced to properties of the source model, leading to a layerwise robustness surrogate and SSSR, a conversion-aware regularizer that requires neither noise injection nor modification of the conversion rule.

3 From Discrete-State Robustness to Conversion-Aware Regularization

A rate-coded SNN represents a neuron's activation through the number of spikes emitted over a finite number of time steps. A conventional non-leaky integrate-and-fire (IF) neuron illustrates how a continuous input is represented by a finite spike count. With membrane potential m_t, input current z_t, and threshold $\lambda > 0$, the reset-by-subtraction dynamics are

$$\begin{aligned} \widetilde{m}_t = m_{t-1} + z_t, \qquad & s_t = H(\widetilde{m}_t - \lambda), \\ & m_t = \widetilde{m}_t - \lambda s_t, \end{aligned} \tag{1}$$

where $H(\cdot)$ is the Heaviside step function and $s_t \in \{0, 1\}$ denotes the emitted spike. Over T timesteps, the spike count and its rate-coded activation are

$$n_T = \sum_{t=1}^{T} s_t \in \{0, \dots, T\}, \qquad \widehat{a}_T = \frac{\lambda}{T} n_T. \tag{2}$$

Thus, finite-time rate coding can represent only $T + 1$ activation values, spaced by λ/T. The relation between the input current and this discrete representation follows directly by summing Eq. (1) over time:

$$\widehat{a}_T = \frac{1}{T} \sum_{t=1}^{T} z_t - \frac{m_T - m_0}{T}. \tag{3}$$

Hence, the spike count provides a discrete representation of the accumulated input to the neuron, up to the residual membrane-potential term. In ANN-to-SNN conversion, this accumulated input is the temporal counterpart of the preactivation in the source ANN, while the spike-count value represents its corresponding activation (Diehl et al., 2015; Rueckauer et al., 2017; Bu et al., 2022b).

3.1 Local robustness

Let u denote the noiseless effective preactivation presented to the discrete activation at layer ℓ, and let $\widehat{a}_\ell(u)$ denote its represented SNN activation level.

Since $\widehat{a}_\ell(\cdot)$ is discrete, a range of preactivation values will map to the same spike-count level (Fig. 2). Transition boundaries separate neighboring ranges that map to different levels. An additive perturbation δ therefore changes the represented activation only when $u + \delta$ crosses one of these boundaries. On the contrary, perturbations that remain within the same range leave the spike-count representation unchanged.

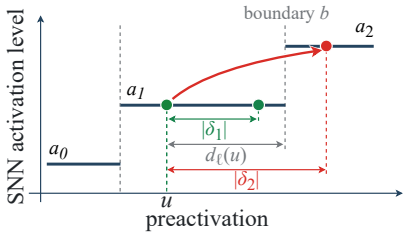


Figure 2: The smaller perturbation, $|\delta_1| < d_\ell(u)$, leaves the represented activation level unchanged; the illustrated rightward perturbation, $|\delta_2| > d_\ell(u)$, crosses the nearest boundary and changes the spike activation level from a_1 to a_2.

This discrete structure provides a natural notion of local representation robustness: the minimum perturbation required to change the represented spike-count level. Given perturbation δ to a local IF neuron, the converted activation becomes non-robust when its activation changes from one desired activation level to another. Let $\mathcal{B}_\ell$ be the preactivation values that separate adjacent activation levels. We define the local representation robustness radius as:

$$d_\ell(u) = \min_{b \in \mathcal{B}_\ell} |u - b|. \tag{4}$$

Any additive perturbation satisfying $|\delta| < d_\ell(u)$ is therefore guaranteed to preserve the represented activation level. Conversely, a representation change becomes possible once the perturbation reaches a transition boundary.

Proposition 1 (Conditional robustness to Gaussian noise)**.** *For fixed u, assume that the propagated perturbation δ is conditionally Gaussian with zero mean and standard deviation $\tau > 0$, i.e., $\delta \mid u \sim \mathcal{N}(0, \tau^2)$. Then*

$$\Pr[\widehat{a}_\ell(u + \delta) \neq \widehat{a}_\ell(u) \mid u] \leq 2Q\left(\frac{d_\ell(u)}{\tau}\right), \tag{5}$$

where $Q(x) = \Pr[Z \geq x]$ for $Z \sim \mathcal{N}(0, 1)$.

Appendix A gives the proof and the exact two-sided interval-exit expression.

Remark 1 (Robustness principle)**.** Proposition 1 shows that local representation stability is governed by the margin-to-perturbation ratio $\rho_\ell(u) = d_\ell(u)/\tau$. For a fixed perturbation scale, increasing the distance to the nearest state-transition boundary reduces the probability of changing the represented spike-count level; for a fixed margin, reducing the propagated perturbation scale has the same effect. Robustness therefore depends jointly on *representation margin* and *perturbation amplification*.

3.2 Layerwise robustness

Proposition 1 shows that local representation robustness depends on the normalized margin $\rho_\ell(u) = d_\ell(u)/\tau$. We next characterize the perturbation scale τ through the preceding transformation and derive a tractable training surrogate.

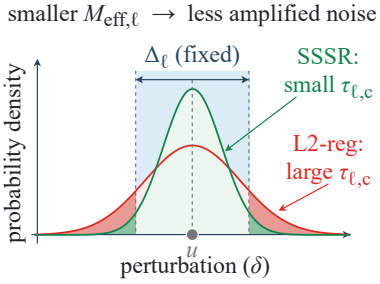


Figure 3: At fixed u and Δ_ℓ, a smaller perturbation standard deviation $\tau_{\ell,c}$ places less probability mass beyond the boundaries.

Consider a convolutional or linear layer ℓ with flattened output-channel weight $W_{\ell,c}$, followed immediately by batch normalization (BN), with channelwise scale $\gamma_{\ell,c}$ and shift $\beta_{\ell,c}$. At inference, the layer and its following BN can be combined into a single transformation through BN folding (Jacob et al., 2018). The resulting BN-folded weight is

$$\widetilde{W}_{\ell,c} = g_{\ell,c} W_{\ell,c}, \qquad g_{\ell,c} = \frac{\gamma_{\ell,c}}{\sqrt{v_{\ell,c} + \epsilon_{\text{BN}}}}, \tag{6}$$

where $v_{\ell,c}$ is the BN running variance; without BN, $g_{\ell,c} = 1$. Suppose independent perturbations $\xi_t \sim \mathcal{N}(0, \sigma^2 I)$ enter this transformation at each timestep. For the time-averaged representation, the resulting preactivation perturbation and its variance are

$$\delta_{\ell,c} = \widetilde{W}_{\ell,c} \frac{1}{T} \sum_{t=1}^{T} \xi_t, \qquad \tau^2_{\ell,c} = \frac{\sigma^2}{T} \|\widetilde{W}_{\ell,c}\|_2^2. \tag{7}$$

Thus, $\tau_{\ell,c}$ is the channel-specific perturbation scale in Proposition 1. Substituting it into the normalized robustness margin gives

$$\rho_{\ell,c}(u) = \frac{\sqrt{T}\, d_\ell(u)}{\sigma \|\widetilde{W}_{\ell,c}\|_2}. \tag{8}$$

Equation (8) exposes the two determinants of local representation robustness: the margin to a state-transition boundary and the perturbation amplification of the preceding transformation.

We summarize the latter at layer level by

$$M_{\text{eff},\ell} = \frac{1}{O_\ell} \sum_{c=1}^{O_\ell} \|\widetilde{W}_{\ell,c}\|_2^2, \tag{9}$$

where O_ℓ is the number of output channels. For completeness, Appendix A.2 derives Eq. (7) by explicitly comparing the clean and perturbed time-averaged preactivations; the folded BN bias is identical in both cases and therefore cancels from their difference.

3.3 A LAYERWISE SURROGATE FOR REPRESENTATION ROBUSTNESS

Equation (8) shows that local representation robustness depends on two quantities: the perturbation amplification of the preceding transformation and the distance of the current preactivation to a state-transition boundary. The latter, $d_\ell(u)$, is sample dependent and therefore cannot be used directly as a fixed layerwise training penalty. We instead seek a sample-independent quantity that captures the same amplification–margin relationship.

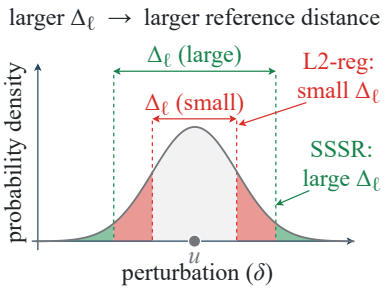


Figure 4: Reference-margin interpretation. Increasing the spacing Δ_ℓ between adjacent activation levels increases the characteristic distance to a state-transition boundary and reduces the probability that a fixed perturbation changes the represented state.

In the non-clipped region, adjacent discrete activation levels are uniformly separated by Δ_ℓ. We use half this spacing, $d_{\text{ref},\ell} = \Delta_\ell/2$, as a characteristic representation margin. Combining it with Eq. (7) gives the channel-wise normalized perturbation variance

$$\frac{\tau_{\ell,c}^2}{d_{\text{ref},\ell}^2} = \frac{4\sigma^2}{T} \frac{\|\widetilde{W}_{\ell,c}\|_2^2}{\Delta_\ell^2}. \tag{10}$$

For fixed perturbation severity σ and number of time steps T, the model-dependent term is therefore the ratio between effective perturbation gain and squared activation-level spacing.

Definition 1 (Layerwise margin-normalized sensitivity)**.** For layer ℓ with discrete activation spacing Δ_ℓ, we define

$$\mathcal{S}_\ell = \frac{M_{\text{eff},\ell}}{\Delta_\ell^2}, \tag{11}$$

where $M_{\text{eff},\ell}$ is the average squared BN-folded weight norm in Eq. (9). For fixed perturbation severity σ and number of time steps T, $\mathcal{S}_\ell$ is proportional to the inverse squared reference robustness margin. It therefore captures perturbation amplification relative to the spacing of the discrete representation: smaller $\mathcal{S}_\ell$ indicates greater local representation stability.

Equivalently, if the perturbation scale across output channels is $\tau_{\text{rms},\ell} = \sigma\sqrt{M_{\text{eff},\ell}/T}$, the corresponding reference normalized margin is

$$\rho_{\text{ref},\ell} = \frac{\Delta_\ell/2}{\tau_{\text{rms},\ell}} = \frac{\sqrt{T}\,\Delta_\ell}{2\sigma\sqrt{M_{\text{eff},\ell}}}, \qquad \mathcal{S}_\ell = \frac{T}{4\sigma^2\rho_{\text{ref},\ell}^2}. \tag{12}$$

Thus, up to the perturbation-dependent constants σ and T, $\mathcal{S}_\ell$ is the inverse squared reference robustness margin. Although $\Delta_\ell/2$ is not the actual margin of every activation, $\mathcal{S}_\ell$ admits a probabilistic interpretation when the position of the preactivation within its quantization interval is averaged.

Proposition 2 (Phase-averaged state-transition bound)**.** *Assume that, within a non-clipped quantization interval of width Δ_ℓ, the preactivation is uniformly distributed and independent of the local perturbation. Under the Gaussian perturbation model of Eq.* (7)*, let $\overline{P}_\ell$ denote the state-transition probability averaged over the within-interval position and output channels. Then*

$$\overline{P}_\ell \le \min\left\{1,\ \sqrt{\frac{2}{\pi}}\,\frac{\sigma}{\sqrt{T}}\sqrt{\mathcal{S}_\ell}\right\}. \tag{13}$$

Appendix A.3 gives the exact phase-averaged crossing probability for a Gaussian perturbation and proves Proposition 2.

Corollary 1 (Training implication)**.** *For fixed σ and T, the upper bound in Eq.* (13) *is non-decreasing in $\mathcal{S}_\ell$. Reducing $\mathcal{S}_\ell$ therefore reduces the theoretical upper envelope of the phase-averaged local state-transition probability.*

3.4 Conversion-aware source-ANN regularization

Corollary 1 suggests a direct training principle: reduce the effective perturbation amplification of each converted layer relative to the spacing of its discrete activation states. Because σ and T are fixed by the perturbation model rather than by the trainable network parameters, we regularize the model-dependent sensitivity $\mathcal{S}_\ell$ directly.

Let $\mathcal{M}$ denote the set of convolutional and linear weight layers associated with a subsequent discrete SNN activation, and let $\mathcal{U}$ contain weight layers for which no corresponding activation-level spacing is defined. For the matched layers, we define

$$\mathcal{R}_M = \sum_{\ell\in\mathcal{M}} \mathcal{S}_\ell = \sum_{\ell\in\mathcal{M}} \frac{M_{\mathrm{eff},\ell}}{\Delta_\ell^2}. \tag{14}$$

We use $\mathcal{S}_\ell$, instead of $\sqrt{\mathcal{S}_\ell}$ in Eq. (13), because both are monotonic in the same layerwise sensitivity, while $\mathcal{S}_\ell$ arises directly from normalized perturbation variance and yields a simple quadratic penalty in the effective weights. The sum across layers is a separable training surrogate; it is not an end-to-end bound on network failure probability.

This objective differs fundamentally from conventional L2 regularization. Standard L2 penalizes the magnitude of the raw weights, whereas Eq. (14) penalizes the BN-folded perturbation gain relative to the spacing of the subsequent discrete representation. Consequently, layers with the same raw weight norm can have different regularization strengths when their effective gains or activation-level spacings differ. The proposed objective is therefore more directly aligned with the local robustness criterion derived above; whether this alignment improves network-level robustness is evaluated empirically in Section 4. For unmatched layers, no corresponding Δ_ℓ is defined. We therefore retain conventional L2 regularization: $\mathcal{R}_U = \frac{1}{2}\sum_{\ell\in\mathcal{U}} \|W_\ell\|_F^2$. The complete source-training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \eta_M \mathcal{R}_M + \eta_U \mathcal{R}_U. \tag{15}$$

Because $M_{\mathrm{eff},\ell}$ is computed from BN-folded weights, $\mathcal{R}_M$ also depends algebraically on the BN scale $\gamma_{\ell,c}$. Empirically, allowing the regularizer to update this scale substantially degrades both clean and noisy accuracy. In SSSR, we therefore stop the gradient of $\mathcal{R}_M$ through γ, while allowing it to update the matched weights and the learned parameter controlling Δ_ℓ. BN scales remain fully trainable through the task loss $\mathcal{L}_{\mathrm{CE}}$, and unmatched weights are regularized through $\mathcal{R}_U$. Section 4 evaluates these gradient-routing choices.

Instantiation with QCFS. We instantiate the proposed framework with quantization clip-floor-shift (QCFS) conversion (Bu et al., 2022b). In QCFS, adjacent activation levels are uniformly spaced by $\Delta_\ell = \lambda_\ell/L$, where λ_ℓ is the learned threshold and L is the number of quantization intervals. Accordingly, Eq. (14) becomes

$$\mathcal{R}_M = \sum_{\ell\in\mathcal{M}} \frac{L^2 M_{\mathrm{eff},\ell}}{\lambda_\ell^2}. \tag{16}$$

Our primary setting uses $T = L$, so the QCFS activation spacing λ_ℓ/L matches the finite-time SNN spike-count spacing λ_ℓ/T. Evaluations with $T \neq L$ reuse the same L-trained checkpoints and

therefore test temporal transfer rather than redefine the training objective. The full QCFS mapping and implementation details are provided in Appendix B.

Relation to Standard L2 Regularization. SSSR in Eq. (14) reduces, under a special case, to the same quadratic form as conventional L2 regularization, but with the regularization scope restricted to convolutional and linear weights rather than all trainable parameters. Specifically, when BN folding is identity and the activation spacing Δ_ℓ is fixed, SSSR becomes a weighted L2 penalty on W_ℓ, while BN affine parameters and learned thresholds λ_ℓ are not directly penalized. In the general case, however, SSSR additionally accounts for BN-folded perturbation amplification and learnable discrete-state spacing, which becomes increasingly important under stronger perturbations. A formal comparison between standard L2, weights-only L2, and SSSR is provided in Appendix A.4.

4 Experiments

Setup and Perturbation Protocol. We evaluate converted VGG-16 and ResNet-18 on CIFAR-10/100 (Krizhevsky, 2009; He et al., 2016), using $L = T = 16$ unless stated otherwise. We adopt rate-uniform inference (Hao et al., 2023a), where each layer completes its T-step spike sequence before propagation to the next layer. To evaluate representation robustness, we inject independent Gaussian noise $\xi_t \sim \mathcal{N}(0, \sigma^2 I)$ after the input IF activation at each timestep; $\sigma = 0$ denotes clean inference. All methods use the same injection site, temporal schedule, and noise realizations.

Baselines and Reporting. We compare SSSR with L2 applied to all trainable parameters (L2-all), weights-only L2 (L2-wo), and weights-only L1 (L1-wo). Main CIFAR results are reported over five training seeds as mean $\pm$ standard deviation. We report top-1 accuracy across noise levels and the average robustness $\overline{A}_{0:5} = \frac{1}{5}\int_0^5 A(\sigma)\, d\sigma$, approximated by the trapezoidal rule. Unless stated otherwise, SSSR uses $\eta_M = 10^{-4}$ and $\eta_U = 5 \times 10^{-4}$ across all settings. Additional training, ablation, and reproducibility details are provided in Appendices B-C.6.

Robustness under Internal Perturbations. Figure 5 compares SSSR with standard clean-trained regularization baselines across VGG-16 and ResNet-18 on CIFAR-10/100. Although all methods achieve comparable accuracy on clean inputs, their robustness diverges sharply as perturbation severity increases. L2-all degrades rapidly, while L2-wo and L1-wo remain stable at low noise but deteriorate under stronger perturbations. In contrast, SSSR maintains accuracy close to its clean performance across the evaluated range and achieves the highest $\overline{A}_{0:5}$ in all four settings. In particular, at $\sigma = 5$, SSSR improves over the strongest standard baseline, L1-wo, by 50.17 and 42.35 percentage points for VGG-16 and by 14.10 and 23.93 points for ResNet-18 on CIFAR-10 and CIFAR-100, respectively. Compared with L2-wo, the gains range from 40.32 to 69.72 points, with only a 1.01-3.48-point reduction in clean accuracy. These results show that post-conversion robustness can be substantially improved through conversion-aware source training. Furthermore, using the same CIFAR hyperparameters, a complementary screen on VGG-13/16/19 and ResNet-18/34 keeps $\sigma = 5$ accuracy within 2.27 points of clean accuracy on both datasets (Appendix Tables 5 and 6).

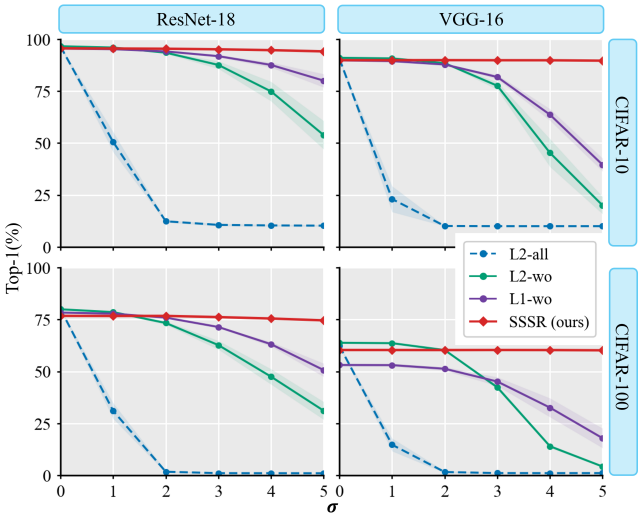


Figure 5: Top-1 test accuracy (%) versus IF Gaussian noise σ at $L = T = 16$. All methods are trained on clean examples. Bands show one sample standard deviation; lines only connect evaluated severities.

Transfer to ImageNet. The ImageNet experiment provides a larger-scale classification check (Figure 6). We train ResNet-18 source ANNs for 90 epochs and evaluate the converted networks on the official validation set with $L = T = 16$ and the same noise location. We use $\eta_M = \eta_U = 10^{-4}$.

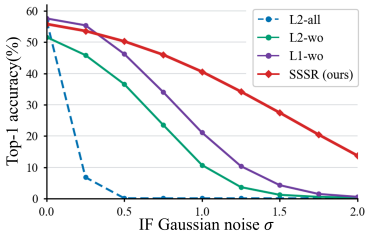


Figure 6: Top-1 accuracy using ImageNet on ResNet-18 at $L = T = 16$.

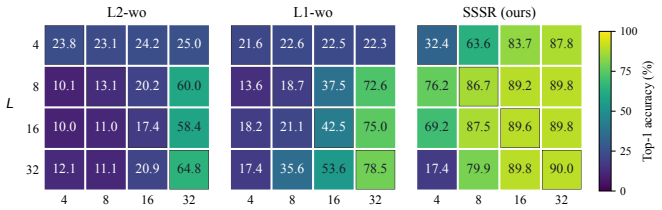
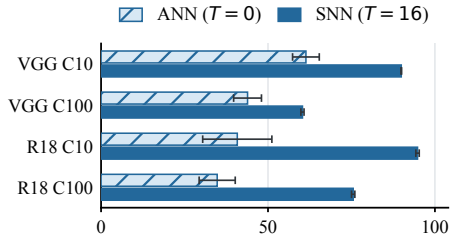


Figure 7: VGG-16 CIFAR-10 top-1 accuracy (%) at $\sigma = 5$ across quantization levels L and inference time steps T. Outlined cells denote $L = T$.

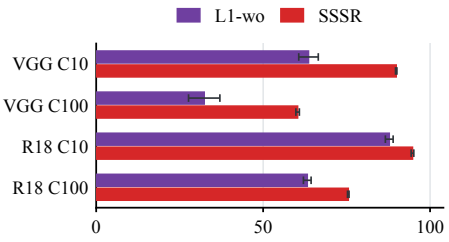
SSSR attains 55.80% clean top-1 accuracy and retains 50.27%, 40.56%, 27.46%, and 13.75% at $\sigma = 0.5, 1, 1.5$, and 2, respectively. The strongest standard baseline, L1-wo, reaches 46.21%, 21.11%, 4.30%, and 0.57% at the same severities, despite having higher clean accuracy (57.57%). L2-wo falls to 36.57%, 10.72%, 1.22%, and 0.31%; L2-all collapses to 0.13% by $\sigma = 0.5$. Thus, clean accuracy alone does not explain the ordering.

Robustness across Quantization Levels and Inference times. We train separate VGG-16 source models at each quantization level $L \in \{4, 8, 16, 32\}$ and evaluate each converted SNN using $T \in \{4, 8, 16, 32\}$ time steps. Figure 7 separates these two axes on CIFAR-10 under $\sigma = 5$ noise. L2-all is omitted because its accuracy is 10.0% in all settings. At fixed $L = 16$, SSSR rises from 69.2% at $T = 4$ to 89.6% at $T = 16$ and 89.8% at $T = 32$. Across all four L values at $T \geq 16$, it reaches 83.7–90.0%; the strongest standard baseline in these cells reaches 78.5%. Appendix C.1 shows accuracy across different quantization levels. The full comparison on CIFAR-100 appears in Appendix Figure 10.

We also compare SSSR as a source ANN and a converted SNN using the same $L = 16$ checkpoints (Figure 8a). At $\sigma = 5$, VGG-16 accuracy rises from 47.27% to 89.72% on CIFAR-10 and from 33.14% to 60.21% on CIFAR-100 after conversion. ResNet-18 similarly rises from 21.94% to 94.20% and from 18.57% to 74.62%, respectively. Thus, the converted SNN is more robust than its source ANN under this noise setting. We then compare regularization methods in the source ANN (Figure 8b). At $\sigma = 1$, SSSR outperforms L1-wo, the strongest baseline, in all four architecture–dataset settings, showing that its robustness advantage is already present before conversion.



(a) Source ANN versus converted SNN at $\sigma = 5$.



(b) Source ANN: SSSR versus L1-wo at $\sigma = 1$.

Figure 8: Top-1 test accuracy at $L = 16$ (average across five random seeds). R18: ResNet-18; C10: CIFAR-10; C100: CIFAR-100. Full results in Appendix C.2.

Layerwise Representation Stability. To examine whether the proposed sensitivity captures the mechanism underlying post-conversion robustness, we measure three quantities across the 14 post-input-IF layers of VGG-16: the layerwise sensitivity $\mathcal{S}_\ell$, the empirical fraction of quantized states that change under perturbation, and the RMS magnitude of the resulting preactivation perturbation. Table 1 reports their layer averages at $L = T = 16$ and $\sigma = 1$.

Across both datasets, SSSR achieves the lowest mean $\mathcal{S}_\ell$ and approximately halves the crossing rate relative to L1-wo: from 8.60% to 4.31% on CIFAR-10 and from 9.58% to 4.85% on CIFAR-100. Importantly, perturbation magnitude alone does not explain this behavior. On CIFAR-10, L2-wo has lower RMS perturbation than L1-wo (2.40 vs. 2.90), yet a higher crossing rate (10.52% vs. 8.60%); the same pattern appears on CIFAR-100 (2.61 vs. 3.00 RMS, but 13.14% vs. 9.58% crossing). L2-all

provides an even stronger contrast: on CIFAR-10 its RMS is almost identical to L1-wo (2.89 vs. 2.90), whereas its mean $\mathcal{S}_\ell$ is over $50\times$ larger and its crossing rate rises to 41.66%. These results support the central prediction of our analysis: representation stability depends on perturbation amplification relative to discrete-state spacing, rather than perturbation magnitude alone. Appendix Figure 13 provides the corresponding layerwise results.

	CIFAR-10			CIFAR-100		
Method	Mean $\mathcal{S}_\ell \downarrow$	Crossing (%) $\downarrow$	RMS $\downarrow$	Mean $\mathcal{S}_\ell \downarrow$	Crossing (%) $\downarrow$	RMS $\downarrow$
L2-all	1210.55 ± 15.93	41.66 ± 1.23	2.89 ± 0.17	1295.40 ± 44.48	39.29 ± 0.85	3.72 ± 0.17
L2-wo	24.70 ± 0.43	10.52 ± 0.26	2.40 ± 0.06	25.17 ± 0.57	13.14 ± 0.16	2.61 ± 0.05
L1-wo	22.52 ± 0.58	8.60 ± 0.07	2.90 ± 0.02	23.04 ± 0.96	9.58 ± 0.40	3.00 ± 0.16
SSSR	7.08 ± 0.02	4.31 ± 0.07	1.20 ± 0.02	11.16 ± 0.19	4.85 ± 0.13	1.18 ± 0.03

Table 1: VGG-16 layerwise diagnostics at $L = T = 16$ and $\sigma = 1$. Crossing: the fraction of quantized states that change; RMS: the preactivation perturbation magnitude.

Ablation Study. Table 2 summarizes both the objective components and their gradient paths. Penalizing BN-folded weights improves strong-noise accuracy over L2-wo, although CIFAR-100 clean accuracy falls. Spacing normalization and full gradients further improve robustness; unmatched-weight L2 mainly helps CIFAR-100. With threshold gradients retained but BN-scale gradients stopped, SSSR achieves the highest $\sigma = 5$ accuracy on both datasets (89.61% and 60.32%). Reversing these gradient paths sharply reduces both clean and noisy accuracy. These single-seed results support the objective design; Appendix C.6 tests whether larger thresholds alone explain the gain.

	Regularizer configuration				CIFAR-10		CIFAR-100	
Matched term	∇_W	∇_λ	∇_γ	$\mathcal{R}_U$	Clean	$\sigma = 5$	Clean	$\sigma = 5$
$\|W\|_F^2$	✓	–	–	✓	91.17	17.44	63.47	4.45
M_{eff}	✓	–	sg	–	89.88	60.41	49.85	30.46
M_{eff}/Δ^2	✓	sg	sg	–	90.50	69.79	59.56	42.82
M_{eff}/Δ^2	✓	✓	✓	–	90.24	82.03	58.69	54.41
M_{eff}/Δ^2	✓	✓	✓	✓	89.97	82.06	60.19	56.27
M_{eff}/Δ^2	✓	sg	✓	✓	80.55	10.14	39.19	1.00
M_{eff}/Δ^2	✓	✓	sg	✓	90.09	**89.61**	61.03	**60.32**

Table 2: Top-1 accuracy on VGG-16 ($L = T = 16$). Matched terms follow Eq. (14); $\mathcal{R}_U$: unmatched-weight L2; ✓: an active regularizer gradient; sg: stop-gradient. Last row is SSSR.

Energy Efficiency. We estimate clean-inference arithmetic cost using 4.6 pJ per 32-bit multiply–accumulate and 0.9 pJ per accumulation (Horowitz, 2014) (Appendix C.7). The SNN estimate counts dense first-layer computation at every timestep and event-driven accumulations in later layers, with five different seeds. It is an operation-based proxy, not measured hardware energy.

At $T = 16$, SSSR has the lowest estimated arithmetic energy among the evaluated VGG-16 SNNs, 13.4% and 13.5% below L1-wo on CIFAR-10 and CIFAR-100, respectively (Table 3). On ResNet-18, its estimate is below L2-all and L2-wo but above L1-wo on both datasets. Across the four architecture–dataset settings, SSSR uses 73.3–78.7% less estimated arithmetic energy than the corresponding dense ANN at $T = 16$.

Method	CIFAR-10	CIFAR-100
VGG-16		
Dense ANN	1.528	1.529
L2-all	0.585 ± 0.006	0.558 ± 0.008
L2-wo	0.505 ± 0.003	0.522 ± 0.002
L1-wo	0.402 ± 0.005	0.429 ± 0.005
SSSR	0.348 ± 0.002	0.371 ± 0.005
ResNet-18		
Dense ANN	2.555	2.555
L2-all	0.721 ± 0.004	0.763 ± 0.014
L2-wo	0.581 ± 0.029	0.706 ± 0.004
L1-wo	0.487 ± 0.010	0.639 ± 0.015
SSSR	0.544 ± 0.006	0.682 ± 0.008

Table 3: Estimated arithmetic energy (mJ per test sample) at $L = T = 16$.

5 Conclusion

We studied what determines robustness after ANN-to-SNN conversion when the source ANN is trained only on clean data. We showed that the stability of finite-time spike-count representations is jointly governed by perturbation amplification and the spacing between adjacent discrete states. This analysis motivates SSSR, a conversion-aware regularizer that directly reduces this sensitivity. Across VGG and ResNet models on CIFAR-10/100, SSSR improves strong-perturbation accuracy over standard L1/L2 regularization by up to 50.17 percentage points while maintaining competitive clean accuracy. Layerwise diagnostics further support the proposed mechanism, showing that perturbation magnitude alone does not explain state stability. Overall, our results show that post-conversion robustness can be shaped effectively during source-ANN training.

AI USE STATEMENT

We used ChatGPT and Codex to assist with drafting and revising theoretical arguments and proofs, providing feedback on experimental design, interpreting results, developing and debugging implementation and analysis code, translating and editing manuscript text, and preparing scientific figures and LaTeX layouts. The authors reviewed the AI-assisted material, checked the theoretical arguments and code, and verified reported numerical values against the experimental records. The authors take responsibility for the final manuscript, including its claims, results, and artifacts.

REPRODUCIBILITY STATEMENT

Training and evaluation code, together with setup instructions, is available in an anonymized repository. Section 4 describes the datasets, baselines, perturbation protocol, evaluation metrics, and reporting conventions. Appendix B provides detailed architecture and training settings, including optimization, data preprocessing, regularization coefficients, training seeds, checkpoint selection, and rate-uniform inference for the CIFAR and ImageNet experiments. Section 3 specifies the regularization objective and gradient-routing rules, with proofs and analytical assumptions provided in Appendix A. Appendix C reports detailed numerical results, additional experimental comparisons, and ablations. Appendix C.7 specifies the operation-count model used for arithmetic-energy estimation.

REFERENCES

Ernesto Araya, Massimiliano Datres, and Gitta Kutyniok. Random spiking neural networks are stable and spectrally simple. In *International Conference on Learning Representations*, 2026.

Velibor Bojkovic, Xiaofeng Wu, and Bin Gu. Temporal misalignment in ANN-SNN conversion and its mitigation via probabilistic spiking neurons. In *Forty-second International Conference on Machine Learning*, 2025.

Tong Bu, Jianhao Ding, Zhaofei Yu, and Tiejun Huang. Optimized potential initialization for low-latency spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11–20, 2022a.

Tong Bu, Wei Fang, Jianhao Ding, Penglin Dai, Zhaofei Yu, and Tiejun Huang. Optimal ANN-SNN conversion for high-accuracy and ultra-low-latency spiking neural networks. In *International Conference on Learning Representations*, 2022b.

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.

Peter U Diehl, Daniel Neil, Jonathan Binas, Matthew Cook, Shih-Chii Liu, and Michael Pfeiffer. Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International Joint Conference on Neural Networks*, pp. 1–8. IEEE, 2015.

Jianhao Ding, Zhiyu Pan, Yujia Liu, Zhaofei Yu, and Tiejun Huang. Robust stable spiking neural networks. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 11016–11029, 2024.

Zecheng Hao, Tong Bu, Jianhao Ding, Tiejun Huang, and Zhaofei Yu. Reducing ANN-SNN conversion error through residual membrane potential. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11–21, 2023a.

Zecheng Hao, Jianhao Ding, Tong Bu, Tiejun Huang, and Zhaofei Yu. Bridging the gap between ANNs and SNNs by calibrating offset spikes. In *International Conference on Learning Representations*, 2023b.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 558–567. IEEE, 2019.

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.

Yangfan Hu, Qian Zheng, Xudong Jiang, and Gang Pan. Fast-snn: Fast spiking neural network by converting quantized ann. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45 (12):14546–14562, 2023.

Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 2704–2713. IEEE, 2018.

Bum Jun Kim, Hyeyeon Choi, Hyeonah Jang, and Sang Woo Kim. Guidelines for the regularization of gammas in batch normalization for deep residual networks. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–20, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Jun Haeng Lee, Tobi Delbruck, and Michael Pfeiffer. Training deep spiking neural networks using backpropagation. *Frontiers in Neuroscience*, 10:508, 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 *International Conference on Machine Learning*, pp. 6316–6325. PMLR, 2021.

Chang Liu, Jiangrong Shen, Xuming Ran, Mingkun Xu, Qi Xu, Yi Xu, and Gang Pan. Efficient ANN-SNN conversion with error compensation learning. In *Forty-second International Conference on Machine Learning*, 2025.

Fangxin Liu, Wenbo Zhao, Yongbiao Chen, Zongwu Wang, and Li Jiang. Spikeconverter: An efficient conversion framework zipping the gap between artificial neural networks and spiking neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 1692–1701, 2022.

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

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.

Ozan Ozdenizci and Robert Legenstein. Adversarially robust spiking neural networks through conversion. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

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.

Abhronil Sengupta, Yuting Ye, Robert Wang, Chiao Liu, and Kaushik Roy. Going deeper in spiking neural networks: Vgg and residual architectures. *Frontiers in Neuroscience*, 13:95, 2019.

Twan Van Laarhoven. L2 regularization versus batch and weight normalization. *arXiv preprint arXiv:1706.05350*, 2017.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in Neuroscience*, 12:331, 2018.

Yuchen Yang, Jingcheng Liu, Chengting Yu, Chengyi Yang, Gaoang Wang, and Aili Wang. Neubridge: bridging quantized activations and spiking neurons for ANN-SNN conversion. *Neuromorphic Computing and Engineering*, 5(2):024018, 2025.

A Proofs and Analytical Qualifications

A.1 Proof of Proposition 1

Fix u and let $b_-(u)$ and $b_+(u)$ be the nearest lower and upper transition boundaries that delimit its current unclipped quantization cell. When both exist, define $d_- = u - b_-$ and $d_+ = b_+ - u$. A code change requires $\delta \le -d_-$ or $\delta \ge d_+$, so for $\delta \mid u \sim \mathcal{N}(0, \tau^2)$,

$$\Pr[\text{cross} \mid u] = Q(d_-/\tau) + Q(d_+/\tau) \le 2Q(\min(d_-, d_+)/\tau) = 2Q(d_\ell(u)/\tau). \tag{17}$$

At a clipped endpoint, only one exit direction can change the code, making the same upper bound valid. At a boundary $d_\ell(u) = 0$, the bound is one and is vacuous. The argument also shows why the sample-dependent $d_\ell(u)$ cannot be replaced by the nominal half-bin spacing to obtain a universal upper bound: within-cell locations are not assumed to be centered.

For a perturbation with conditional mean $\mathbb{E}[\delta \mid u] = 0$ and conditional variance $\mathrm{Var}(\delta \mid u) = \tau^2$, Chebyshev's inequality gives the weaker distribution-free statement $\Pr[\text{cross} \mid u] \le \min(1, \tau^2/d_\ell(u)^2)$ for $d_\ell(u) > 0$. Neither result bounds the final class decision after a sequence of nonlinear layers. The distribution-free statement is weaker than the Gaussian bound and does not imply identical robustness across noise distributions.

A.2 Derivation of the local perturbation variance

We derive Eq. (7) by explicitly relating perturbations at the input of a folded linear transformation to the resulting preactivation perturbation. The same derivation applies to a convolution after flattening the input patch and the corresponding output-channel kernel.

Consider output channel c of layer ℓ. After folding an immediately following BN layer, its time-averaged clean preactivation can be written as

$$u_{\ell,c} = \widetilde{W}_{\ell,c} \left(\frac{1}{T} \sum_{t=1}^{T} x_t \right) + \widetilde{b}_{\ell,c}, \tag{18}$$

where x_t is the layer input at timestep t and $\widetilde{b}_{\ell,c}$ is the folded bias.

Suppose the input at each timestep is perturbed as $x'_t = x_t + \xi_t$, where $\xi_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 I)$. The corresponding perturbed preactivation is

$$\begin{aligned} u'_{\ell,c} &= \widetilde{W}_{\ell,c} \left(\frac{1}{T} \sum_{t=1}^{T} (x_t + \xi_t) \right) + \widetilde{b}_{\ell,c} \\ &= u_{\ell,c} + \widetilde{W}_{\ell,c} \frac{1}{T} \sum_{t=1}^{T} \xi_t. \end{aligned} \tag{19}$$

Hence, the induced preactivation perturbation is

$$\delta_{\ell,c} = u'_{\ell,c} - u_{\ell,c} = \widetilde{W}_{\ell,c} \frac{1}{T} \sum_{t=1}^{T} \xi_t. \tag{20}$$

The folded bias cancels because it is unchanged by the additive input perturbation.

Because the perturbations are independent across timesteps,

$$\frac{1}{T} \sum_{t=1}^{T} \xi_t \sim \mathcal{N}\left(0, \frac{\sigma^2}{T} I\right). \tag{21}$$

A linear transformation of this Gaussian variable therefore gives

$$\delta_{\ell,c} \sim \mathcal{N}\left(0, \frac{\sigma^2}{T} \|\widetilde{W}_{\ell,c}\|_2^2\right), \tag{22}$$

and thus

$$\tau_{\ell,c}^2 = \mathrm{Var}(\delta_{\ell,c}) = \frac{\sigma^2}{T}\|\widetilde{W}_{\ell,c}\|_2^2, \tag{23}$$

which yields Eq. (7). This is a local calculation for independent additive perturbations entering a fixed folded transformation; it does not assert that perturbations propagated through preceding nonlinear layers remain independent, Gaussian, or isotropic.

A.3 Phase-averaged connection between sensitivity and state transitions

We derive the phase-averaged state-transition probability underlying Proposition 2. Consider first a single channel and an unbounded uniform quantizer with interval width Δ. Let the position of the noiseless preactivation within its current interval be uniformly distributed on $[0, \Delta)$ and independent of an additive perturbation δ.

For a fixed perturbation magnitude $|\delta| < \Delta$, a state transition occurs only when the initial preactivation lies within distance $|\delta|$ of the boundary in the direction of the perturbation. Because the within-interval position is uniform, the fraction of positions that cross a boundary is $|\delta|/\Delta$. If $|\delta| \geq \Delta$, every initial position changes the quantized state. Hence, after averaging over the within-interval position,

$$P_{\mathrm{phase}} = \mathbb{E}_\delta\left[\min\left(1, \frac{|\delta|}{\Delta}\right)\right]. \tag{24}$$

For $\delta = \tau Z$, where $Z \sim \mathcal{N}(0,1)$ and $\tau > 0$, define $a = \Delta/\tau$. Direct integration gives

$$P_{\mathrm{phase}} = \frac{2\tau}{\Delta}\left[\phi(0) - \phi(a)\right] + 2Q(a), \tag{25}$$

where $\phi(\cdot)$ and $Q(\cdot)$ are the standard normal density and upper tail probability, respectively.

For the bound used in the main text, Eq. (24) immediately gives

$$\begin{aligned} P_{\mathrm{phase}} &\leq \min\left\{1, \frac{\mathbb{E}|\delta|}{\Delta}\right\} \\ &= \min\left\{1, \sqrt{\frac{2}{\pi}}\frac{\tau}{\Delta}\right\}, \end{aligned} \tag{26}$$

because $\mathbb{E}|\delta| = \tau\sqrt{2/\pi}$ for a zero-mean Gaussian perturbation.

For channel c of layer ℓ, Eq. (7) gives

$$\tau_{\ell,c} = \frac{\sigma}{\sqrt{T}}\|\widetilde{W}_{\ell,c}\|_2. \tag{27}$$

Substituting this into Eq. (26) yields

$$P_{\ell,c} \leq \min\left\{1, \sqrt{\frac{2}{\pi}}\frac{\sigma}{\sqrt{T}}\frac{\|\widetilde{W}_{\ell,c}\|_2}{\Delta_\ell}\right\}. \tag{28}$$

Averaging across the O_ℓ output channels and using

$$\frac{1}{O_\ell}\sum_{c=1}^{O_\ell}\|\widetilde{W}_{\ell,c}\|_2 \leq \sqrt{\frac{1}{O_\ell}\sum_{c=1}^{O_\ell}\|\widetilde{W}_{\ell,c}\|_2^2} = \sqrt{M_{\mathrm{eff},\ell}}, \tag{29}$$

gives

$$\overline{P}_\ell \leq \min\left\{1, \sqrt{\frac{2}{\pi}}\frac{\sigma}{\sqrt{T}}\frac{\sqrt{M_{\mathrm{eff},\ell}}}{\Delta_\ell}\right\}. \tag{30}$$

Using the definition $\mathcal{S}_\ell = M_{\mathrm{eff},\ell}/\Delta_\ell^2$ therefore gives

$$\overline{P}_\ell \leq \min\left\{1, \sqrt{\frac{2}{\pi}}\frac{\sigma}{\sqrt{T}}\sqrt{\mathcal{S}_\ell}\right\}, \tag{31}$$

which proves Proposition 2.

For clipped quantizers, merging the endpoint regions cannot introduce a state transition when the corresponding unbounded quantizer would remain in the same state, so the unbounded result remains a valid upper bound.

The uniform-within-interval assumption is important. If activations concentrate near state-transition boundaries, their observed transition probability may be higher than suggested by a reference margin based only on Δ_ℓ. Moreover, source training can alter the within-interval activation distribution itself. Proposition 2 therefore establishes a phase-averaged analytical connection between $\mathcal{S}_\ell$ and local state stability, rather than a per-sample or end-to-end robustness guarantee.

A.4 Relation between SSSR and standard L2 regularization

We compare SSSR with two standard regularization scopes used in our experiments. Let $\mathcal{W}$ denote the set of convolutional and linear weights, and let Θ_{other} contain the remaining trainable parameters, including BN affine parameters and learned activation thresholds. L2-all applies a common quadratic penalty to all trainable parameters,

$$\mathcal{R}_{\text{L2-all}} = \frac{\eta}{2}\left(\sum_{W_\ell \in \mathcal{W}} \|W_\ell\|_F^2 + \sum_{\theta \in \Theta_{\text{other}}} \|\theta\|_2^2\right), \tag{32}$$

whereas weights-only L2 (L2-wo) regularizes only convolutional and linear weights,

$$\mathcal{R}_{\text{L2-wo}} = \frac{\eta}{2}\sum_{W_\ell \in \mathcal{W}} \|W_\ell\|_F^2. \tag{33}$$

In contrast, SSSR uses the matched sensitivity penalty in Eq. (14), together with ordinary L2 on unmatched weights. The following special case makes its relation to L2-wo explicit.

Proposition 3 (Reduction to weights-only L2)**.** *Suppose BN folding is identity, $\widetilde{W}_{\ell,c} = W_{\ell,c}$, and the activation spacings Δ_ℓ are fixed during optimization. Assume that $\mathcal{M}$ and $\mathcal{U}$ partition the convolutional and linear weights. Then the complete SSSR regularization term reduces to*

$$\eta_M \mathcal{R}_M + \eta_U \mathcal{R}_U = \sum_{\ell \in \mathcal{M}} \frac{\eta_M}{O_\ell \Delta_\ell^2}\|W_\ell\|_F^2 + \frac{\eta_U}{2}\sum_{\ell \in \mathcal{U}} \|W_\ell\|_F^2. \tag{34}$$

Hence SSSR is a weights-only quadratic regularizer with fixed layerwise coefficients. It is equivalent to uniform L2-wo when the coefficients are equal for all matched and unmatched weights; if $\mathcal{U}$ is nonempty, this requires $\eta_M/(O_\ell\Delta_\ell^2) = \eta_U/2$ for every matched layer.

Proof. Under identity BN folding,

$$M_{\text{eff},\ell} = \frac{1}{O_\ell}\sum_{c=1}^{O_\ell} \|W_{\ell,c}\|_2^2 = \frac{1}{O_\ell}\|W_\ell\|_F^2. \tag{35}$$

Substituting this expression into Eq. (14) and adding the unmatched-weight L2 contribution gives Eq. (34). Because Δ_ℓ is fixed, each coefficient $\eta_M/(O_\ell\Delta_\ell^2)$ is constant with respect to optimization. The resulting penalty therefore has the parameter scope of L2-wo rather than L2-all, with fixed layerwise coefficients. □

Remark 2 (When SSSR differs from L2)**.** Outside this fixed-spacing, identity-folding limit, SSSR penalizes BN-folded amplification relative to learned activation spacing, rather than raw weight magnitude alone. The matched penalty updates weights and the QCFS threshold λ, but its gradient through BN-γ is stopped; BN affine parameters remain trainable through $\mathcal{L}_{\text{CE}}$. Thus SSSR differs from L2-wo in both its penalty and gradient paths, without directly penalizing all trainable parameters as L2-all does.

This reduction identifies a relation between regularizers, not a guarantee that SSSR improves end-to-end robustness in every noise regime.

A.5 Scope of the residual-layer surrogate

Let the perturbations arriving from a residual and shortcut branch be δ_r and δ_s. The post-addition perturbation satisfies

$$\mathrm{Var}(\delta_r + \delta_s) = \mathrm{Var}(\delta_r) + \mathrm{Var}(\delta_s) + 2\,\mathrm{Cov}(\delta_r, \delta_s). \tag{36}$$

The residual-aware map associates branch-ending weights with the shared post-addition IF, but the separable penalty in Eq. (16) does not estimate cross-branch covariance, upstream Jacobians, or propagated input covariance. Even zero cross-branch covariance is therefore insufficient to identify local folded-weight norms with the exact residual-block variance. The resulting objective is a local training surrogate, not an exact residual-block risk or an end-to-end robustness certificate.

B Implementation and Reproducibility

B.1 Training algorithm

All source models are trained as clean QCFS ANNs ($T = 0$). For each minibatch:

1. Compute the cross-entropy loss.
2. For each matched Conv/Linear weight in $\mathcal{M}$, locate its associated IF activation and fold any immediately associated BN. Compute $M_{\mathrm{eff},\ell}/\Delta_\ell^2$ using the current weight and threshold. The BN scaling factor is treated as a stop-gradient quantity in this penalty.
3. Accumulate $\|W_\ell\|_F^2/2$ over unmatched Conv/Linear weights in $\mathcal{U}$; exclude biases, BN affine parameters, and IF thresholds.
4. Add the two regularization terms to cross-entropy and apply SGD. The matched penalty updates weights and thresholds, but not BN-γ. Cross-entropy still updates all trainable parameters, including BN affine parameters and thresholds; BN running statistics are non-gradient buffers.

No perturbations are sampled during source training.

Rate-uniform inference. As stated in the main-text setup, evaluation uses rate-uniform inference (Hao et al., 2023a). Each IF layer first receives its entire T-step input sequence and accumulates the input charge $z = \sum_{t=1}^{T} x_t$. Its spike count is

$$n = \mathrm{clip}(\mathrm{round}(z/\lambda),\, 0,\, T)\,. \tag{37}$$

A lookup table places these n spikes approximately uniformly across the T steps, with each spike carrying amplitude λ and a fixed zero phase offset. The resulting sequence is then passed to the next layer. This is a layerwise, sequence-level evaluation protocol, not causal online IF simulation that emits spikes before all inputs have arrived. At every timestep, independent Gaussian noise is added to the feature map after the input IF. A single noise tensor reused across timesteps would define a different, temporally correlated protocol. Source-ANN controls use $T = 0$ and noise at the corresponding activation site.

B.2 Architectures and optimization

Primary CIFAR experiments. QCFS VGG-16 with BN and CIFAR ResNet-18 are trained on all 50,000 training images at $L = 16$ for 300 epochs. We use batch size 128, SGD momentum 0.9, initial learning rate 0.1, and cosine annealing over the training run. Augmentation includes random crops with four-pixel padding, horizontal flips, CIFAR AutoAugment, and Cutout; test images receive only dataset normalization. Seeds 40–44 vary initialization and data order. L2-all and L2-wo use coefficient 5×10^{-4}, and L1-wo uses 10^{-5}. For SSSR, $\eta_M = 10^{-4}$ and $\eta_U = 5\times10^{-4}$. The matched penalty has no additional optimizer weight decay. These coefficients are fixed across CIFAR-10/100 and the tested CIFAR architectures, without dataset-specific retuning.

The VGG-16 map contains 15 matched weights and one unmatched final classifier. The residual-aware ResNet-18 map contains 20 matched weights, including the stem, residual convolutions, and projection shortcuts; the final fully connected weight is unmatched. The reported CIFAR checkpoints

use the epoch with the highest clean accuracy on the official test split; noisy accuracy is not used for epoch selection.

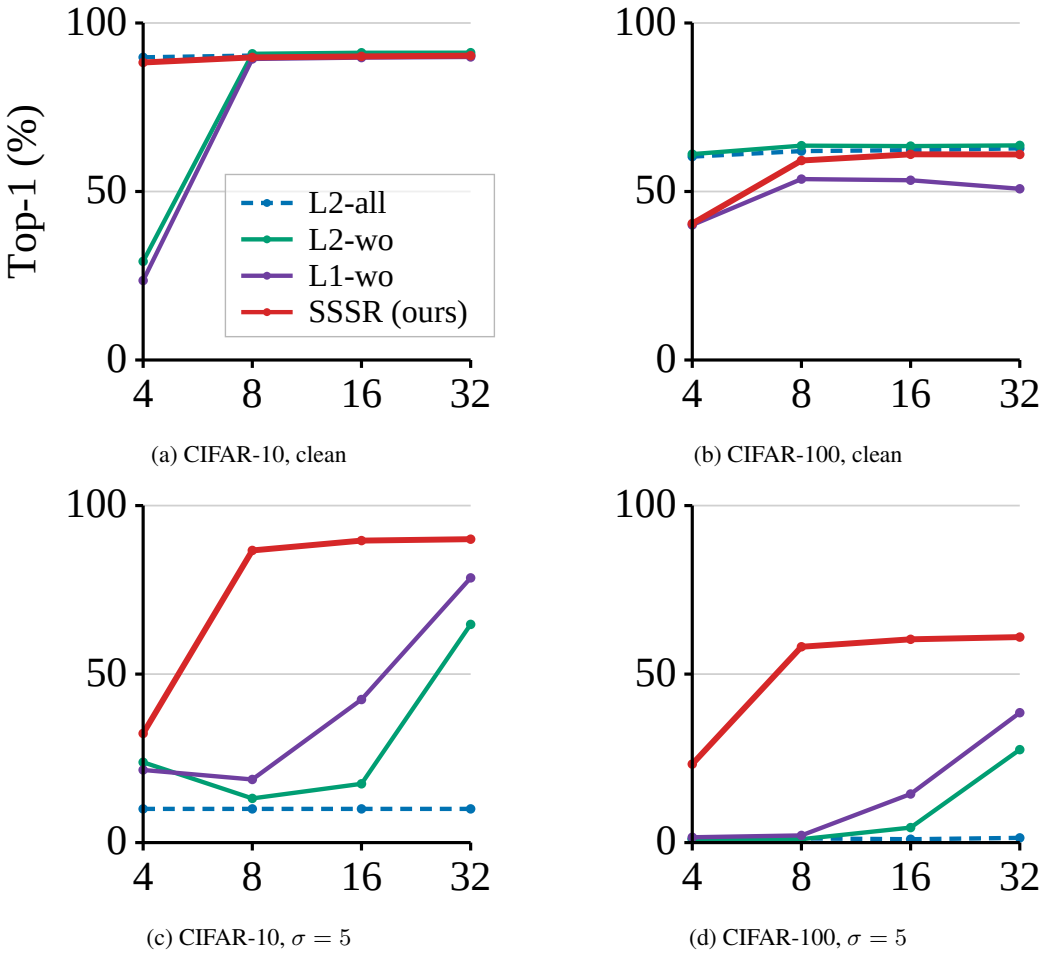
Network-depth comparisons. The additional VGG-13/19 and CIFAR ResNet-34 source models use the same 300-epoch CIFAR optimization and augmentation settings, with $L = T = 16$ and the same method-specific coefficients. The depth tables report seed 42. VGG models use the sequential Conv/Linear–IF map, whereas ResNet-34 uses the residual-aware map. Each model's final classifier receives unmatched-weight L2 under SSSR. The VGG-13/19 tables also include seed-42 L2-all, L2-wo, and L1-wo results; corresponding CIFAR ResNet-34 baseline results are not available.

Quantization-level and simulation-length comparisons. For VGG-16 on CIFAR-10/100, separate source models are trained at $L \in \{4, 8, 16, 32\}$ using the same CIFAR training recipe. Only the quantization level changes; regularization coefficients are not retuned across L. Each source checkpoint is reused at $T \in \{4, 8, 16, 32\}$ without retraining when T changes. The matched-$L = T$ comparisons and CIFAR-10 grid use seed 42; the CIFAR-100 grid reports mean accuracy over the available checkpoints in each cell. The $T = 0$ controls evaluate the quantized source representation, not a separately trained continuous-activation ANN.

ImageNet experiments. ImageNet ResNet-18 source models are trained at $L = 16$ for 90 epochs, using batch size 256, SGD momentum 0.9, initial learning rate 0.1, and cosine annealing. Training uses random resized crops to 224×224, horizontal flips, and ImageNet normalization. Validation resizes the shorter side to 256 pixels and applies a 224×224 center crop with the same normalization. L2-all and L2-wo use coefficient 10^{-4}, L1-wo uses 10^{-5}, and SSSR uses $\eta_M = \eta_U = 10^{-4}$ with the residual-aware map. All ImageNet results use seed 42 and $T = 16$ on the official validation set. The source checkpoint is selected by the highest clean ANN top-1 accuracy on this validation set during training, rather than by the final epoch; no noisy validation accuracy is used for checkpoint selection. The noise sweep covers $\sigma = 0$ to 5 in increments of 0.25; the main figure displays the range up to 2. These are single-checkpoint results, not five-seed estimates.

C Additional Empirical Results

C.1 Matched quantization levels and inference length



(a) CIFAR-10, clean

(b) CIFAR-100, clean

(c) CIFAR-10, $\sigma = 5$

(d) CIFAR-100, $\sigma = 5$

Figure 9: VGG-16 top-1 test accuracy for independently trained source models at $L = T \in \{4, 8, 16, 32\}$ on CIFAR-10/100 (seed 42). Gaussian noise is added after the input IF at every timestep. The same method-specific regularization coefficients are used at every L.

Figure 9 retains the matched-$L = T$ clean and strong-noise comparison. From $L = T = 8$ onward, SSSR reaches at least 86.69% accuracy on CIFAR-10 and 58.11% on CIFAR-100 at $\sigma = 5$, exceeding all three standard baselines at the corresponding operating points. At $L = T = 4$, it has the highest $\sigma = 5$ accuracy on both datasets (32.39% and 23.29%), but its CIFAR-100 clean accuracy (40.48%) is 20.61 points below L2-wo.

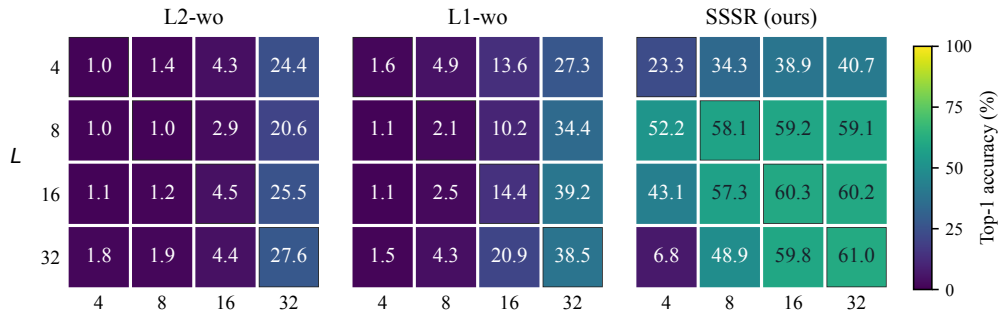


Figure 10: VGG-16 CIFAR-100 top-1 test accuracy (%) under post-input-IF Gaussian noise at $\sigma = 5$. Rows vary the source quantization level L; columns vary the converted SNN simulation length T. The color scale matches Figure 7, and outlined cells have $L = T$. L2-all is omitted because its accuracy remains near the 1% chance level throughout the grid.

Figure 10 extends the $L \times T$ comparison to CIFAR-100. At $L = 16, T = 32$, SSSR reaches 60.2% accuracy, compared with 39.2% for L1-wo and 25.5% for L2-wo. With only four time steps at $L = 32$, however, SSSR drops to 6.8%, showing that the advantage also depends on sufficient inference time on this dataset.

ResNet-18 at shorter simulation lengths. Table 4 complements the VGG-16 grids with fixed-$L = 16$ ResNet-18 evaluations at $T = 4, 8, 16$. At $\sigma = 5$, SSSR reaches 78.83% and 58.60% on CIFAR-10/100 at $T = 4$, increasing to 91.22% and 70.36% at $T = 8$. It outperforms the standard regularizers at both shorter simulation lengths, although clean accuracy is lower than at $T = 16$.

Table 4: ResNet-18 test accuracy (%) at fixed $L = 16$ and $T = 4, 8, 16$ (five-seed mean $\pm$ sample standard deviation). Each method reuses the same checkpoints across T; Gaussian noise is added after the input IF.

		$T = 4$		$T = 8$		$T = 16$	
Dataset	Method	Clean	$\sigma = 5$	Clean	$\sigma = 5$	Clean	$\sigma = 5$
CIFAR-10	L2-all	95.72 ± 0.17	10.06 ± 0.36	96.30 ± 0.14	10.24 ± 0.31	96.53 ± 0.16	10.24 ± 0.38
	L2-wo	95.34 ± 0.22	11.62 ± 1.29	96.38 ± 0.15	18.98 ± 2.86	96.68 ± 0.07	53.88 ± 6.84
	L1-wo	93.63 ± 0.17	23.32 ± 5.46	95.15 ± 0.11	50.66 ± 6.63	95.53 ± 0.08	80.10 ± 3.20
	SSSR	92.81 ± 0.27	78.83 ± 5.15	95.25 ± 0.15	91.22 ± 1.75	95.67 ± 0.06	94.20 ± 0.73
CIFAR-100	L2-all	77.59 ± 0.31	0.99 ± 0.03	79.37 ± 0.22	0.98 ± 0.04	79.99 ± 0.22	0.98 ± 0.04
	L2-wo	76.90 ± 0.49	2.55 ± 0.63	79.33 ± 0.20	9.45 ± 2.15	79.96 ± 0.32	31.11 ± 4.28
	L1-wo	73.57 ± 0.49	3.56 ± 1.23	77.35 ± 0.37	20.83 ± 4.76	78.37 ± 0.27	50.69 ± 3.19
	SSSR	71.56 ± 0.46	58.60 ± 1.56	75.72 ± 0.39	70.36 ± 0.82	76.78 ± 0.32	74.62 ± 0.39

C.2 Source ANN and converted SNN noise curves

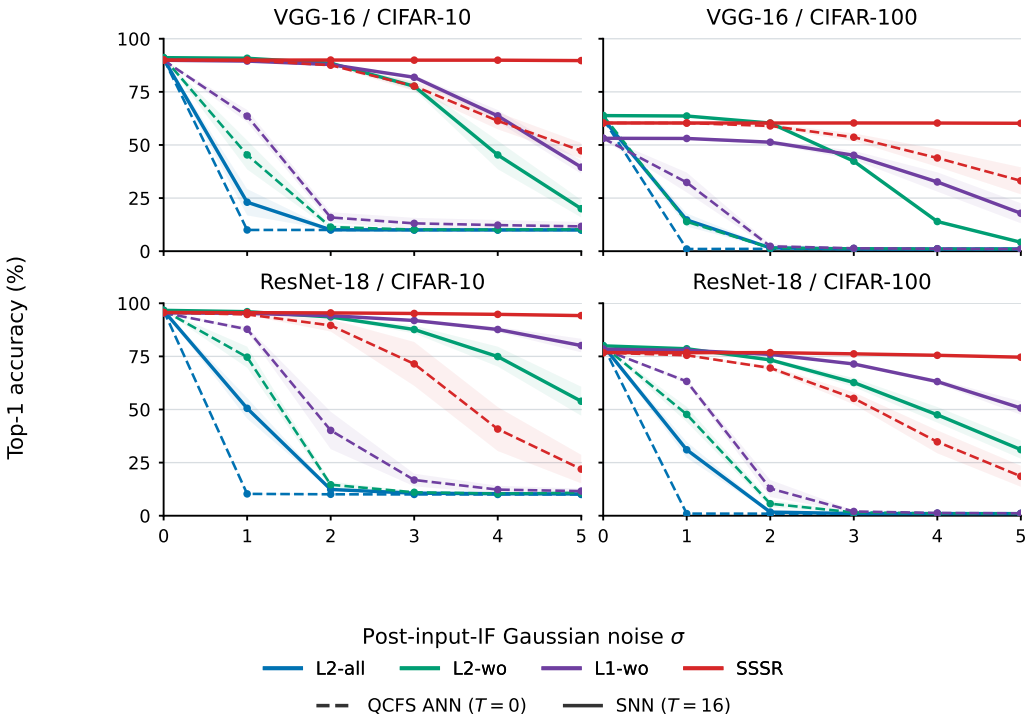


Figure 11: Top-1 test accuracy under post-input-IF Gaussian noise for the same $L = 16$ source checkpoints evaluated as QCFS ANNs ($T = 0$, dashed) and converted SNNs ($T = 16$, solid). Curves show five-seed means at the common evaluated severities $\sigma \in \{0, 1, 2, 3, 4, 5\}$; shading shows one sample standard deviation. Lines only connect measured points.

Figure 11 includes L2-all, L2-wo, L1-wo, and SSSR on both CIFAR datasets and architectures. At $\sigma = 5$, the SSSR source ANN outperforms every standard regularizer in all four settings, so its robustness advantage is not created solely by SNN conversion. Converted SNN accuracy is higher at $T = 16$, but this comparison also changes the forward representation and how noise is encountered over time; it is not a causal estimate of temporal integration alone.

C.3 TRANSFER ACROSS NETWORK DEPTHS

Tables 5 and 6 give the complete single-seed depth-screen results for SSSR and seed-42 comparisons with L2-all, L2-wo, and L1-wo on VGG-13/19 at integer noise levels. Gaussian noise is injected after the input IF at every timestep with $L = T = 16$. The same method-specific regularization coefficients are used across architectures; these single-seed results are separate from the five-seed main comparison. Corresponding CIFAR baseline results for ResNet-34 are not available.

Table 5: CIFAR-10 test top-1 accuracy (%) at $L = T = 16$, seed 42. Top: the full SSSR depth scan. Bottom: VGG-13/19 regularizer comparison at integer IF Gaussian noise levels.

σ	VGG-13	VGG-16	VGG-19	ResNet-18	ResNet-34
0	90.22	90.09	89.83	95.64	95.49
0.25	90.16	90.09	89.81	95.67	95.66
0.5	90.15	90.03	89.70	95.68	95.70
0.75	90.25	90.14	89.79	95.59	95.58
1	90.14	90.10	89.69	95.49	95.67
1.25	90.23	90.05	89.70	95.69	95.56
1.5	90.17	90.01	89.69	95.62	95.76
1.75	90.18	90.13	89.67	95.56	95.54
2	90.27	90.01	89.70	95.53	95.65
2.25	90.27	89.98	89.66	95.49	95.61
2.5	90.20	90.01	89.70	95.46	95.45
2.75	90.30	89.93	89.58	95.33	95.42
3	90.19	89.93	89.58	95.26	95.49
3.25	90.25	90.00	89.64	95.16	95.30
3.5	90.18	90.07	89.55	95.13	95.17
3.75	90.12	89.91	89.53	95.18	95.09
4	90.13	89.87	89.58	94.98	95.08
4.25	90.04	89.72	89.53	94.89	94.97
4.5	89.92	89.70	89.39	94.85	94.77
4.75	89.87	89.56	89.47	94.63	94.70
5	89.96	89.61	89.41	94.40	94.34

Model	Method	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$
VGG-13	L2-all	90.94	19.69	10.19	10.00	10.00	10.00
	L2-wo	91.42	91.10	89.19	79.30	51.60	23.84
	L1-wo	90.34	90.21	88.81	82.03	63.28	42.07
	SSSR	90.22	90.14	90.27	90.19	90.13	89.96
VGG-19	L2-all	90.33	23.36	10.00	10.00	10.00	10.00
	L2-wo	90.80	90.48	88.64	80.29	50.34	17.55
	L1-wo	87.95	87.60	86.56	83.06	72.78	54.52
	SSSR	89.83	89.69	89.70	89.58	89.58	89.41

On CIFAR-10, SSSR retains 89.41–89.96% accuracy across the three VGG depths at $\sigma = 5$; both ResNet variants remain above 94%. The VGG-13/19 comparisons show that the robustness gain is not restricted to VGG-16: the strongest conventional baseline reaches only 42.07% and 54.52%, respectively. These are single-seed comparisons, not estimates of variation across training runs.

Table 6: CIFAR-100 test top-1 accuracy (%) at $L = T = 16$, seed 42. Top: the full SSSR depth scan. Bottom: VGG-13/19 regularizer comparison at integer IF Gaussian noise levels.

σ	VGG-13	VGG-16	VGG-19	ResNet-18	ResNet-34
0	62.31	61.03	57.32	76.75	77.32
0.25	62.24	61.13	57.26	77.01	77.29
0.5	62.38	60.99	57.33	77.02	77.07
0.75	62.40	61.08	57.30	76.87	77.23
1	62.37	61.06	57.46	76.67	77.28
1.25	62.53	61.14	57.60	76.42	77.35
1.5	62.43	61.05	57.46	76.67	77.17
1.75	62.54	61.04	57.35	76.59	77.24
2	62.54	61.03	57.44	76.77	76.78
2.25	62.65	61.08	57.50	76.61	77.07
2.5	62.71	61.07	57.43	76.22	76.98
2.75	62.77	60.86	57.46	76.05	76.84
3	62.80	60.83	57.45	76.00	76.70
3.25	62.71	60.96	57.63	75.99	76.47
3.5	62.92	60.85	57.67	75.81	76.26
3.75	62.88	61.05	57.56	75.73	76.18
4	62.76	60.81	57.59	75.37	76.06
4.25	62.90	60.71	57.61	75.04	75.88
4.5	62.69	60.50	57.55	74.97	76.05
4.75	62.86	60.64	57.42	74.49	75.35
5	62.70	60.32	57.34	74.48	75.19

Model	Method	$\sigma = 0$	$\sigma = 1$	$\sigma = 2$	$\sigma = 3$	$\sigma = 4$	$\sigma = 5$
VGG-13	L2-all	64.58	7.40	1.02	1.00	1.00	1.00
	L2-wo	65.89	65.62	61.78	44.37	16.14	4.93
	L1-wo	51.96	51.89	51.32	47.93	40.79	29.19
	SSSR	62.31	62.37	62.54	62.80	62.76	62.70
VGG-19	L2-all	58.07	14.86	1.06	1.20	1.19	1.10
	L2-wo	60.18	60.46	57.43	41.92	13.88	4.05
	L1-wo	44.51	44.33	43.18	34.55	21.46	10.27
	SSSR	57.32	57.46	57.44	57.45	57.59	57.34

On CIFAR-100, SSSR reaches 62.70% and 57.34% at $\sigma = 5$ on VGG-13 and VGG-19, respectively, compared with 29.19% and 10.27% for L1-wo. The ResNet-18/34 depth scan retains 74.48% and 75.19%, respectively.

C.4 Regularization-scope motivation

Figure 12 reports the single-seed four-way scope experiment used to motivate the study. It is not the final method comparison and does not establish that penalizing BN-γ is generally harmful. It shows that, in this VGG-16 configuration, the two scopes containing BN-γ collapse at moderate noise while weights-only and weights plus BN-β remain stable.

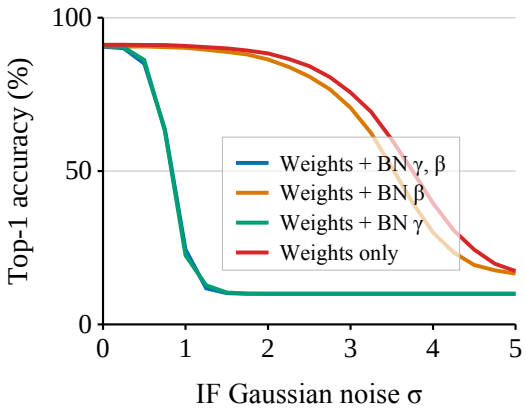


Figure 12: CIFAR-10 VGG-16 regularization-scope control, one training seed. Gaussian noise is independent across feature-map elements and timesteps.

C.5 Layerwise sensitivity and state changes

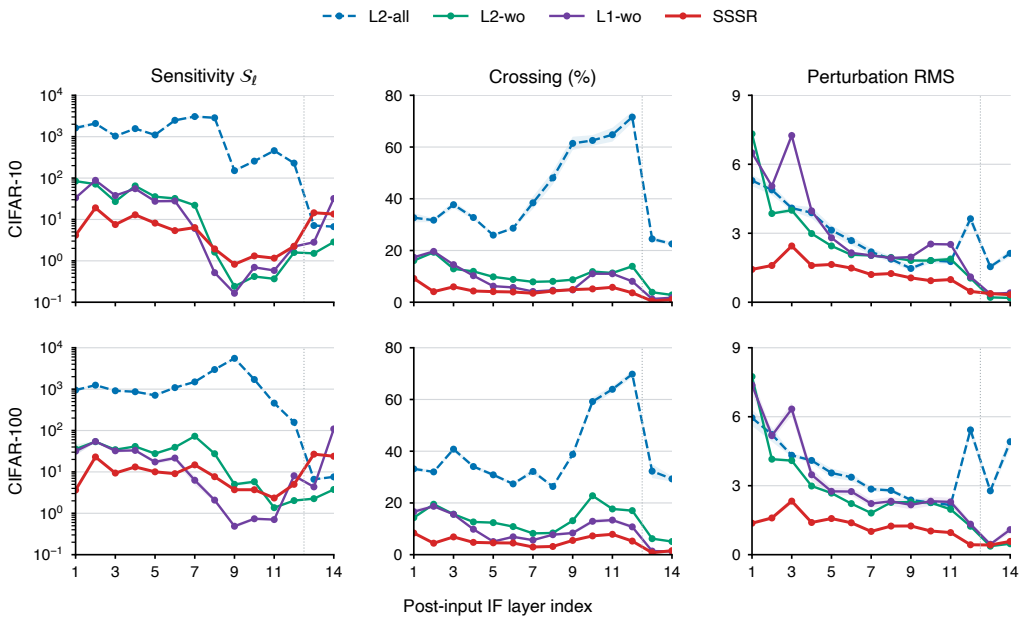


Figure 13: VGG-16 layerwise diagnostics at $L = T = 16$ and $\sigma = 1$ on CIFAR-10/100. Curves show means over five training seeds; bands show one sample standard deviation. Each checkpoint uses the same 1,280 test images and noise draws. Indices 1–12 are post-input IF layers in the convolutional stack; indices 13–14 are classifier IF layers. Sensitivity uses a logarithmic scale.

Figure 13 gives the per-layer measurements behind Table 1. The differences in crossing rates are largest in the later convolutional layers. The sensitivity ordering is not uniform: some later layers have lower $\mathcal{S}_\ell$ under L1-wo or L2-wo than under SSSR, so the table's layer averages should not be read as a pointwise claim.

C.6 Ablation details

Component and gradient-path comparison. Figure 14 visualizes the VGG-16 ablation in Table 2, using the same seven configurations, seed 42, and $L = T = 16$. No five-seed uncertainty estimate is attached to these single-seed results.

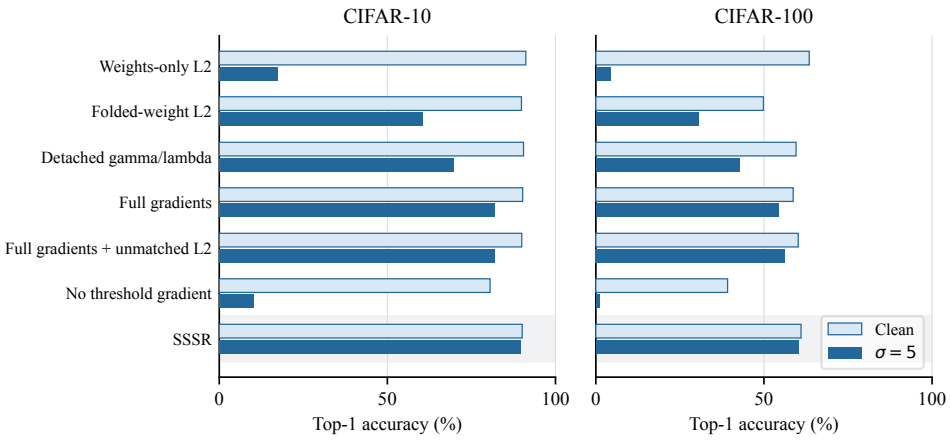


Figure 14: Clean and strong-noise accuracy for the seven VGG-16 ablation configurations on CIFAR-10/100 ($L = T = 16$, seed 42). The shaded final row is SSSR. Bars show measured accuracy, not a noise sweep.

Threshold magnitude alone is insufficient. Because SSSR learns larger thresholds, one possible explanation is that its robustness improvement comes solely from threshold inflation. To test this hypothesis, we freeze an L2-wo checkpoint and uniformly scale all IF thresholds. Matching their mean threshold to that of SSSR raises $\sigma = 5$ accuracy only to 32.58% on CIFAR-10 and 15.75% on CIFAR-100, far below the corresponding SSSR results of 89.61% and 60.32% (Table 7). A training-time $-\operatorname{mean}_\ell \log \lambda_\ell$ penalty with coefficient 0.1 recovers substantially more robustness, but requires approximately threefold larger mean thresholds and still produces higher firing than SSSR. The gain therefore cannot be explained by uniform post-hoc threshold inflation alone; jointly adapting perturbation amplification and activation spacing is important. This control tests uniform scaling and does not exclude every possible layerwise threshold allocation.

Table 7: Threshold controls on VGG-16 at $L = T = 16$ (seed 42). "Matched" uniformly scales the frozen L2-wo thresholds to the mean threshold of SSSR; λ-grow is trained with L2-wo plus $-\operatorname{mean}_\ell \log \lambda_\ell$ with coefficient 0.1. Clean and $\sigma = 5$ columns report top-1 test accuracy (%); firing is the clean-inference activity fraction.

Dataset	Configuration	Clean	$\sigma = 5$	Mean λ	Firing
CIFAR-10	L2-wo	91.17	17.44	3.77	0.101
	Post-hoc matched λ	90.29	32.58	6.51	0.075
	L2-wo + λ-grow	90.37	88.70	19.31	0.065
	SSSR	90.09	**89.61**	6.51	**0.054**
CIFAR-100	L2-wo	63.47	4.45	3.14	0.105
	Post-hoc matched λ	59.87	15.75	5.91	0.073
	L2-wo + λ-grow	62.83	54.66	16.81	0.072
	SSSR	61.03	**60.32**	5.91	**0.061**

Classifier scale is not a robustness mechanism by itself. For clean and noisy logits $f(x)$ and $\widetilde{f}(x)$, multiplying both by the same $a > 0$ leaves their predicted classes unchanged. The true-class margin $m = f_y - \max_{k \neq y} f_k$ and the standard deviation s_m of its perturbation both scale by a, so m/s_m is invariant. Consequently, a larger classifier norm or absolute margin alone does not explain better robust accuracy. Unmatched L2 changes optimization and potentially classifier directions and features, rather than providing a monotone benefit through head-scale reduction.

C.7 ARITHMETIC-ENERGY ESTIMATION DETAILS

The clean-inference energy comparison is reported in Table 3. For a dense ANN and event-driven SNN, we use

$$E_{\mathrm{ANN}} = N_{\mathrm{MAC}} E_{\mathrm{MAC}}, \qquad E_{\mathrm{SNN}} = N_{\mathrm{MAC}}^{(1)} E_{\mathrm{MAC}} + N_{\mathrm{AC}}^{(>1)} E_{\mathrm{AC}}, \tag{38}$$

with 32-bit 45-nm reference costs $E_{\mathrm{MAC}} = 4.6$ pJ and $E_{\mathrm{AC}} = 0.9$ pJ (Horowitz, 2014). The first layer receives dense analog inputs over T steps, so $N_{\mathrm{MAC}}^{(1)}$ includes all T steps; later accumulation counts are weighted by observed presynaptic events. The estimates are operation-based proxies, not measured hardware energy, and exclude membrane updates, memory access, data movement, control, and communication. They do not establish an energy advantage at matched accuracy and robustness.

Tables 8 and 9 extend the main energy comparison to $T = 4, 8, 16$, including average robustness $\overline{A}_{0:5}$, clean-inference firing density, and event counts. For SSSR on ResNet-18/CIFAR-10, estimated energy increases from 0.137 to 0.273 and 0.544 mJ as T increases. The tables expose the trade-off between average robustness and clean-inference arithmetic cost. Firing density is a per-timestep activity fraction, whereas event counts accumulate over the full simulation length. Both tables use five-seed means $\pm$ sample standard deviations at $L = 16$. Average robustness is integrated on the common measured grid $\sigma \in \{0, 1, 2, 3, 5\}$; the later $\sigma = 4$ measurements are not included in these summaries. Energy, firing, and event counts refer to clean inference.

Table 8: VGG-16 robustness–energy trade-off at $L = 16$. Columns report average robustness (%), estimated arithmetic energy (mJ per sample), IF firing density (%), and synaptic event counts (10^6 per sample).

Dataset	Method	T	$\overline{A}_{0:5}$ (%)	Energy (mJ)	Firing (%)	Events (10^6)
CIFAR-10	L2-all	4	18.05 ± 0.04	0.142 ± 0.001	10.46 ± 0.30	128.5 ± 1.6
CIFAR-10	L2-all	8	18.27 ± 0.21	0.288 ± 0.003	10.47 ± 0.30	260.9 ± 3.3
CIFAR-10	L2-all	16	20.69 ± 1.24	0.585 ± 0.006	10.47 ± 0.30	532.3 ± 6.7
CIFAR-10	L2-wo	4	43.01 ± 2.34	0.123 ± 0.001	10.21 ± 0.12	107.0 ± 0.9
CIFAR-10	L2-wo	8	56.79 ± 2.73	0.248 ± 0.002	10.24 ± 0.11	217.1 ± 1.8
CIFAR-10	L2-wo	16	72.27 ± 1.37	0.505 ± 0.003	10.24 ± 0.11	443.5 ± 3.7
CIFAR-10	L1-wo	4	49.25 ± 1.13	0.097 ± 0.001	7.66 ± 0.13	78.9 ± 1.2
CIFAR-10	L1-wo	8	64.27 ± 1.18	0.197 ± 0.002	7.66 ± 0.14	160.0 ± 2.4
CIFAR-10	L1-wo	16	76.93 ± 0.98	0.402 ± 0.005	7.68 ± 0.14	329.0 ± 5.2
CIFAR-10	SSSR	4	80.80 ± 1.19	0.084 ± 0.000	5.55 ± 0.07	63.6 ± 0.5
CIFAR-10	SSSR	8	89.48 ± 0.17	0.171 ± 0.001	5.45 ± 0.06	131.1 ± 0.9
CIFAR-10	SSSR	16	89.91 ± 0.09	0.348 ± 0.002	5.36 ± 0.06	269.4 ± 2.0
CIFAR-100	L2-all	4	7.25 ± 0.04	0.135 ± 0.002	10.58 ± 0.31	121.1 ± 2.2
CIFAR-100	L2-all	8	7.54 ± 0.14	0.274 ± 0.004	10.59 ± 0.31	245.9 ± 4.4
CIFAR-100	L2-all	16	10.01 ± 0.66	0.558 ± 0.008	10.59 ± 0.31	502.1 ± 8.7
CIFAR-100	L2-wo	4	22.90 ± 0.37	0.127 ± 0.000	10.49 ± 0.08	111.9 ± 0.5
CIFAR-100	L2-wo	8	32.01 ± 0.41	0.257 ± 0.001	10.50 ± 0.08	226.7 ± 1.1
CIFAR-100	L2-wo	16	44.72 ± 0.37	0.522 ± 0.002	10.50 ± 0.08	462.9 ± 2.3
CIFAR-100	L1-wo	4	24.84 ± 1.98	0.104 ± 0.001	8.42 ± 0.21	86.5 ± 1.3
CIFAR-100	L1-wo	8	34.64 ± 2.17	0.211 ± 0.003	8.42 ± 0.22	175.5 ± 2.8
CIFAR-100	L1-wo	16	43.31 ± 1.86	0.429 ± 0.005	8.43 ± 0.22	359.6 ± 5.8
CIFAR-100	SSSR	4	53.00 ± 1.52	0.087 ± 0.001	6.08 ± 0.27	67.8 ± 1.6
CIFAR-100	SSSR	8	59.90 ± 0.70	0.181 ± 0.003	5.99 ± 0.20	142.5 ± 2.9
CIFAR-100	SSSR	16	60.32 ± 0.49	0.371 ± 0.005	5.88 ± 0.18	295.1 ± 5.3

For VGG-16, SSSR has the lowest estimated energy and the highest average robustness at every reported T on both datasets. Its firing density changes little with T, while total event counts and energy increase. Thus, lower activity per timestep does not make longer simulations free: the additional timesteps still incur arithmetic cost.

Table 9: ResNet-18 robustness–energy trade-off at $L = 16$. Metrics and reporting conventions match Table 8.

Dataset	Method	T	$\overline{A}_{0:5}$ (%)	Energy (mJ)	Firing (%)	Events (10^6)
CIFAR-10	L2-all	4	19.09 ± 0.26	0.183 ± 0.001	10.25 ± 0.09	173.5 ± 1.0
CIFAR-10	L2-all	8	20.55 ± 0.27	0.362 ± 0.002	10.18 ± 0.09	343.1 ± 2.1
CIFAR-10	L2-all	16	27.47 ± 0.83	0.721 ± 0.004	10.16 ± 0.08	683.6 ± 4.2
CIFAR-10	L2-wo	4	53.03 ± 3.35	0.147 ± 0.008	8.61 ± 0.71	134.4 ± 8.3
CIFAR-10	L2-wo	8	71.20 ± 1.91	0.292 ± 0.015	8.52 ± 0.71	265.2 ± 16.4
CIFAR-10	L2-wo	16	84.67 ± 1.89	0.581 ± 0.029	8.50 ± 0.70	528.4 ± 32.5
CIFAR-10	L1-wo	4	67.60 ± 3.01	0.124 ± 0.002	7.25 ± 0.26	108.1 ± 2.4
CIFAR-10	L1-wo	8	82.48 ± 1.77	0.244 ± 0.005	7.18 ± 0.27	212.8 ± 5.6
CIFAR-10	L1-wo	16	91.03 ± 0.81	0.487 ± 0.010	7.16 ± 0.28	423.6 ± 11.1
CIFAR-10	SSSR	4	88.89 ± 1.39	0.137 ± 0.001	7.39 ± 0.20	123.3 ± 1.4
CIFAR-10	SSSR	8	94.07 ± 0.58	0.273 ± 0.003	7.35 ± 0.22	244.4 ± 3.2
CIFAR-10	SSSR	16	95.18 ± 0.30	0.544 ± 0.006	7.33 ± 0.22	487.2 ± 6.9
CIFAR-100	L2-all	4	8.78 ± 0.13	0.193 ± 0.004	11.69 ± 0.32	184.8 ± 4.2
CIFAR-100	L2-all	8	10.17 ± 0.37	0.383 ± 0.008	11.63 ± 0.32	366.6 ± 8.4
CIFAR-100	L2-all	16	15.05 ± 0.77	0.763 ± 0.014	11.62 ± 0.32	730.6 ± 16.1
CIFAR-100	L2-wo	4	35.67 ± 2.36	0.179 ± 0.001	11.04 ± 0.07	169.0 ± 1.3
CIFAR-100	L2-wo	8	50.57 ± 2.13	0.354 ± 0.002	10.98 ± 0.08	334.6 ± 2.4
CIFAR-100	L2-wo	16	63.42 ± 1.56	0.706 ± 0.004	10.96 ± 0.08	666.9 ± 4.9
CIFAR-100	L1-wo	4	42.22 ± 1.71	0.163 ± 0.004	9.59 ± 0.35	151.3 ± 4.5
CIFAR-100	L1-wo	8	59.38 ± 1.44	0.321 ± 0.007	9.48 ± 0.32	297.8 ± 8.1
CIFAR-100	L1-wo	16	70.16 ± 0.63	0.639 ± 0.015	9.44 ± 0.32	592.3 ± 17.1
CIFAR-100	SSSR	4	67.49 ± 0.50	0.173 ± 0.003	9.86 ± 0.26	163.0 ± 3.1
CIFAR-100	SSSR	8	74.08 ± 0.49	0.342 ± 0.004	9.76 ± 0.21	321.6 ± 4.8
CIFAR-100	SSSR	16	76.17 ± 0.40	0.682 ± 0.008	9.72 ± 0.21	640.2 ± 9.2

For ResNet-18, SSSR also has the highest average robustness at each T, but L1-wo uses less estimated energy. At $T = 4$, SSSR already reaches 88.89% and 67.49% average robustness on CIFAR-10/100, with estimated costs of 0.137 and 0.173 mJ per sample. The ranking therefore depends on whether the comparison prioritizes robustness or arithmetic cost; SSSR is not the lowest-energy regularizer for both architectures.

C.8 DETAILED STANDARD-REGULARIZER COMPARISON

Table 10 reports the endpoint accuracies and normalized area under the noise–accuracy curve corresponding to Figure 5. Clean accuracy is retained to show the robustness–accuracy trade-off. For VGG-16, L1-wo is the strongest standard baseline at $\sigma = 5$, but its CIFAR-100 clean accuracy is lower than that of L2-wo. SSSR leads under strong noise in all four settings without having the highest clean accuracy, so the two rankings should not be conflated.

Tables 8 and 9 use trapezoidal integration on the common noise grid $\sigma \in \{0, 1, 2, 3, 5\}$, whereas the SSSR entries in Table 10 use the full available, denser noise grid. These estimates approximate the same normalized integral; small differences between corresponding $T = 16$ entries reflect integration-grid resolution.

Table 10: Detailed statistics for Figure 5 at $L = T = 16$ (five-seed mean $\pm$ sample standard deviation). Accuracy and normalized area $\overline{A}_{0:5}$ are reported in percent. All methods are trained on clean examples and evaluated with independent post-input-IF Gaussian noise at each timestep. Bold and underlining mark the largest and second-largest $\sigma = 5$ means within each architecture–dataset setting, not statistical significance.

Architecture	Dataset	Method	Clean	$\sigma = 1$	$\sigma = 3$	$\sigma = 5$	$\overline{A}_{0:5}$
VGG-16	CIFAR-10	L2-all	90.59 ± 0.13	23.06 ± 6.22	10.00 ± 0.00	10.00 ± 0.00	20.69 ± 1.24
		L2-wo	91.10 ± 0.07	90.82 ± 0.07	77.64 ± 1.82	20.00 ± 4.13	72.27 ± 1.37
		L1-wo	89.87 ± 0.20	89.49 ± 0.14	81.86 ± 1.00	$\underline{39.55 \pm 3.11}$	76.93 ± 0.98
		SSSR (ours)	89.98 ± 0.12	89.96 ± 0.14	89.93 ± 0.13	$\mathbf{89.72 \pm 0.09}$	89.93 ± 0.09
VGG-16	CIFAR-100	L2-all	62.27 ± 0.37	14.71 ± 3.09	1.06 ± 0.04	1.05 ± 0.07	10.01 ± 0.66
		L2-wo	63.83 ± 0.26	63.64 ± 0.23	42.36 ± 0.96	4.20 ± 0.32	44.72 ± 0.37
		L1-wo	53.13 ± 0.73	53.07 ± 0.68	45.15 ± 2.11	$\underline{17.86 \pm 4.72}$	43.31 ± 1.86
		SSSR (ours)	60.35 ± 0.53	60.35 ± 0.60	60.35 ± 0.49	$\mathbf{60.21 \pm 0.50}$	60.33 ± 0.52
ResNet-18	CIFAR-10	L2-all	96.53 ± 0.16	50.56 ± 4.88	10.62 ± 0.53	10.24 ± 0.38	27.47 ± 0.83
		L2-wo	96.68 ± 0.07	96.05 ± 0.10	87.66 ± 1.85	53.88 ± 6.84	84.67 ± 1.89
		L1-wo	95.53 ± 0.08	95.24 ± 0.05	91.90 ± 0.58	$\underline{80.10 \pm 3.20}$	91.03 ± 0.81
		SSSR (ours)	95.67 ± 0.06	95.59 ± 0.12	95.20 ± 0.33	$\mathbf{94.20 \pm 0.73}$	95.23 ± 0.29
ResNet-18	CIFAR-100	L2-all	79.99 ± 0.22	31.04 ± 3.80	1.02 ± 0.05	0.98 ± 0.04	15.05 ± 0.77
		L2-wo	79.96 ± 0.32	78.60 ± 0.27	62.70 ± 2.12	31.11 ± 4.28	63.42 ± 1.56
		L1-wo	78.37 ± 0.27	77.86 ± 0.14	71.42 ± 0.38	$\underline{50.69 \pm 3.19}$	70.16 ± 0.63
		SSSR (ours)	76.78 ± 0.32	76.74 ± 0.43	76.20 ± 0.44	$\mathbf{74.62 \pm 0.39}$	76.17 ± 0.39

ImageNet statistics. Table 11 supplements Figure 6 with validation accuracy at selected noise severities, including $\sigma = 3, 5$ outside the plotted range. Each method uses one seed-42 checkpoint; no across-seed uncertainty estimate is available. At $\sigma = 1$, SSSR retains 40.56% accuracy versus 21.11% for L1-wo. By $\sigma = 5$, all methods are near the 0.1% chance level; the transfer result is therefore limited to the evaluated moderate-noise range.

Table 11: ImageNet ResNet-18 validation top-1 accuracy (%) at $L = T = 16$ (seed 42). All methods use the same post-input-IF Gaussian noise protocol. These are single-checkpoint results, not five-seed means. Bold marks the largest noisy accuracy in each column.

Method	Clean	$\sigma = 0.5$	$\sigma = 1$	$\sigma = 1.5$	$\sigma = 2$	$\sigma = 3$	$\sigma = 5$
L2-all	56.98	0.13	0.10	0.10	0.10	0.10	0.10
L2-wo	51.62	36.57	10.72	1.22	0.31	0.16	0.09
L1-wo	57.57	46.21	21.11	4.30	0.57	0.14	0.10
SSSR	55.80	**50.27**	**40.56**	**27.46**	**13.75**	**1.83**	**0.24**