DEEP NEURAL NETWORKS AS FINITE-STEP HOPFIELD DYNAMICS

Wooyul Jung, Hochan Bang, Hee Bin Yoo, Byoung-Tak Zhang
Interdisciplinary Program in Artificial Intelligence, Seoul National University
{wjddnduf5730, hochan0115, yooheebin, btzhang}@snu.ac.kr

ABSTRACT

Energy-based models describe neural computation as descent on a scalar energy function, exemplified by Hopfield networks, whereas modern deep networks are typically viewed as static compositions of nonlinear maps. We show that each layer update can be written as a gradient descent step on a Hopfield-style energy, so network evaluation corresponds to a finite sequence of explicit energy descent steps. Focusing on finite-step execution rather than equilibrium convergence, we identify monotonic energy decrease along forward trajectories as a natural stability criterion. In a weight-shared setting, we relate this condition to the Jacobian of the induced transformation, whose spectral norm governs effective energy smoothness and perturbation amplification. Motivated by this connection, we introduce a Jacobian quantile regularizer that suppresses rare expansive updates during training. Experiments on MNIST show improved robustness to additive perturbations while preserving clean accuracy and producing more consistent finite-step energy evolution.

1 INTRODUCTION

Energy-based models (EBMs) describe neural computation as the evolution of system states under an explicit scalar energy function Hopfield (1982); LeCun et al. (2006); Larochelle & Bengio (2008). In classical Hopfield networks, recurrent updates provably decrease a global energy and converge to stable attractors, establishing a tight link between memory, inference, and optimization. Subsequent developments extended associative memory beyond its original formulations Krotov & Hopfield (2016; 2020), and modern reinterpretations have shown that attention mechanisms admit Hopfield-type energy formulations Ramsauer et al. (2020). These results suggest that energy minimization remains a structurally meaningful principle in neural architectures.

In contrast, modern deep neural networks are typically viewed as static compositions of nonlinear mappings applied layer by layer LeCun et al. (2015). While equilibrium-based approaches reinterpret depth through fixed-point equations Bai et al. (2019); Du & Mordatch (2019), standard network evaluation is rarely analyzed as the explicit execution of an underlying energy descent process. This leaves open the question of whether deep networks can be understood through the same dynamical lens that governs classical energy-based models.

In this work, we show that each layer transformation in a deep neural network can be written as one step of gradient descent on a suitable Hopfield energy defined over hidden representations. As a consequence, evaluating a deep network corresponds to executing a finite sequence of energy descent steps. This perspective reframes depth not merely as functional composition, but as truncated dynamical evolution in state space.

Rather than focusing on convergence to equilibrium, we analyze the finite-step behavior of this explicit energy descent process. In particular, we identify monotonic energy decrease along forward trajectories as a natural primary stability criterion. In a simplified weight-shared setting, we show how this condition can be related to the Jacobian of the induced transformation, whose spectral norm governs both effective energy smoothness and the amplification of small perturbations across repeated updates.

Motivated by these insights, we introduce a training objective that promotes trajectory-level control of the Jacobian spectrum, thereby encouraging monotone energy descent and suppressing expansive updates during finite execution.

2 Deep Neural Networks as Finite-Step Energy Dynamics

We begin by defining the map

$$T(v) := M^\top \sigma(Mv), \tag{1}$$

where $v \in \mathbb{R}^d$ denotes a hidden representation, $M \in \mathbb{R}^{N\times d}$ is a learned weight matrix, and σ is a nonlinear activation function applied elementwise. The associated discrete-time dynamical system is obtained by iteration,

$$v_{k+1} = T(v_k). \tag{2}$$

Maps of this form arise naturally in two-layer networks with tied input–output weights, in associative memory models, and in weight-shared iterative architectures. Rather than viewing T merely as a nonlinear transformation composed layer by layer, we ask whether it admits an interpretation as an explicit descent step on an energy functional defined over hidden states.

2.1 Energy Formulation of the Map

Assume the activation satisfies $\sigma(u) = \nabla\Phi(u)$ for some continuously differentiable potential $\Phi : \mathbb{R}^N \to \mathbb{R}$, which holds for standard choices such as $\tanh$, sigmoid, and ReLU. Following the Universal Hopfield Network formulation Millidge et al. (2022), consider the energy

$$E(v) = \frac{1}{2}\|v\|_2^2 - \Phi(Mv). \tag{3}$$

The quadratic term regularizes the state, while $\Phi(Mv)$ encodes learned interactions through M, mirroring classical Hopfield energies.

Its gradient is

$$\nabla_v E(v) = v - M^\top \nabla\Phi(Mv),$$

so substituting $\sigma = \nabla\Phi$ gives

$$\nabla_v E(v) = v - T(v),$$

revealing T as the negative gradient component of the energy.

A gradient descent step with step size $\alpha > 0$ yields

$$v_{k+1} = v_k - \alpha\nabla_v E(v_k) = (1-\alpha)v_k + \alpha T(v_k). \tag{4}$$

Thus the update interpolates between the identity and T. When $\alpha = 1$, we recover $v_{k+1} = T(v_k)$ exactly, so a standard feedforward layer corresponds to a unit-step descent update on E. For $0 < \alpha < 1$, the dynamics perform a damped descent step.

This shows that T arises from an explicit energy functional, and forward propagation can be viewed as executing a finite sequence of descent steps on a learned landscape rather than as static composition. Unlike classical Hopfield networks, we analyze truncated execution rather than fixed-point convergence: depth corresponds to the number of explicit descent iterations.

Finally, the local behavior of the dynamics is governed by the Hessian

$$\nabla^2 E(v) = I - M^\top H_\Phi(Mv)M,$$

so the Jacobian of the induced map satisfies

$$J_T(v) = M^\top H_\Phi(Mv)M.$$

Energy curvature and local sensitivity are therefore directly linked, enabling analysis of effective smoothness and perturbation growth in the next section.

3 FINITE-STEP ENERGY MONOTONICITY AND STABILITY

Viewing forward evaluation as a finite sequence of explicit gradient descent steps on the energy equation 3 suggests that stability should be formulated in terms of the behavior of this truncated descent process. Rather than requiring convergence to a fixed point, we focus on finite-step energy monotonicity:

$$E(v_{k+1}) \leq E(v_k), \qquad k = 0, \ldots, K-1. \tag{5}$$

When this condition holds along forward trajectories, the induced dynamics behave as a well-posed truncated descent process rather than an unconstrained iterative map. We now derive sufficient conditions under which the residual dynamics guarantee monotone energy decrease during finite execution, and show how these conditions relate directly to Jacobian control.

3.1 ENERGY MONOTONICITY AND EFFECTIVE SMOOTHNESS

Consider the residual update equation 4. Energy decrease is is controlled by the local curvature of E. Using $\sigma(u) = \nabla\Phi(u)$, the Hessian admits the decomposition

$$\nabla^2 E(v) = I - J_T(v), \qquad J_T(v) = M^\top H_\Phi(Mv)M, \tag{6}$$

where $H_\Phi(u) = \nabla^2\Phi(u)$. Thus, the curvature of the energy is directly determined by the Jacobian of the induced map.

Suppose that along a forward trajectory $\{v_k\}_{k=0}^{K-1}$,

$$\|J_T(v_k)\|_2 \leq m. \tag{7}$$

Then

$$\|\nabla^2 E(v_k)\|_2 = \|I - J_T(v_k)\|_2 \leq 1 + m,$$

so the energy is locally $(1+m)$-smooth along this trajectory. Standard descent results imply that if

$$0 < \alpha \leq \frac{1}{1+m}, \tag{8}$$

then

$$E(v_{k+1}) \leq E(v_k)$$

for each step of the finite execution.

This yields a trajectory-level sufficient condition for finite-step energy monotonicity. Stability depends only on the Jacobian norms encountered along forward trajectories; global contraction is unnecessary. Violations arise when large Jacobian realizations inflate the effective smoothness constant so that the fixed step size exceeds the descent regime, breaking the monotonicity guarantee. Because multi-step execution compounds local effects multiplicatively, even rare expansive updates can lead to disproportionate energy increases and error growth over depth.

4 JACOBIAN QUANTILE REGULARIZATION

To operationalize trajectory-level Jacobian control, we penalize large Jacobian spectral norms on mini-batch states. Given a mini-batch $\{v_i\}_{i=1}^B$, define

$$j(v_i) := \|J_T(v_i)\|_2,$$

and let J_q denote the empirical upper q-quantile of $\{j(v_i)\}$. We introduce the Jacobian quantile regularizer

$$\mathcal{R}_{\text{jacq}} = \lambda\,[\max(J_q - m, 0)]^2, \tag{9}$$

where m is a target non-expansiveness threshold.

Unlike penalties that constrain all samples uniformly or enforce global spectral bounds on M, this objective selectively suppresses upper-tail Jacobian outliers. Because perturbation propagation across K steps is governed by products of local Jacobian-dependent factors, even rare expansive realizations can dominate finite-step error growth. By reducing the frequency of states for which $\|J_T(v)\|_2$ exceeds m, the regularizer limits multiplicative perturbation amplification and promotes finite-step energy monotonicity during forward execution without unnecessarily restricting the entire spectrum.

Table 1: Classification accuracy under clean and corrupted inputs for ($K = 1, 3, 5$) induced dynamics. Models are trained on clean MNIST and evaluated on both clean test data and inputs perturbed with additive Gaussian noise. Results are averaged over 10 random seeds.

		$K = 1$		$K = 3$		$K = 5$	
Activation	Method	Test	Noisy	Test	Noisy	Test	Noisy
ReLU	Base	97.90 ± 0.07	68.75 ± 2.70	97.51 ± 0.20	80.32 ± 3.63	97.17 ± 0.20	79.40 ± 3.51
	JacF	97.63 ± 0.22	90.13 ± 1.69	97.39 ± 0.14	65.60 ± 2.61	97.23 ± 0.17	66.93 ± 2.65
	Lip	98.06 ± 0.14	87.67 ± 2.87	97.78 ± 0.33	81.95 ± 2.71	97.33 ± 0.38	81.64 ± 2.23
	JacQ(Ours)	98.04 ± 0.33	**93.97** ± 1.02	97.84 ± 0.24	**90.56** ± 2.52	97.73 ± 0.16	**92.23** ± 1.28
Sigmoid	Base	97.74 ± 0.05	67.59 ± 3.29	97.52 ± 0.13	62.26 ± 2.71	97.04 ± 0.18	62.48 ± 2.71
	JacF	97.06 ± 0.23	72.79 ± 3.00	96.33 ± 0.20	66.68 ± 4.10	95.82 ± 0.42	64.79 ± 3.61
	Lip	98.06 ± 0.08	**83.88** ± 2.74	97.64 ± 0.21	72.33 ± 2.72	96.98 ± 0.33	65.58 ± 1.88
	JacQ(Ours)	97.30 ± 0.21	78.41 ± 5.45	96.75 ± 0.34	**74.18** ± 4.17	96.43 ± 0.31	**75.46** ± 5.63
Tanh	Base	97.64 ± 0.54	66.03 ± 2.43	97.26 ± 0.26	71.28 ± 3.38	96.30 ± 0.50	72.26 ± 2.08
	JacF	96.59 ± 0.29	79.19 ± 5.19	96.58 ± 0.37	71.39 ± 3.11	96.44 ± 0.28	72.99 ± 1.88
	Lip	98.05 ± 0.13	**85.12** ± 3.04	97.37 ± 0.27	76.48 ± 3.12	96.59 ± 0.40	72.84 ± 3.72
	JacQ(Ours)	97.25 ± 0.34	83.12 ± 3.34	96.74 ± 0.20	**77.99** ± 5.18	96.91 ± 0.33	**80.91** ± 5.37

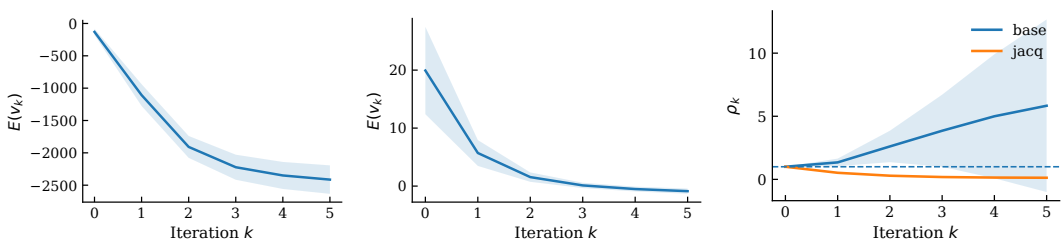
5 EXPERIMENTS

We evaluate Jacobian quantile regularization (JacQ) on MNIST under $K = 1, 3, 5$ induced iterations and compare it against an unregularized baseline (Base), a Frobenius-norm Jacobian penalty (JacF), and a spectral-norm penalty on M (Lip). Performance is measured on clean data and inputs corrupted by additive Gaussian noise ($\sigma = 0.25$). Detailed training settings and step-size choices are provided in Appendix E.

5.1 CLASSIFICATION PERFORMANCE

Table 1 reports clean and corrupted test accuracy across $K = 1, 3, 5$. Clean performance remains comparable across regularizers for all K, indicating that Jacobian-based penalties do not significantly degrade in-distribution accuracy. Under additive Gaussian noise, robustness depends on both activation and iteration depth. For $K = 1$, the best method is depends on the activation (JacQ for ReLU; Lip for Sigmoid and Tanh). As K increases, however, JacQ consistently achieves the highest corrupted accuracy across all activations, and the gap relative to the baseline widens—most notably for ReLU under multi-step dynamics. These results align with the finite-step analysis: as the induced map is iterated, robustness becomes increasingly sensitive to large Jacobian realizations. Suppressing upper-tail spectral growth therefore yields greater stability under repeated updates.

5.2 ENERGY DESCENT AND LOCAL CONTRACTION



(a) Energy: baseline (b) Energy: JacQ (c) Distance ratio ρ_k

Figure 1: **Finite-step energy descent and local perturbation growth.** (a,b) Mean ± standard deviation of the energy $E(v_k)$ across samples along K iterations of the induced dynamics. (c) Distance ratio ρ_k measuring finite-step perturbation amplification.

To directly evaluate the stability properties predicted in Section 3, we examine two trajectory-level diagnostics of the induced finite-step dynamics: (i) energy monotonicity along forward execution, and (ii) perturbation amplification under repeated updates. Unless otherwise stated, we use $K = 5$ iterations with step size $\alpha = 0.5$, $\tanh$ activation, and models trained for 100 epochs.

Energy evolution across trajectories. For each input sample, we generate the trajectory $\{v_k\}_{k=0}^{K}$ under the residual dynamics and compute the energy $E(v_k)$ at every step. Figure 1(a,b) reports the mean $\pm$ standard deviation across samples. Both models decrease energy on average, consistent with the finite-step energy monotonicity condition derived in Section 3. However, the baseline exhibits increasing cross-sample dispersion as k grows. This behavior aligns with the trajectory-level analysis: when $\|J_T(v_k)\|_2$ occasionally becomes large, the effective smoothness constant increases, weakening the descent guarantee and allowing variability to accumulate across iterations. In contrast, JacQ produces substantially tighter energy bands whose variance decreases over iterations. By penalizing the upper quantile of Jacobian spectral norms during training, JacQ suppresses rare expansive updates, thereby maintaining a more consistent effective smoothness along data-dependent trajectories. As a result, forward execution more closely approximates uniform finite-step descent across samples.

Local perturbation propagation. We further evaluate the perturbation bounds discussed in Section 3. For each input, we initialize two nearby states and track their separation under repeated application of the residual map. We measure the distance ratio

$$\rho_k = \frac{\|v_k^{(1)} - v_k^{(2)}\|_2}{\|v_0^{(1)} - v_0^{(2)}\|_2}, \tag{10}$$

which reflects cumulative amplification governed by products of local Jacobian norms. Figure 1(c) shows that JacQ rapidly drives $\rho_k < 1$ and maintains contraction with low dispersion, whereas the baseline exhibits weaker and more variable behavior, including late-stage re-expansion. This empirical trend matches the theory: controlling upper-tail Jacobian realizations reduces violations of the trajectory-level bound $\|J_T(v_k)\|_2 \leq m$, thereby limiting both smoothness inflation and perturbation amplification during finite execution.

6 Discussion and Conclusion

We showed that the layer map $T(v) = M^\top \sigma(Mv)$ can be interpreted as a gradient descent step on a Hopfield-style energy, reframing forward propagation as finite-step dynamical evolution rather than static composition. This shifts stability from fixed-point convergence to finite-horizon energy behavior, with monotonic energy decrease along trajectories as a natural criterion. Our analysis further showed that the Jacobian $J_T(v) = M^\top H_\Phi(Mv)M$, governs both effective energy smoothness and perturbation amplification. Motivated by this, we introduced a Jacobian quantile regularizer that suppresses rare expansive updates, improving robustness under multi-step execution while preserving clean accuracy.

More broadly, this perspective extends naturally to recurrent neural networks with shared iterative dynamics, and to transformer architectures through their links to Hopfield networks. Beyond stability, the energy-based view also offers a lens for interpretability and robustness: forward trajectories correspond to descent on a learned landscape, where deviations from monotonic energy decrease indicate sensitive regions.

Extending this viewpoint to large-scale architectures is promising, as increased depth and sequence length amplify the impact of these effects. Understanding how representations evolve along such trajectories may provide insights into the behavior and failure modes of modern deep networks.

Acknowledgments

This work was partly supported by grants from the IITP (RS-2021-II211343-GSAI/10%, RS-2022-II220951-LBA/15%, RS-2022-II220953-PICA/15%), the NRF (RS-2024-00353991-SPARC/15%, RS-2023-00274280-HEI/15%, RS-2025-25425421/10%), the KEIT (RS-2025-25453780/10%), and the KIAT (RS-2025-25460896/10%), funded by the Korean government.

REFERENCES

Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in neural information processing systems*, 32, 2019.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International conference on machine learning*, pp. 854–863. PMLR, 2017.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in neural information processing systems*, 32, 2019.

Henry Gouk, Eibe Frank, Bernhard Pfahringer, and Michael J Cree. Regularisation of neural networks by enforcing lipschitz continuity. *Machine Learning*, 110(2):393–416, 2021.

Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *arXiv preprint arXiv:1912.03263*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

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.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Daniel Jakubovitz and Raja Giryes. Improving dnn robustness to adversarial attacks using jacobian regularization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 514–529, 2018.

Dmitry Krotov and John Hopfield. Large associative memory problem in neurobiology and machine learning. *arXiv preprint arXiv:2008.06996*, 2020.

Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in neural information processing systems*, 29, 2016.

Dmitry Krotov, Benjamin Hoover, Parikshit Ram, and Bao Pham. Modern methods in associative memory. *arXiv preprint arXiv:2507.06211*, 2025.

Hugo Larochelle and Yoshua Bengio. Classification using discriminative restricted boltzmann machines. In *Proceedings of the 25th international conference on Machine learning*, pp. 536–543, 2008.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, Fujie Huang, et al. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

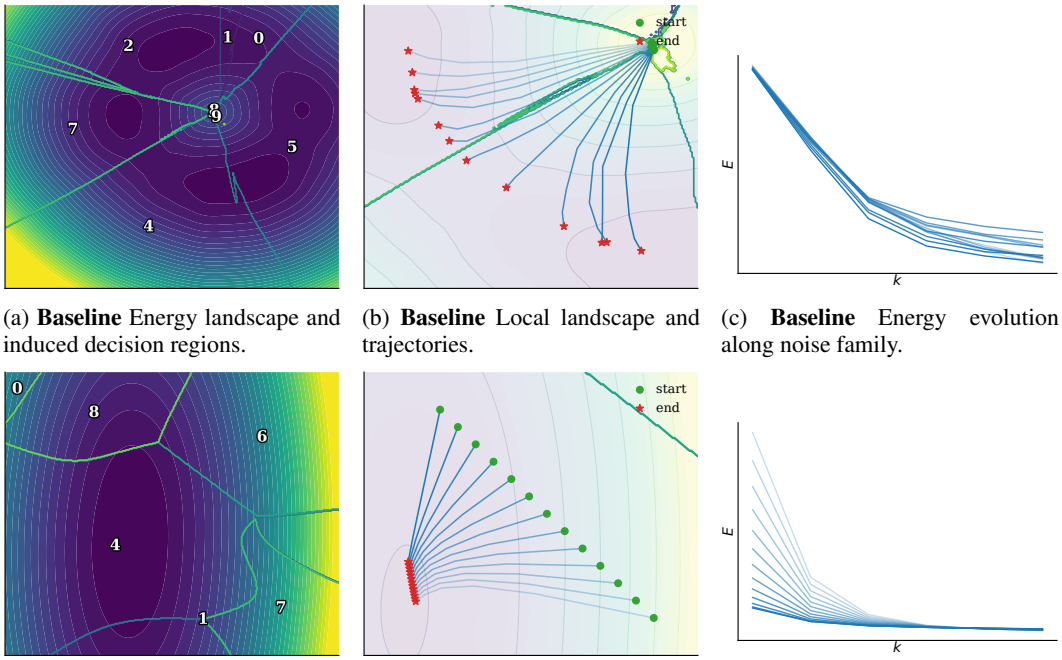
Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal hopfield networks: A general framework for single-shot associative memory models. In *International Conference on Machine Learning*, pp. 15561–15583. PMLR, 2022.

Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, et al. Hopfield networks is all you need. *arXiv preprint arXiv:2008.02217*, 2020.

Yuichi Yoshida and Takeru Miyato. Spectral norm regularization for improving the generalizability of deep learning. *arXiv preprint arXiv:1705.10941*, 2017.

A VISUALIZATION OF ENERGY GEOMETRY AND TRAJECTORIES



(a) **Baseline** Energy landscape and induced decision regions.

(b) **Baseline** Local landscape and trajectories.

(c) **Baseline** Energy evolution along noise family.

(d) **JacQ** Energy landscape and induced decision regions.

(e) **JacQ** Local landscape and trajectories.

(f) **JacQ** Energy evolution along noise family.

Figure 2: **Finite-step energy geometry under progressive perturbations.** Hidden states generated by the induced dynamics $v_{k+1} = T(v_k)$ are projected onto a shared two-dimensional PCA subspace. The top row corresponds to the baseline model, and the bottom row to JacQ. **(a,d)** Energy landscape $E(v)$ restricted to the PCA slice together with induced decision regions obtained after K forward steps. **(b,e)** Local landscape with trajectories initialized from a clean input and progressively corrupted variants (start: green circle; end: red star). Trajectory color intensity decreases as the input noise standard deviation increases from $\sigma = 0$ to $\sigma = 0.55$. **(c,f)** Energy sequences $E(v_k)$ along the same noise-level trajectories, with curves ordered by increasing noise level.

To complement the quantitative analysis, we visualize the finite-step dynamics $v_{k+1} = T(v_k)$ under $K = 5$, $\alpha = 0.5$, $\tanh$ activation, trained for 100 epochs. For each model, hidden states $\{v_k(x)\}_{k=0}^{K}$ are collected over a batch, and PCA is fit to their union. The top two components define a shared projection plane for Figure 2.

The left panels show the energy $E(v)$ restricted to this PCA slice together with induced decision regions obtained after K forward steps. The baseline landscape appears irregular with uneven decision regions, whereas JacQ yields smoother geometry and more uniformly shaped basins, indicating reduced local sensitivity of the induced dynamics. The center panels display trajectories from a clean input and progressively corrupted variants. In the baseline, trajectories separate rapidly and frequently cross induced boundaries, reflecting stronger perturbation amplification. Under JacQ, trajectories remain clustered within the same basin, consistent with stronger local contraction. The right panels show the corresponding energy sequences $E(v_k)$. The baseline exhibits more variable and occasionally non-monotonic energy evolution across noise levels, while JacQ produces more consistent finite-step descent.

Overall, the visualization supports the Jacobian-based analysis: limiting large Jacobian realizations reduces perturbation growth and yields smoother finite-step energy geometry.

B Perturbation Amplification and the Role of Jacobian Quantile Control

We formalize the perturbation amplification mechanism underlying finite-step instability and explain how Jacobian quantile regularization mitigates it.

B.1 Finite-Step Perturbation Growth

Consider two nearby trajectories under the residual dynamics

$$v_{k+1} = v_k - \alpha \nabla E(v_k), \quad \delta v_k = v_k^{(1)} - v_k^{(2)}.$$

Linearization yields

$$\delta v_{k+1} = \big(I - \alpha \nabla^2 E(v_k)\big)\delta v_k.$$

Using $\nabla^2 E(v) = I - J_T(v)$,

$$\delta v_{k+1} = \big((1-\alpha)I + \alpha J_T(v_k)\big)\delta v_k.$$

Hence

$$\|\delta v_K\|_2 \le \left(\prod_{k=0}^{K-1} \big\|(1-\alpha)I + \alpha J_T(v_k)\big\|_2\right) \|\delta v_0\|_2.$$

Perturbation growth is therefore governed by a product of local Jacobian-dependent operators along the trajectory.

B.2 Energy Descent Breakdown

Along the trajectory, the energy is locally L_k-smooth with

$$L_k = 1 + \|J_T(v_k)\|_2.$$

Monotone descent is guaranteed when

$$\alpha \le \frac{1}{L_k}.$$

If $\|J_T(v_k)\|_2$ becomes large, this condition may be violated, and the operator

$$(1-\alpha)I + \alpha J_T(v_k)$$

can become expansive. Because multi-step execution compounds these effects multiplicatively, even rare large Jacobian realizations can dominate the product bound and induce disproportionate energy increase and perturbation amplification.

B.3 Effect of Jacobian Quantile Regularization

Jacobian quantile regularization reduces the upper tail of $\|J_T(v)\|_2$ over data-dependent states. By suppressing rare expansive realizations, it lowers the probability that $\alpha > 1/L_k$ along forward trajectories and limits the magnitude of individual factors in the product bound. Consequently, both finite-step energy monotonicity and perturbation contraction are stabilized without enforcing global contraction over the entire state space.

C Activation Potentials and Curvature Bounds

Table 2 lists common activation functions, their associated potentials Φ satisfying $\sigma(u) = \nabla\Phi(u)$, their Hessians, and the corresponding uniform curvature constants L_Φ.

Table 2: Activation functions, associated potentials, and curvature bounds.

Activation $\sigma(u)$	Potential $\Phi(u)$	Hessian $H_\Phi(u)$	L_Φ
$\tanh(\beta u)$	$\sum_i \frac{1}{\beta}\log\cosh(\beta u_i)$	$\mathrm{diag}\big(\beta(1-\tanh^2(\beta u_i))\big)$	β
$\mathrm{sigmoid}(\beta u)$	$\sum_i \frac{1}{\beta}\log(1+e^{\beta u_i})$	$\mathrm{diag}(\beta s_i(1-s_i))$	$\beta/4$
$\mathrm{ReLU}(u)$	$\sum_i \frac{1}{2}\mathrm{ReLU}(u_i)^2$	$\mathrm{diag}(\mathbb{I}[u_i>0])$	1

D Related Work

D.1 Energy-Based and Dynamical Perspectives

Energy-based models formulate computation as descent on an explicit scalar functional LeCun et al. (2006). The classical Hopfield network introduced recurrent state updates that monotonically decrease a global energy and converge to stable attractors Hopfield (1982). Subsequent work expanded associative memory capacity while preserving an energy formulation Krotov & Hopfield (2016; 2020); Krotov et al. (2025), and modern reinterpretations showed that attention mechanisms admit Hopfield-type energy views Ramsauer et al. (2020). Universal Hopfield Networks unified these variants under a common energy functional and update rule Millidge et al. (2022). In parallel, deep networks have been studied as dynamical systems defined by fixed points or continuous flows. Deep Equilibrium Models treat outputs as solutions to fixed-point equations Bai et al. (2019), energy-based generative models reinterpret classifiers as energy functions Du & Mordatch (2019); Grathwohl et al. (2019), and Neural ODEs model depth as continuous-time dynamics Chen et al. (2018). These works focus on recurrent convergence or equilibrium solutions. By contrast, we analyze explicit discrete-time dynamics executed for a finite number of iterations and study their stability through the structure of the induced energy functional.

D.2 Jacobian Control and Stability

Instability of deep networks has long been linked to gradient propagation across layers Bengio et al. (1994). Architectural mechanisms such as residual connections modify effective layer-to-layer dynamics to improve signal preservation He et al. (2016), and initialization strategies influence stability of rectified activations He et al. (2015). Robustness and sensitivity control have been pursued through spectral or Lipschitz constraints Yoshida & Miyato (2017); Cisse et al. (2017); Gouk et al. (2021), as well as through Jacobian-based regularization that penalizes local derivative norms Alain & Bengio (2014); Jakubovitz & Giryes (2018). These approaches regulate the network primarily as a static input–output mapping. Our work instead analyzes stability of the induced discrete-time dynamics arising from repeated layer application and derives its Jacobian structure directly from an explicit energy formulation.

E Additional Experimental Details

All experiments use the update rule in equation 1 with hidden dimension $N = 128$. For $K = 1$, we use step size $\alpha = 1$, corresponding to a full gradient step on the induced energy. For multi-step dynamics ($K = 3, 5$), we set $\alpha = 0.5$ to ensure stable finite-step evolution. After K iterations, the resulting state v_K is fed to a linear classifier. Models are trained end-to-end using cross-entropy loss and optimized with Adam for 100 epochs. Unless otherwise stated, we report results using the final model at the end of training. Each configuration is trained with 10 random seeds, and we report mean and standard deviation across seeds. Robustness is evaluated under additive Gaussian noise with standard deviation 0.25, applied only at test time.