Bifurcation Models: Learning Set-Valued Solution Maps with Weight-Tied Dynamics

Caleb Jore
University of Central Florida
Orlando, FL 32826
calebjore@gmail.com

Jialin Liu*
University of Central Florida
Orlando, FL 32826
jialin.liu@ucf.edu

Abstract

Many scientific and optimization instances define deterministic sets of admissible solutions through governing equations, constraints, symmetries, or optimality conditions. We study learning such set-valued solution maps with one weight-tied operator whose initialization-dependent trajectories can converge to different valid branches. We prove representability of broad finite-branch maps with locally Lipschitz branches and almost-everywhere regularity of the induced selectors, while unconstrained branch selection can create arbitrarily irregular targets. Experiments show that these dynamics can discover multiple low-energy Ising solutions without complete branch information and outperform single-branch fitting. Allen–Cahn experiments show that diversity is not automatic: explicit promotion reveals an accuracy–diversity tradeoff.

1 Introduction

Many scientific, geometric, and optimization problems define a set of admissible solutions through equations, constraints, symmetries, or optimality conditions. A frustrated spin system may admit many low-energy configurations [22]; a nonlinear PDE may have multiple steady states [35]; an inverse-design problem may admit several valid designs [40]. We study learning one parameterized solver across such problem instances. The target is a deterministic set-valued solution map

$$\mathcal{F} : \mathcal{K} \rightrightarrows \mathcal{Y}, \qquad \boldsymbol{x} \mapsto \mathcal{F}(\boldsymbol{x}),$$

where $\mathcal{F}(\boldsymbol{x})$ is the set of admissible solutions for the same input $\boldsymbol{x}$.

A common single-branch fitting pipeline selects $u(x) \in \mathcal{F}(x)$ and trains a predictor to imitate it. This is benign when a regular selector is known, but in many multi-solution problems no such selector is specified: labels may come from solver initialization [9, 43], a symmetry-breaking convention [23], or an arbitrary choice among valid solutions. The labeling procedure therefore defines a hidden selector, which may switch between branches as x varies and inject artificial irregularity into the learning problem. An alternative is to learn the target as set-valued. Neural architectures for sets can represent and process unordered collections [32, 41]. In multi-solution scientific problems, however, the target is not merely an unordered list but a solution set whose elements may correspond to distinct physical, geometric, or combinatorial branches. This raises a basic question: *should multi-solution problems be represented by choosing one branch, or by learning the solution set itself?*

We study this question through the lens of weight-tied parameterized dynamics. Starting from an initial state $\boldsymbol{y}_0$ that may be fixed, sampled, or chosen by the user, the model iterates

$$\boldsymbol{y}_{t+1} = g_\theta(\boldsymbol{y}_t, \boldsymbol{x}), \qquad t = 0, 1, 2, \ldots.$$

*The authors are listed alphabetically. Corresponding author: Jialin Liu, jialin.liu@ucf.edu.

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

Here, g_θ may itself be a multi-layer neural module, such as an MLP, GNN, or neural operator. The term ***weight-tied*** means that, when the dynamics are unrolled over time, the same parameters θ are reused at every iteration t, rather than using a different layer at each depth.

For a fixed input $\boldsymbol{x}$, different initializations $\boldsymbol{y}_0$ may converge to different stable equilibria. Thus the model does not need to output a single solver-selected branch. Instead, it can represent the attractor set

$$\mathcal{A}_\theta(\boldsymbol{x}) := \{\boldsymbol{y}_\star \in \mathcal{Y} : \boldsymbol{y}_\star \text{ is a locally attracting fixed point of } g_\theta(\cdot, \boldsymbol{x})\}, \tag{1}$$

where locally attracting means that every initialization in some neighborhood of the fixed point converges to it. We write $g^t(\boldsymbol{y}_0, \boldsymbol{x})$ for t updates starting from $\boldsymbol{y}_0$. For the same input $\boldsymbol{x}$, trajectories from different initial states $\boldsymbol{y}_0$ may split over iterations into different basins of attraction and converge to different solution branches. We refer to this viewpoint as a ***bifurcation model***.

Rather than introducing a new architecture, we focus on three fundamental questions:

(Q1) **Representation.** Can weight-tied models represent set-valued mappings in a regular, learnable way? More precisely, for what class of set-valued maps $\mathcal{F} : \mathcal{K} \rightrightarrows \mathcal{Y}$ does there exist an operator $g : \mathcal{Y} \times \mathcal{K} \to \mathcal{Y}$ whose stable fixed points recover $\mathcal{F}(\boldsymbol{x})$?

(Q2) **Branch selection.** If such a representation exists, and suppose all solution branches are available during training. Should we learn the whole set-valued map, or select one element of $\mathcal{F}(\boldsymbol{x})$ as the label and train a single-valued predictor? What regularity or stability can be lost when a set-valued problem is collapsed into a manually chosen solution selector?

(Q3) **Discovery without complete branch information.** In practice, the full set of branches is often not observed. Given only an unsupervised physical objective and multiple initializations, will a weight-tied dynamic model automatically discover multiple solutions without full branch information during training?

Centered around the three questions, we contribute the following:

- **Set-Valued Representation.** We prove that broad finite-branch solution maps with locally Lipschitz branches admit regular weight-tied dynamics: outside unstable-collapse inputs, almost every initialization converges to a valid branch, and varying it recovers all branches.
- **Selector Regularity.** Unconstrained branch selection can create arbitrarily irregular targets. In contrast, the selectors induced by our constructed dynamics are locally Lipschitz almost everywhere on the stable domain.
- **Unsupervised Branch Discovery.** Energy-only training discovers multiple low-energy Ising solutions, but Allen–Cahn dynamics can collapse to a dominant branch. Explicit diversity promotion enables broader discovery with an accuracy–diversity tradeoff.

Here we discuss four lines of related works. The first line of work involves implicit or equilibrium models (e.g., [4, 7, 10, 15, 12]). While these models usually focus on designing network layers that converge to a single, unique fixed point for deterministic prediction [7, 37, 21, 33, 18, 27], here we treat the model as a dynamical system explicitly capable of representing an entire set of valid solutions through multiple stable equilibria. The second line of work is unrolling networks (e.g., [31, 16, 39, 38, 29, 42, 1, 26]), including recurrent networks [5, 36], looped transformers [13, 11], etc. We share their computational realization of unrolling weight-tied iterations during training, but our focus is fundamentally different: rather than optimizing a finite-depth refinement map, we study the limiting attractor structure of the shared operator to capture multi-basin landscapes. The third line of related work, [24], also uses an iterated map to recover multiple optima, but specifically aims to emulate the proximal-point algorithm for an explicit objective. In contrast, we study weight-tied dynamics more broadly as a fundamental representation of set-valued solution maps, analyzing their representational capacity, selector regularity, and unsupervised branch discovery. Finally, [43] uses deep ensembles of PINNs to discover multiple PDE solutions, which requires training an independent ensemble for each individual problem instance. Conversely, we learn an amortized solver over a distribution of instances, where solution multiplicity can arise from different state initializations within a single shared weight-tied model.

Scope. Candidate solutions are independently verifiable. We study structural solution multiplicity, rather than semantic multi-label classification, annotation noise, or learning $p(\boldsymbol{y} \mid \boldsymbol{x})$. Random initialization explores attraction basins; its induced branch probabilities need not equal a ground-truth

conditional distribution. Noisy inputs, stochastic admissibility, and calibrated probabilities require additional modeling; see Appendix G.

Section 2 analyzes weight-tied dynamics independently of specific neural parameterizations. Sections 3 and 4 instantiate the same principle using different parameterized operators.

2 Theoretical Foundations

In this section, we address (Q1) and (Q2) theoretically. We consider finite-branch set-valued maps $\mathcal{F}(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), \cdots, f_n(\boldsymbol{x})\}$, allowing different branches to coincide at some inputs.

2.1 Can Weight-Tied Dynamics Represent Set-Valued Maps?

To isolate the fundamental capacity of weight-tied dynamics from specific network structures, we study this in a generic nonparametric setting; thus, we drop the parameter subscript θ and study the existence/properties of general operators g. Accordingly, the attractor set in (1) can be written as

$$\mathcal{A}_g(\boldsymbol{x}) := \{\boldsymbol{y}_\star \in \mathcal{Y} : \boldsymbol{y}_\star \text{ is a locally attracting fixed point of } g(\cdot, \boldsymbol{x})\}. \tag{2}$$

We now formalize (Q1) in this context: *Let $\mathcal{F}$ be a set-valued map. Does there exist g such that its attractor set exactly recovers the target, i.e., $\mathcal{A}_g(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{x})$, for all $\boldsymbol{x} \in \mathcal{K}$?*

In fact, a naive answer satisfies the above condition. Consider the following g with $0 < \eta < 1$:

$$g(\boldsymbol{y}, \boldsymbol{x}) := (1-\eta)\boldsymbol{y} + \eta \Pi_{\mathcal{F}(\boldsymbol{x})}(\boldsymbol{y}) \tag{3}$$

where $\Pi_{\mathcal{F}(\boldsymbol{x})}(\boldsymbol{y})$ denotes a projection of $\boldsymbol{y}$ onto $\mathcal{F}(\boldsymbol{x})$, with an arbitrary fixed tie-breaking rule when the projection is nonunique. Because this operator explicitly forces $\boldsymbol{y}$ toward the closest point in the target set, any trajectory will unconditionally converge to a branch in $\mathcal{F}(\boldsymbol{x})$, and sweeping across all initializations $\boldsymbol{y}_0$ will successfully recover every branch.

However, this naive construction is not practically insightful. Let's consider an example: $\mathcal{K} = \mathcal{Y} = \mathbb{R}$ and $\mathcal{F}(x) = \{x+1, x-1\}$ with two branches $f_1(x) = x+1$ and $f_2(x) = x-1$. Then (3) becomes

$$g(y, x) = (1-\eta)y + \eta(x+1)\mathbb{I}(y \geq x) + \eta(x-1)\mathbb{I}(y < x)$$

Now we fix $y = 0$. It can be shown that $g(0, x)$ is not Lipschitz continuous near $x = 0$:

$$\lim_{x \to 0^+} g(0,x) = -\eta, \quad \lim_{x \to 0^-} g(0,x) = \eta, \quad \lim_{\delta \to 0} \sup_{x_1, x_2 \in \mathbb{B}(0,\delta)} \frac{|g(0,x_1) - g(0,x_2)|}{|x_1 - x_2|} = \infty$$

Therefore, $(0,0)$ is a singular point of $g(0,x)$. In fact, all the points on the switching boundaries $\{(x,y) : y = x\}$ are singular. This reveals a key limit of this naive construction: *it's hard to parameterize such an operator g with neural networks.* While neural networks have the ability to approximate operators with singular points, doing so demands much higher network complexity—i.e., increasing depth/width to capture the growing steepness near the singularity (see, e.g., [34]).

Therefore, (Q1) asks whether such a representation exists with an input-Lipschitz operator, making it easier to parameterize with neural networks.

Our findings show an affirmative result, provided we exclude the *unstable collapse set* $\mathcal{U}$: a specific subset of inputs where branch intersections are not locally constant.

Definition 2.1 (Stable and Unstable Collapse)**.** Let $f_1, \ldots, f_n : \mathcal{K} \to \mathcal{Y}$ be branch maps. A point $\boldsymbol{x} \in \mathcal{K}$ exhibits a *stable collapse pattern* if there exists a neighborhood around $\boldsymbol{x}$ wherein the equality relationship between any pair of branches f_i and f_j remains constant. We denote the set of all such points as the stable set $\mathcal{S}$. Its complement, $\mathcal{U} = \mathcal{K} \setminus \mathcal{S}$, is defined as the *unstable collapse set*.

The unstable-collapse set need not have measure zero. For $f_1(x) = 0$ and $f_2(x) = \mathrm{ReLU}(x)$, it is $\mathcal{U} = \{0\}$. By contrast, if $f_2(x) = \mathrm{dist}(x, \mathcal{C})$ for a fat Cantor set $\mathcal{C}$, then $\mathcal{U} = \mathcal{C}$ has positive measure. Our convergence and selector guarantees apply on the stable domain, without assuming that its complement has measure zero.

The following theorem gives such an input-regular operator; its proof is in Appendix A.

Theorem 2.2. *Let $\mathcal{K} \subset \mathbb{R}^d$ be bounded and $\mathcal{Y} = \mathbb{R}^m$. Assume the branch maps $f_1, \ldots, f_n : \mathcal{K} \to \mathcal{Y}$ are locally Lipschitz. Then there exists an operator $g : \mathcal{Y} \times \mathcal{K} \to \mathcal{Y}$ satisfying:*

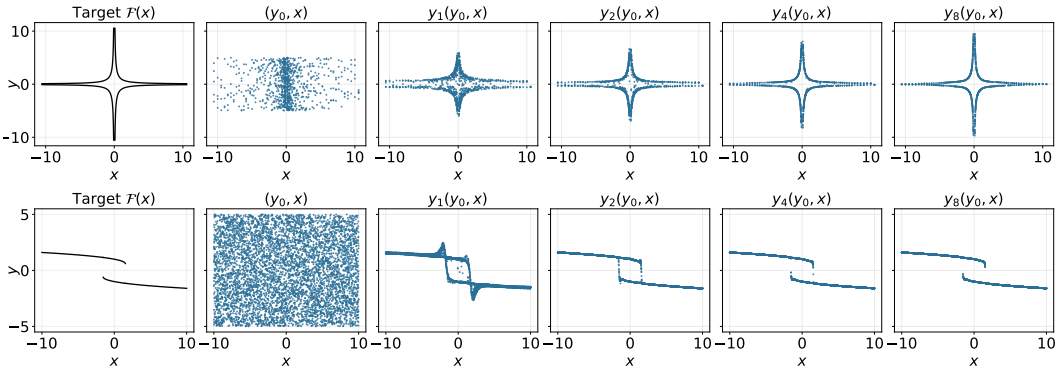

Figure 1: Numerical verification of Theorem 2.2. Each row shows one toy set-valued map. The first column shows the target branches $\mathcal{F}(x)$; the second column shows sampled initial states y_0; and the remaining columns show the iterates y_1, y_2, y_4, y_8 produced by the same weight-tied update map. For each fixed input x, different random initializations move toward different admissible branches. As the iteration proceeds, the point cloud aligns with the full target set.

1. *(Input Regularity). For every $\boldsymbol{y} \in \mathcal{Y}$, the map $\boldsymbol{x} \mapsto g(\boldsymbol{y}, \boldsymbol{x})$ is globally Lipschitz on $\mathcal{K}$.*
2. *(Reliable Convergence). For every stable input $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$, the iteration $\boldsymbol{y}_{t+1} = g(\boldsymbol{y}_t, \boldsymbol{x})$ converges to a valid branch in $\mathcal{F}(\boldsymbol{x})$ for Lebesgue-almost every initialization $\boldsymbol{y}_0 \in \mathcal{Y}$. (The exceptional initializations form a zero-measure set $\Sigma^\circ(\boldsymbol{x})$).*
3. *(Attractor Completeness). For every $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$, varying the initialization completely recovers the target solution set. Formally, $\mathcal{A}_g(\boldsymbol{x}) = \{\lim_{t\to\infty} g^t(\boldsymbol{y}_0, \boldsymbol{x}) : \boldsymbol{y}_0 \in \mathcal{Y} \setminus \Sigma^\circ(\boldsymbol{x})\} = \mathcal{F}(\boldsymbol{x})$.*

Theorem 2.2 requires locally Lipschitz branches and a bounded domain, which need not be compact or connected. For example, $\log x$ on $(0, 1]$ and $1/x$ on $(-1, 0) \cup (0, 1)$ are locally but not globally Lipschitz. The guarantees exclude unstable-collapse inputs and exceptional initializations. The construction collects singularities and switching boundaries in the bad set of Definition A.10. As the margin to that set shrinks, the convergence bound in Theorem A.20 can weaken. Residual-based stopping, longer rollouts, continuation, or domain decomposition may help in practice, but do not extend the theorem to inputs violating its assumptions.

Numerical verification. As an initial verification of Theorem 2.2, we evaluate two toy multi-solution problems in an idealized setting where all valid labels are known. Crucially, we do not force the model to fit a predefined branch for a given input x. Instead, we supply the full set of labels (Algorithm 1), sample random initializations y_0, dynamically match each initialization to its nearest label, and train the weight-tied dynamics to converge accordingly. Details are in Appendix D.

Figure 1 provides the numerical evidence. In both toy examples, the iterates of the learned weight-tied dynamics evolve regularly with respect to (y_0, x), and different initializations can converge to different valid branches of the target set-valued map. At the same time, the collection of converged states recovers the full branch structure rather than collapsing to a single solution. This is exactly the behavior predicted by Theorem 2.2: a single regular weight-tied dynamics can represent a multi-branch function, and the branch realized at convergence can depend on the initialization while remaining consistent with the target set-valued mapping. Quantitative results are given in Table 1.

2.2 When is Multi-Branch Learning Preferable?

Let us now address (Q2): if multiple valid solution branches exist, what is the fundamental theoretical benefit of representing the entire set-valued map rather than isolating a single branch? Returning to the recurrent operator g constructed in Theorem 2.2, we find that multi-branch dynamics provide a natural, built-in regularization. Fixing an initialization $\boldsymbol{y}_0$ implicitly defines a solution selector $u(\boldsymbol{y}_0, \boldsymbol{x})$ that routes the input to a specific branch. The following theorem guarantees that this dynamically induced selector is exceptionally well-behaved—it is locally Lipschitz almost everywhere.

Theorem 2.3 (Lipschitz selector)**.** *Let $\mathcal{K} \subset \mathbb{R}^d$ be bounded and measurable, $\mathcal{Y} = \mathbb{R}^m$, and $f_1, \ldots, f_n : \mathcal{K} \to \mathcal{Y}$ be locally Lipschitz branch maps. Let $\mathcal{K}^\star := \mathcal{K} \setminus \mathcal{U}$ denote the stable domain, and let $g : \mathcal{Y} \times \mathcal{K} \to \mathcal{Y}$ be the operator constructed in Theorem 2.2. For $\boldsymbol{y}_0 \in \mathcal{Y}$, define*

Table 1: Test error for the bifurcation-model selector and manually selected branch targets. The manual selectors alternate between branches over N_{int} intervals. $N_{\text{int}} \in \{2, 4, 8, 16, 32, 64\}$.

Problem	Bifurcation	Manual selector: N_{int}					
		2	4	8	16	32	64
Toy example 1	0.0028	0.0181	0.1847	0.8982	5.7052	9.6052	10.5995
Toy example 2	0.0002	0.0159	0.0321	0.2401	0.2698	0.5019	0.5767

$u(\boldsymbol{y}_0, \boldsymbol{x}) := \lim_{t\to\infty} g^t(\boldsymbol{y}_0, \boldsymbol{x})$. *Then, for Lebesgue-almost every initialization* $\boldsymbol{y}_0 \in \mathcal{Y}$*, it holds that*

$$\Big|\big\{\boldsymbol{x} \in \mathcal{K}^\star : \boldsymbol{x} \mapsto u(\boldsymbol{y}_0, \boldsymbol{x}) \text{ is not locally Lipschitz at } \boldsymbol{x}\big\}\Big| = 0$$

Here, $|\cdot|$ *denotes the* d*-dimensional Lebesgue measure.*

Theorem 2.3 says that, for almost every initialization, the recurrent dynamics does not choose branches in an arbitrary way. It may switch between branches, but the set of inputs where the induced selector fails to be locally Lipschitz has measure zero. This provides a structural contrast with the single-branch fitting reduction of a multi-solution problem. In that reduction, one first constructs a manual selector $u : \mathcal{K}^\star \to \mathcal{Y}$ by choosing $u(\boldsymbol{x}) \in \mathcal{F}(\boldsymbol{x})$ for every input $\boldsymbol{x}$. Such a choice may come from a solver initialization, an ordering convention, a symmetry-breaking rule, or simply from whichever solution is observed. Unless additional regularity is imposed, there is no reason for this selector to respect the geometry of the branches: nearby inputs can be assigned to far-apart solutions even when all underlying branches are regular. The following theorem formalizes this failure mode.

Theorem 2.4. *Let* $\mathcal{K}^\star \subset \mathbb{R}^d$ *be bounded and measurable, and let* $\mathcal{Y} = \mathbb{R}^m$*. Let* $f_1, \ldots, f_n : \mathcal{K}^\star \to \mathcal{Y}$ *be locally Lipschitz branch maps. Define the genuinely multi-branch region*

$$\mathcal{M} := \Big\{\boldsymbol{x} \in \mathcal{K}^\star : \exists i \neq j \text{ such that } f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x})\Big\}.$$

Then, for every $a \in [0, |\mathcal{M}|]$*, there exists a measurable solution selector* $u_a : \mathcal{K}^\star \to \mathcal{Y}$ *such that*

$$u_a(\boldsymbol{x}) \in \mathcal{F}(\boldsymbol{x}), \ \text{for all } \boldsymbol{x} \in \mathcal{K}^\star, \qquad \text{and } \Big|\operatorname{Disc}_{\mathcal{K}^\star}(u_a)\Big| = a.$$

Here, $\operatorname{Disc}_{\mathcal{K}^\star}(u_a)$ *denotes the set of discontinuity points of* u_a *relative to the domain* $\mathcal{K}^\star$*.*

Theorem 2.4 identifies a possible failure mode of unconstrained branch selection. It is a worst-case existence result, not a genericity theorem: it does not assert that practical selectors are always discontinuous or that their discontinuities necessarily prevent learning.

Importantly, this does not mean multi-branch learning should unconditionally replace conventional supervised learning. If a smooth selector is known, or if the task explicitly requires convergence to a particular branch, then single-branch supervision is a natural objective. However, when no such selector is available, forcing the model to imitate one arbitrary branch can introduce avoidable jumps.

Numerical verification. We use a controlled stress test to isolate the effect of irregular branch assignment, rather than to model a typical dataset. On the same two toy examples, we partition the input domain into N_{int} intervals and alternate the selected branch between adjacent intervals. Increasing N_{int} introduces more artificial switching; we compare fitting these targets with learning the full solution set.

Figure 2 shows the effect: the selector induced by the implicit dynamics remains smooth and natural, while the human-designed selectors become progressively more irregular. Table 1 shows the same phenomenon quantitatively: as the interval count increases, test errors grow substantially. These experiments validate Theorem 2.3 by illustrating that the selector induced by the weight-tied dynamics is regular, and they validate Theorem 2.4 by showing that arbitrary branch assignment can create artificial discontinuities that make the learning problem significantly harder.

3 Discovering Multiple Solutions Without Complete Branch Information

The toy experiments in Section 2 verify the theory in an idealized setting: for each input $\boldsymbol{x}$, all valid branch labels $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$ are known, and each sampled initialization $\boldsymbol{y}_0$ is trained to converge to

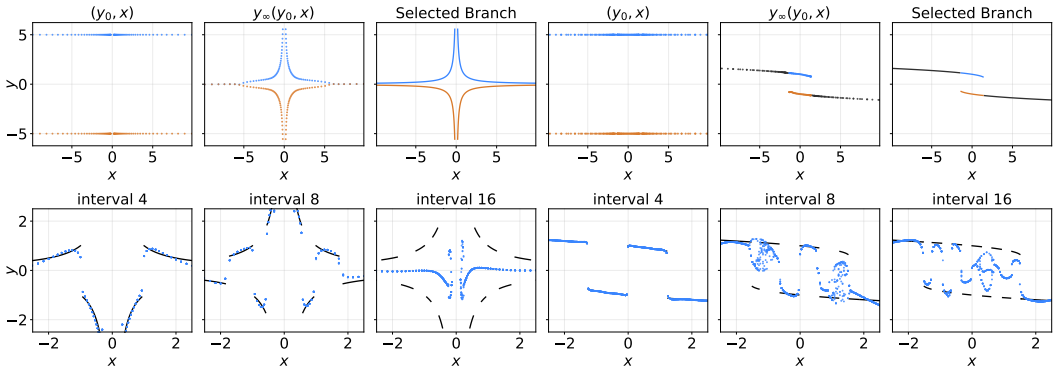

Figure 2: Comparison of dynamical (top row) and manual (bottom row) solution selectors for Example 1 (left) and Example 2 (right). The dynamic model naturally maps a fixed initialization y_0 to a regular converged state $y_\infty(y_0, x)$. Conversely, manual selection introduces arbitrary switching and artificial discontinuities as the interval increases, hindering training and generalization.

Algorithm 1 Multi-branch fitting

Require: Full branch data $\mathcal{D} = \left\{\boldsymbol{x}_i, \{\boldsymbol{y}_{i,j}\}_{j=1}^{n_i}\right\}_i$

1: **for** each mini-batch $\mathcal{B} \subset \mathcal{D}$ **do**

2: Sample initialized states $\boldsymbol{y}_{i,k}^{(0)} \sim \rho$ for $\left(\boldsymbol{x}_i, \{\boldsymbol{y}_{i,j}\}_{j=1}^{n_i}\right) \in \mathcal{B}, \quad k = 1, \cdots, M$

3: Choose the closest branch: $\boldsymbol{y}_{i,k}^\star = \arg \min_{\boldsymbol{y} \in \{\boldsymbol{y}_{i,1}, \cdots, \boldsymbol{y}_{i,n_i}\}} \|\boldsymbol{y} - \boldsymbol{y}_{i,k}^{(0)}\|$

4: Unroll for $t = 0, 1, \cdots, T-1$: $\boldsymbol{y}_{i,k}^{(t+1)} = g_\theta\left(\boldsymbol{y}_{i,k}^{(t)}, \boldsymbol{x}_i\right)$

5: Minimize $\mathcal{L}(\theta) := \text{mean}\left(\|\boldsymbol{y}_{i,k}^{(T)} - \boldsymbol{y}_{i,k}^\star\|_2^2\right)$

6: **end for**

Algorithm 2 Multi-branch discovery

Require: Input data $\mathcal{D} = \{\boldsymbol{x}_i\}_i$

1: **for** each mini-batch $\mathcal{B} \subset \mathcal{D}$ **do**

2: Sample initialized states $\boldsymbol{y}_{i,k}^{(0)} \sim \rho$ for $\boldsymbol{x}_i \in \mathcal{B}, \quad k = 1, \cdots, M$

3: Unroll for $t = 0, 1, \cdots, T-1$: $\boldsymbol{y}_{i,k}^{(t+1)} = g_\theta\left(\boldsymbol{y}_{i,k}^{(t)}, \boldsymbol{x}_i\right)$

4: Minimize $\mathcal{L}(\theta) := \text{mean}\left(E\left(\boldsymbol{y}_{i,k}^{(T)}, \boldsymbol{x}_i\right)\right)$

5: **end for**

its nearest branch label. In realistic scientific systems, however, enumerating all solution branches is usually impossible or prohibitively expensive. What is often available instead is an unsupervised objective $E(\boldsymbol{y}, \boldsymbol{x})$, such as an energy functional, whose low values indicate physically valid solutions.

We therefore replace the nearest-branch fitting loss in Algorithm 1 with the unsupervised procedure in Algorithm 2. For each input $\boldsymbol{x}$, we sample multiple initial states $\boldsymbol{y}_0$, unroll the same weight-tied dynamics, and minimize the terminal physical objective $E(\boldsymbol{y}_T, \boldsymbol{x})$. No branch labels are provided, and the dynamics are free to converge to any physically valid branch.

This raises several questions. *Can a branch-flexible recurrent model discover multiple solutions for the same input without being shown all branches? Does unsupervised branch-flexible training outperform supervised fitting to a single solver-selected label? And does avoiding a prescribed branch make the learned solution map less brittle?* We study these questions below in a frustrated Ising system, where many low-energy spin configurations can coexist for the same graph.

Basic formulation. The Ising model [20] is a fundamental mathematical model in statistical mechanics used to study the magnetic properties of materials. Each Ising instance is a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with coupling weights J_{ij} and both orientations of each undirected edge included in $\mathcal{E}$. A spin configuration is a vector $\boldsymbol{s} \in \{-1, +1\}^{|\mathcal{V}|}$, representing the binary magnetic dipole (up or down) of each node. Finding the system's stable physical ground state at zero temperature corresponds to solving the minimal energy problem:

$$\min_{\boldsymbol{s} \in \{-1,+1\}^{|\mathcal{V}|}} E_{\text{Ising}}(\boldsymbol{s}, \mathcal{G}) = -\frac{1}{2} \sum_{(i,j) \in \mathcal{E}} J_{ij} s_i s_j. \tag{4}$$

When all couplings are positive ($J_{ij} > 0$), the ferromagnetic ground states are trivial: all spins align, giving the uniform all-$(+1)$ or all-(-1) states. We therefore focus on the antiferromagnetic case

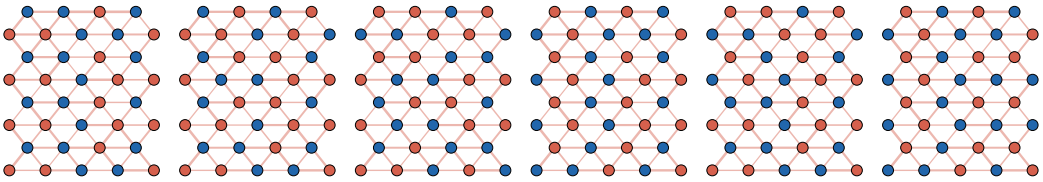

Figure 3: Unsupervised multi-solution discovery. The same Ising graph instance yields distinct spin configurations from different initializations. Red nodes denote $+1$ spins and blue for -1.

($J_{ij} < 0$), where *geometric frustration* [30] leads to many competing low-energy configurations of (4), i.e., the map from a graph to its low-energy states is set-valued.

Model. To parameterize the recurrent update g, we use a standard message-passing graph neural network (GNN) [14] on the weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. At iteration t, the current continuous spin vector $\boldsymbol{s}_t$ is treated as the node feature, and the GNN updates it using the graph structure and edge weights J_{ij}: $\boldsymbol{s}_{t+1} = g_{\mathrm{GNN}}(\boldsymbol{s}_t, \mathcal{G})$. Starting from different initializations $\boldsymbol{s}_0$, the same weight-tied update is unrolled for T steps, and the final continuous prediction is $\hat{\boldsymbol{s}} = \boldsymbol{s}_T$. The goal of training is that, as T grows, the iterates converge to low-energy spin configurations of the same graph; different initializations may therefore lead to different low-energy states.

As a non-recurrent counterpart, we also consider a vanilla GNN baseline. This model applies message passing to the weighted graph as a feed-forward predictor and directly outputs a spin $\hat{\boldsymbol{s}} = h_{\mathrm{GNN}}(\mathcal{G})$.

Loss and evaluation. During neural training, the model outputs continuous spins $\hat{\boldsymbol{s}} \in [-1, 1]^{|\mathcal{V}|}$ through a $\tanh$ readout rather than clipping. To penalize non-binary outputs, we use the following unsupervised training loss:

$$E(\hat{\boldsymbol{s}}; \mathcal{G}) = \frac{1}{|\mathcal{V}|}\Big(-\frac{1}{2} \sum_{(i,j)\in\mathcal{E}} J_{ij}\hat{s}_i\hat{s}_j + \sum_{i\in\mathcal{V}} (\hat{s}_i^2 - 1)^2 \Big). \tag{5}$$

For evaluation, we round the final prediction to $\bar{s}_i \in \{-1, +1\}$ and report the physical energy $E_{\mathrm{Ising}}(\bar{\boldsymbol{s}}, \mathcal{G})/|\mathcal{V}|$. Here, low-energy spin configurations are candidate solutions; #Sols counts distinct rounded outputs, with quality assessed separately by energy. Dataset and metric details are in Appendix E.

Can unsupervised training discover many solutions? Yes. In this Ising setting, the energy-trained weight-tied dynamics shows clear multi-solution behavior. Figure 3 gives a representative visual example, and Table 2 provides a quantitative summary. Across 20 initializations, the energy-trained recurrent GNN finds 13.635 distinct rounded configurations on average modulo global spin flip, with Spread 0.287 and Recall@10% 0.743. Thus, the outputs exhibit both separation and coverage of the sampled low-energy reference pool, rather than merely a large count. A fuller histogram of the solution counts is provided in Appendix E.

Label fitting or energy minimization? We next compare two training objectives: fitting one solver-selected label versus minimizing the physical energy directly. For the label-fitting objective, we use Gurobi [17] to solve the binary quadratic Ising problem in (4), and train the model to imitate one returned spin configuration for each graph. For the energy objective, we train directly with (5). Table 2 shows that energy minimization gives lower rounded-spin energy for both vanilla and recurrent GNNs. The vanilla GNN still outputs a single configuration, but the energy objective lets it choose a low-energy branch instead of imitating an arbitrary Gurobi-selected one. The recurrent GNN has additional freedom: different initializations can follow different trajectories and settle into different low-energy basins. Thus, energy-based training not only avoids a potentially brittle single-branch label, but also enables the recurrent model to exploit its multi-solution dynamics.

Why does energy minimization yield a more robust solver? This gap reflects the regularity viewpoint in Section 2.2. Under energy minimization, the model is not forced to imitate a solver-selected branch; it can dynamically route each input to a low-energy attraction basin. Theorem 2.3 establishes selector regularity for the constructed operator, whereas unconstrained manual selectors can be arbitrarily irregular (Theorem 2.4). Regularity of the trained neural solver is assessed empirically rather than guaranteed by these representation results. Thus, single-branch fitting may introduce artificial branch switches. Table 3 supports this contrast empirically. Energy-based models

Table 2: Ising accuracy, diversity, and reference-pool coverage. Test energy is $E_{\text{Ising}}(\bar{s}, \mathcal{G})/|\mathcal{V}|$ for rounded predictions. Diversity metrics use 200 test graphs and 20 outputs per graph, with random starts for the energy-trained recurrent GNN and repeated deterministic outputs for the single-output baselines. #Sols counts distinct configurations modulo global sign; Spread is mean pairwise symmetry-adjusted Hamming distance; Recall@10% measures reference-pool coverage at distance < 0.1. Entries report mean $\pm$ population standard deviation across graphs.

Objective	Model	Test energy ↓	#Sols ↑	Spread ↑	Recall@10% ↑
Gurobi label	Vanilla GNN	-0.059 ± 0.358	1.000 ± 0.000	0.000 ± 0.000	0.016 ± 0.122
Gurobi label	Recurrent GNN	-0.092 ± 0.361	1.000 ± 0.000	0.000 ± 0.000	0.041 ± 0.196
Energy-based	Vanilla GNN	-0.750 ± 0.190	1.000 ± 0.000	0.000 ± 0.000	0.074 ± 0.248
Energy-based	Recurrent GNN	$\mathbf{-0.983 \pm 0.101}$	13.635 ± 5.760	0.287 ± 0.071	0.743 ± 0.246

Table 3: Empirical perturbation sensitivity at noise level $\varepsilon = 0.005$. Values represent percentiles of the normalized output variation, $\|\hat{s}(J + \Delta J) - \hat{s}(J)\|_2 / \|\Delta J\|_2$, across the test set. To illustrate, $p_{40} = 15.184$ indicates that 40% of the tested perturbations resulted in an output variation of 15.184 or lower. Energy-based models remain highly stable for the vast majority of inputs, whereas models trained on Gurobi labels exhibit sensitivity globally, which supports our theory in Section 2.2.

Obj.	Model	p_{40}	p_{50}	p_{60}	p_{70}	p_{80}	p_{90}	p_{95}
Gurobi	Vanilla GNN	15.184	16.730	18.411	20.693	23.767	27.601	31.606
Gurobi	Recurrent GNN	9.919	10.762	11.600	12.591	14.020	15.612	17.523
Energy	Vanilla GNN	0.000	0.000	0.000	0.000	0.000	0.000	25.978
Energy	Recurrent GNN	0.000	0.000	0.000	0.000	0.000	0.025	38.168

have near-zero perturbation sensitivity for most inputs, while Gurobi-label models remain sensitive across the distribution. This suggests that energy minimization avoids fitting an unnecessarily irregular selector.

Larger graphs. Appendix E extends the evaluation to graphs with up to $\sim$15,000 nodes. Appendix E.5–E.8 provide initialization, multi-solution baseline, perturbation, and solver comparisons.

4 Encouraging Bifurcations: Explicit Diversity in Training

The Ising experiment shows a favorable regime where unsupervised weight-tied dynamics naturally uses different initializations to discover multiple low-energy solutions. We now ask a complementary question: *is such diversity automatic once the model has multi-equilibrium capacity?* To answer this question, we use the Allen–Cahn equation as a testbed.

Problem formulation. Allen–Cahn equation [2] is a fundamental reaction-diffusion equation widely used to study complex pattern formation and phase transitions in continuous media. On the domain $\Omega = (0, 2\pi)^2$, with phase field $u : \Omega \to \mathbb{R}$, forcing $f : \Omega \to \mathbb{R}$, and $\varepsilon > 0$, we consider

$$-\varepsilon^2 \Delta u + u^3 - u - f = 0 \qquad \text{in } \Omega,$$

with periodic boundary conditions, i.e., on the flat torus $\mathbb{T}^2$. It is the Euler–Lagrange equation of

$$E_{\text{AC}}(u; f) = \frac{1}{|\Omega|} \int_\Omega \left[\frac{\varepsilon^2}{2} \|\nabla u(x)\|_2^2 + \frac{1}{4} \big(u(x)^2 - 1\big)^2 - f(x) u(x) \right] \mathrm{d}x.$$

The energy balances interface smoothing, external forcing, and a double-well potential that drives the system toward distinct macroscopic phases ($u \approx \pm 1$). This competition naturally creates multiple stable steady states for the exact same forcing f [3]. This continuous, multi-basin landscape provides an ideal testbed, requiring the network to genuinely capture coexisting attraction basins.

Model. Given a forcing field f, our goal is to learn a solver that maps f to one or more steady phase fields u. Since both the input and output are periodic functions on Ω, we parameterize the update map with a Fourier neural operator (FNO) [25]. We do not use the FNO as a one-shot predictor $f \mapsto u$. Instead, following the implicit neural operator viewpoint of [28], we use it as a shared recurrent update: $u_{t+1} = g_{\text{FNO}}(u_t, f)$. Starting from a randomly generated initial state u_0, we unroll the

weight-tied dynamics for T iterations. During training, the model parameters are optimized entirely unsupervised by directly minimizing the terminal physical energy $E_{\mathrm{AC}}(u_T; f)$.

Evaluation metrics. To evaluate physical accuracy, we measure the steady-state PDE residual using the discrete mean squared error (MSE) over an $N \times N$ spatial grid:

$$\mathrm{MSE}_{\mathrm{res}}(u; f) = \frac{1}{N^2} \sum_{p=1}^{N} \sum_{q=1}^{N} \left(-\varepsilon^2 \Delta u + u^3 - u - f\right)_{p,q}^2 .$$

Does multi-equilibrium capacity imply automatic branch discovery? Not necessarily. Figure 4 shows that the energy-only model maps very different initial states to visually similar final states, indicating collapse to a dominant branch. Table 4 confirms this quantitatively: the energy-only model produces one output cluster under the calibrated counting threshold. Thus, while the weight-tied dynamics has the capacity to represent multiple attractors, minimizing the energy alone does not force those attractors to be used.

How does energy training compare with IMEX-label fitting? Although the energy-only model does not automatically bifurcate, it is still much more accurate than fitting a single IMEX-generated label; see Table 4. For the single-branch target, we use the classical periodic IMEX solver [6] to construct one target branch for each forcing field. Forcing the network to fit this selected branch gives substantially worse physical residuals and energy than the energy-only model. By contrast, energy training does not prescribe which branch the network must imitate; it only asks the dynamics to reach a low-energy steady state. The learned model can therefore settle on the branch that is easiest to represent and optimize, rather than spending capacity matching a particular label-selection rule.

If we still want multiple solutions, how do we achieve that? We add an explicit diversity-promoting term to the energy objective. For a fixed forcing field f, we sample M independent initial states $u_k^{(0)} \sim \rho$, evolve each of them $k = 1, \ldots, M$ with the same weight-tied dynamics, $u_k^{(t+1)} = g_\theta(u_k^{(t)}, f)$. Denote the final states by $\hat{u}_k := u_k^{(T)}$. The diversity-regularized objective is

$$\mathcal{L}_\lambda(\theta; f) = \frac{1}{M} \sum_{k=1}^{M} E_{\mathrm{AC}}(\hat{u}_k; f) - \lambda \, \mathcal{D}_{\mathrm{train}}\left(\{\hat{u}_k\}_{k=1}^{M}\right),$$

where the training regularizer uses mean pairwise squared distances:

$$\mathcal{D}_{\mathrm{train}}\left(\{\hat{u}_k\}_{k=1}^{M}\right) = \frac{1}{M(M-1)} \sum_{i \neq j} \frac{1}{N^2} \|\hat{u}_i - \hat{u}_j\|_2^2.$$

Here, the penalty weight λ is decayed over the progress of training. For clarity, we refer to each specific decay schedule by its initial penalty weight, λ_{init} (full details in Appendix F).

Figure 4 illustrates the effect. The top row shows five independent initial states for the same forcing field. With energy-only training, $\lambda_{\mathrm{init}} = 0$, the middle row shows that these trajectories collapse to visually similar final states. With diversity regularization, $\lambda_{\mathrm{init}} = 1.2$, the bottom row shows that the same initial states lead to visibly distinct steady-state candidates. Table 4 confirms the same trend: the average number of detected output clusters increases from 1.00 at $\lambda_{\mathrm{init}} = 0$ to 16.11 at $\lambda_{\mathrm{init}} = 1.2$. This increased separation persists after IMEX refinement: refined diversity rises from 0.420 to 1.304.

Is there an accuracy–diversity tradeoff? Yes. Table 4 shows a clear tradeoff. Among the learned models, the energy-only model gives the lowest residual MSE and energy, but it discovers only one output cluster. Diversity regularization can produce more distinct outputs: for example, $\lambda_{\mathrm{init}} = 1.2$ increases the average number of output clusters from 1.00 to 16.11. However, this comes at a cost: the test residual MSE increases from 9.19×10^{-4} to 5.74×10^{-3}, and the test energy becomes less favorable. The dependence on the diversity weight is not monotone: the mean cluster count reaches 19.51 at $\lambda_{\mathrm{init}} = 2.5$, then decreases to 12.18 at 3.0. Thus, in this experiment, multi-solution discovery is not free. This tradeoff gives a practical way to tune the solver: setting $\lambda = 0$ prioritizes physical accuracy, while increasing λ encourages broader exploration of the solution landscape.

Additional experiments. Appendix F reports solver refinement, resolution scaling, initialization sensitivity, and matched training-rule comparisons, including computational cost and quantitative convergence statistics.

Codes. Our codes are provided here: `https://github.com/liujl11git/bifurcation_models`

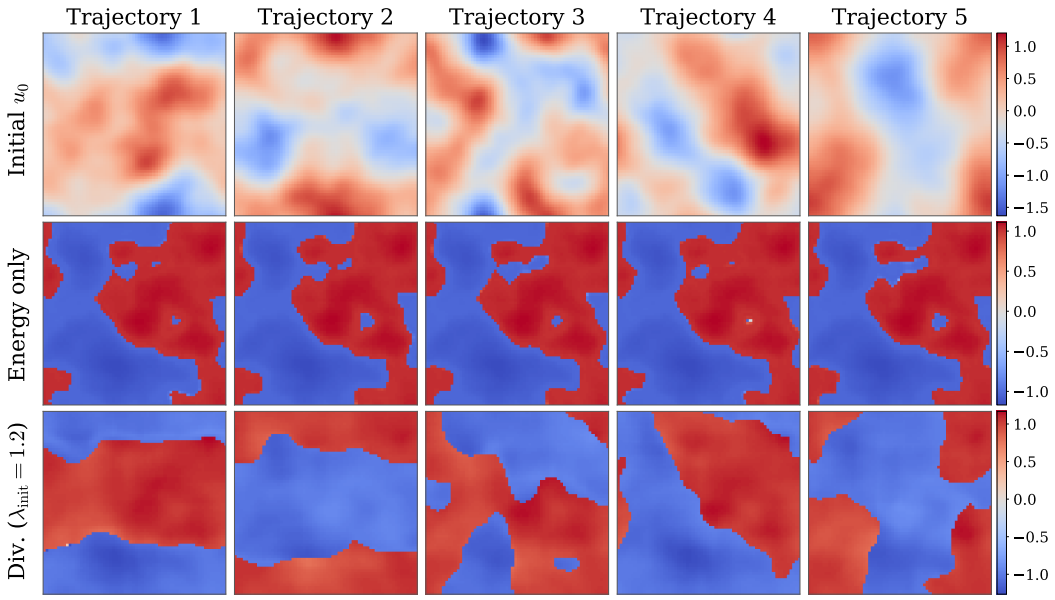

Figure 4: Five trajectories for the same f. **Top row:** five independent initial states. **Middle row:** final predictions from the energy-only model: collapse to visually similar states. **Bottom row:** final predictions from the diversity-regularized model: discover distinct steady-state candidates.

Table 4: Allen–Cahn test evaluation at $\varepsilon = 0.01$ on a 64×64 grid, using 20 GRF-initialized trajectories and 50 iterations per forcing instance. "# Sols" is the average number of distinct output clusters among the final states. Diversity and refined diversity measure mean pairwise RMSE before and after high-accuracy IMEX refinement. Entries report mean $\pm$ standard deviation; metric-specific aggregation is described in Appendix F.

Model / objective	Test residual MSE ↓ ($\times 10^{-3}$)	Test energy ↓ ($\times 10^{-2}$)	#Sols ↑	Diversity ↑	Refined diversity ↑
IMEX label	95.470 ± 17.310	14.270 ± 0.858	1.00 ± 0.00	$(1.065 \pm 0.248) \times 10^{-8}$	$(1.570 \pm 11.630) \times 10^{-11}$
Energy only	0.919 ± 0.036	$\mathbf{-11.480 \pm 0.020}$	1.00 ± 0.00	0.429 ± 0.066	0.420 ± 0.066
Div. $\lambda_{\text{init}} = 0.5$	1.552 ± 0.176	-11.440 ± 0.027	1.00 ± 0.00	0.426 ± 0.062	0.440 ± 0.066
Div. $\lambda_{\text{init}} = 0.6$	1.570 ± 0.165	-11.440 ± 0.027	1.00 ± 0.00	0.429 ± 0.062	0.442 ± 0.066
Div. $\lambda_{\text{init}} = 0.7$	1.785 ± 0.256	-11.390 ± 0.039	1.00 ± 0.07	0.475 ± 0.072	0.497 ± 0.078
Div. $\lambda_{\text{init}} = 0.8$	2.140 ± 0.321	-11.360 ± 0.053	1.02 ± 0.16	0.460 ± 0.082	0.491 ± 0.087
Div. $\lambda_{\text{init}} = 0.9$	2.671 ± 0.223	-11.420 ± 0.026	1.00 ± 0.00	0.420 ± 0.061	0.437 ± 0.066
Div. $\lambda_{\text{init}} = 1.0$	4.351 ± 0.342	-11.140 ± 0.096	1.04 ± 0.21	0.425 ± 0.090	0.456 ± 0.095
Div. $\lambda_{\text{init}} = 1.1$	5.135 ± 0.696	-9.929 ± 0.493	4.09 ± 2.72	0.790 ± 0.135	0.812 ± 0.139
Div. $\lambda_{\text{init}} = 1.2$	5.742 ± 2.076	-4.220 ± 2.959	16.11 ± 1.87	1.346 ± 0.047	1.304 ± 0.053
Div. $\lambda_{\text{init}} = 2.0$	4.071 ± 0.738	-0.795 ± 3.387	18.84 ± 1.04	1.368 ± 0.018	1.355 ± 0.031
Div. $\lambda_{\text{init}} = 2.5$	1.404 ± 0.224	1.303 ± 3.183	19.51 ± 0.68	1.371 ± 0.019	1.358 ± 0.031
Div. $\lambda_{\text{init}} = 3.0$	10.410 ± 3.204	-7.173 ± 1.656	12.18 ± 2.79	1.055 ± 0.073	1.069 ± 0.082
Pure IMEX	$\mathbf{0.158 \pm 0.157}$	-3.624 ± 2.784	19.61 ± 0.66	1.350 ± 0.033	1.346 ± 0.037

5 Conclusions

We studied deterministic set-valued solver learning for scientific and optimization instances with independently verifiable solutions. Bifurcation models represent their multiplicity through weight-tied dynamics with initialization-dependent equilibria. Under the stated assumptions, broad finite-branch maps admit regular dynamics and almost-everywhere regular induced selectors, whereas unconstrained branch selection can introduce irregularity. Empirically, these dynamics discover low-energy Ising solutions without complete branch information. Allen–Cahn training instead reveals an accuracy–diversity tradeoff: explicit diversity promotion broadens exploration, making branch discovery a tunable objective rather than an automatic consequence of multi-equilibrium capacity.

Acknowledgements and Disclosures

This work received no third-party funding or support. The authors declare no competing interests.

References

[1] Jonas Adler and Ozan Öktem. Learned primal-dual reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1322–1332, 2018.

[2] Samuel M Allen and John W Cahn. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta metallurgica*, 27(6):1085–1095, 1979.

[3] João Henrique Andrade, Dario Corona, Stefano Nardulli, Paolo Piccione, and Raoní Ponciano. From bubbles to clusters: Multiple solutions to the allen–cahn system. *Journal of Differential Equations*, 464:114189, 2026.

[4] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. Deep equilibrium models. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. *Advances in Neural Information Processing Systems*, 35:20232–20242, 2022.

[6] Long Qing Chen and Jie Shen. Applications of semi-implicit fourier-spectral method to phase field equations. *Computer Physics Communications*, 108(2-3):147–158, 1998.

[7] Laurent El Ghaoui, Fangda Gu, Bertrand Travacca, Armin Askari, and Alicia Tsai. Implicit deep learning. *SIAM Journal on Mathematics of Data Science*, 3(3):930–958, 2021.

[8] Lawrence C. Evans and Ronald F. Gariepy. *Measure Theory and Fine Properties of Functions*. CRC Press, revised edition, 2015.

[9] Patrick E. Farrell, Ásgeir Birkisson, and Simon W. Funke. Deflation techniques for finding distinct solutions of nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 37(4):A2026–A2045, 2015. doi: 10.1137/140984798.

[10] Samy Wu Fung, Howard Heaton, Qiuwei Li, Daniel McKenzie, Stanley Osher, and Wotao Yin. JFB: Jacobian-free backpropagation for implicit networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[11] Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

[12] Zhengyang Geng, Xin-Yu Zhang, Shaojie Bai, Yisen Wang, and Zhouchen Lin. On training implicit models. *Advances in Neural Information Processing Systems*, 34:24247–24260, 2021.

[13] Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, Jason D. Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 11398–11442. PMLR, 2023.

[14] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. Pmlr, 2017.

[15] Davis Gilton, Gregory Ongie, and Rebecca Willett. Deep equilibrium architectures for inverse problems in imaging. *IEEE Transactions on Computational Imaging*, 7:1123–1133, 2021.

[16] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, pages 399–406, 2010.

[17] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2024. URL https://www.gurobi.com.

[18] Aaron Havens, Alexandre Araujo, Siddharth Garg, Farshad Khorrami, and Bin Hu. Exploiting connections between Lipschitz structures for certifiably robust deep equilibrium models. *Advances in Neural Information Processing Systems*, 36:21658–21674, 2023.

[19] Yao Huang, Wenrui Hao, and Guang Lin. HomPINNs: Homotopy physics-informed neural networks for learning multiple solutions of nonlinear elliptic differential equations. *Computers & Mathematics with Applications*, 121:62–73, 2022. doi: 10.1016/j.camwa.2022.07.002.

[20] Ernst Ising. Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, 1925. doi: 10.1007/BF02980577.

[21] Saber Jafarpour, Alexander Davydov, Anton Proskurnikov, and Francesco Bullo. Robust implicit networks via non-Euclidean contractions. *Advances in Neural Information Processing Systems*, 34:9857–9868, 2021.

[22] K. Kim, M.-S. Chang, S. Korenblit, R. Islam, E. E. Edwards, J. K. Freericks, G.-D. Lin, L.-M. Duan, and C. Monroe. Quantum simulation of frustrated ising spins with trapped ions. *Nature*, 465(7298):590–593, 2010. doi: 10.1038/nature09071.

[23] Hannah Lawrence, Vasco Portilheiro, Yan Zhang, and Sékou-Oumar Kaba. Improving equivariant networks with probabilistic symmetry breaking. In *ICML 2024 Workshop on Geometry-grounded Representation Learning and Generative Modeling*, 2024. URL https://openreview.net/forum?id=1VlRaXNMWO.

[24] Lingxiao Li, Noam Aigerman, Vladimir Kim, Jiajin Li, Kristjan Greenewald, Mikhail Yurochkin, and Justin Solomon. Learning proximal operators to discover multiple optima. In *The Eleventh International Conference on Learning Representations*, 2023.

[25] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021.

[26] Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations (ICLR)*, 2019.

[27] Jialin Liu, Lisang Ding, Stanley Osher, and Wotao Yin. Expressive power of implicit models: Rich equilibria and test-time scaling. In *The Fourteenth International Conference on Learning Representations*, 2026.

[28] Tanya Marwah, Ashwini Pokle, J Zico Kolter, Zachary Lipton, Jianfeng Lu, and Andrej Risteski. Deep equilibrium based neural operators for steady-state PDEs. *Advances in Neural Information Processing Systems*, 36:15716–15737, 2023.

[29] Christopher A Metzler, Ali Mousavi, and Richard G Baraniuk. Learned D-AMP: Principled neural network based compressive image recovery. *Advances in Neural Information Processing Systems*, pages 1773–1784, 2017.

[30] Roderich Moessner and Arthur P Ramirez. Geometrical frustration. *Physics Today*, 59(2): 24–29, 2006.

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

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[33] Max Revay, Ruigang Wang, and Ian R Manchester. Lipschitz bounded equilibrium networks. *arXiv preprint arXiv:2010.01732*, 2020.

[34] Matus Telgarsky. Neural networks and rational functions. In *International Conference on Machine Learning*, pages 3387–3393. PMLR, 2017.

[35] Hannes Uecker. Continuation and bifurcation in nonlinear pdes – algorithms, applications, and experiments. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 124(1):43–80, 2022. doi: 10.1365/s13291-021-00241-5.

[36] Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*, 2025.

[37] Ezra Winston and J Zico Kolter. Monotone operator equilibrium networks. *Advances in Neural Information Processing Systems*, 33:10718–10728, 2020.

[38] Bo Xin, Yizhou Wang, Wen Gao, David Wipf, and Baoyuan Wang. Maximal sparsity with deep networks? *Advances in Neural Information Processing Systems*, 29:4340–4348, 2016.

[39] Yan Yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for compressive sensing MRI. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 10–18, 2016.

[40] Hao You, Liuge Du, Xiao Xu, and Jia Zhao. Deep learning methods for solving non-uniqueness of inverse design in photonics. *Optics Communications*, 554:130122, 2024.

[41] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

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

[43] Zongren Zou, Zhicheng Wang, and George Em Karniadakis. Learning and discovering multiple solutions using physics-informed neural networks with random initialization and deep ensemble. *Proceedings of the Royal Society A*, 481(2325):20250205, 2025. doi: 10.1098/rspa.2025.0205.

A Proof of Theorem 2.2

Throughout this section, let $\mathcal{K} \subset \mathbb{R}^d$ be bounded, let $\mathcal{Y} = \mathbb{R}^m$, and let

$$f_1, \dots, f_n : \mathcal{K} \to \mathcal{Y}$$

be locally Lipschitz branch maps. These branches may collapse. Define the finite set of distinct branch values

$$\mathcal{F}(x) := \{f_1(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})\} \subset \mathcal{Y}.$$

A.1 Nearest solution selector

Definition A.1 (Priority nearest selector)**.** Fix a priority order

$$1 \prec 2 \prec \cdots \prec n.$$

For $\boldsymbol{y} \in \mathcal{Y}$ and $\boldsymbol{x} \in \mathcal{K}$, define

$$\pi_{\boldsymbol{y}}(\boldsymbol{x}) := \min \operatorname*{argmin}_{1 \le i \le n} \|\boldsymbol{y} - f_i(\boldsymbol{x})\|,$$

where the minimum is taken according to the fixed priority order. Define

$$P_{\boldsymbol{y}}(\boldsymbol{x}) := f_{\pi_{\boldsymbol{y}}(\boldsymbol{x})}(\boldsymbol{x}).$$

Thus $P_{\boldsymbol{y}}(\boldsymbol{x})$ is single-valued for every $(\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{Y} \times \mathcal{K}$.

Definition A.2 (Voronoi switching set)**.** Define the reduced switching set

$$\Sigma^\circ(\boldsymbol{x}) := \Big\{ \boldsymbol{y} \in \mathcal{Y} : \exists i \neq j,\ f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x}),\ \|\boldsymbol{y} - f_i(\boldsymbol{x})\| = \|\boldsymbol{y} - f_j(\boldsymbol{x})\| = \min_k \|\boldsymbol{y} - f_k(\boldsymbol{x})\| \Big\}.$$

Define the graph-closure switching set

$$\widetilde{\Sigma}(\boldsymbol{x}_0) := \{ \boldsymbol{y} \in \mathcal{Y} : \exists \boldsymbol{x}_\ell \to \boldsymbol{x}_0,\ \exists \boldsymbol{y}_\ell \to \boldsymbol{y} \text{ such that } \boldsymbol{y}_\ell \in \Sigma^\circ(\boldsymbol{x}_\ell) \}.$$

Equivalently,

$$\operatorname{graph}(\widetilde{\Sigma}) = \overline{\operatorname{graph}(\Sigma^\circ)} \subset \mathcal{K} \times \mathcal{Y}.$$

In particular, $\operatorname{graph}(\widetilde{\Sigma})$ is closed.

Lemma A.3 (Continuity of solution selector)**.** *Let $K \subset \mathbb{R}^d$ and let $Y = \mathbb{R}^m$ with the Euclidean norm. Let $f_1, \dots, f_n : \mathcal{K} \to \mathcal{Y}$ be locally Lipschitz. For fixed $\boldsymbol{y} \in \mathcal{Y}$, define*

$$\mathcal{J}_{\boldsymbol{y}} := \{ \boldsymbol{x} \in \mathcal{K} : (\boldsymbol{x}, \boldsymbol{y}) \in \operatorname{graph}(\widetilde{\Sigma}) \}.$$

Then $P_{\boldsymbol{y}}$ is locally Lipschitz on $\mathcal{K} \setminus \mathcal{J}_{\boldsymbol{y}}$.

Proof. Fix $\boldsymbol{x}_0 \in \mathcal{K} \setminus \mathcal{J}_{\boldsymbol{y}}$. Since

$$(\boldsymbol{x}_0, \boldsymbol{y}) \notin \operatorname{graph}(\widetilde{\Sigma})$$

and $\operatorname{graph}(\widetilde{\Sigma})$ is closed, there exist a relative neighborhood

$$\mathbb{V} \subset \mathcal{K}$$

of $\boldsymbol{x}_0$, and a number $\varepsilon > 0$, such that

$$(\mathbb{V} \times \mathbb{B}_\varepsilon(\boldsymbol{y})) \cap \operatorname{graph}(\Sigma^\circ) = \varnothing.$$

Equivalently, for every $\boldsymbol{x} \in \mathbb{V}$ and every $\boldsymbol{q} \in \mathbb{B}_\varepsilon(\boldsymbol{y})$, the point $\boldsymbol{q}$ is not on the reduced Voronoi switching set of $\mathcal{F}(\boldsymbol{x})$.

For fixed $\boldsymbol{x} \in \mathbb{V}$, the ball $\mathbb{B}_\varepsilon(\boldsymbol{y})$ is connected. Since no point in this ball crosses a reduced switching boundary, the same distinct branch value remains the unique nearest value throughout the ball. Therefore, if

$$\boldsymbol{u} = P_{\boldsymbol{y}}(\boldsymbol{x}),$$

then for every $\boldsymbol{q} \in \mathbb{B}_\varepsilon(\boldsymbol{y})$ and every $\boldsymbol{w} \in \mathcal{F}(\boldsymbol{x})$,

$$\|\boldsymbol{q} - \boldsymbol{u}\| \le \|\boldsymbol{q} - \boldsymbol{w}\|.$$

Moreover, if $\boldsymbol{w} \neq \boldsymbol{u}$, choose

$$\boldsymbol{q} = \boldsymbol{y} + \frac{\varepsilon}{2} \frac{\boldsymbol{w} - \boldsymbol{u}}{\|\boldsymbol{w} - \boldsymbol{u}\|}.$$

and plug it into $\|\boldsymbol{q} - \boldsymbol{u}\|^2 \leq \|\boldsymbol{q} - \boldsymbol{w}\|^2$ gives

$$\left\| \boldsymbol{y} - \boldsymbol{u} + \frac{\varepsilon}{2} \frac{\boldsymbol{w} - \boldsymbol{u}}{\|\boldsymbol{w} - \boldsymbol{u}\|} \right\|^2 \leq \left\| \boldsymbol{y} - \boldsymbol{w} + \frac{\varepsilon}{2} \frac{\boldsymbol{w} - \boldsymbol{u}}{\|\boldsymbol{w} - \boldsymbol{u}\|} \right\|^2$$

$$\|\boldsymbol{y} - \boldsymbol{u}\|^2 + \varepsilon \left\langle \boldsymbol{y} - \boldsymbol{u}, \frac{\boldsymbol{w} - \boldsymbol{u}}{\|\boldsymbol{w} - \boldsymbol{u}\|} \right\rangle + \frac{\varepsilon^2}{4} \leq \|\boldsymbol{y} - \boldsymbol{w}\|^2 + \varepsilon \left\langle \boldsymbol{y} - \boldsymbol{w}, \frac{\boldsymbol{w} - \boldsymbol{u}}{\|\boldsymbol{w} - \boldsymbol{u}\|} \right\rangle + \frac{\varepsilon^2}{4}$$

and hence by possibly shrinking ε, we have

$$\|\boldsymbol{y} - \boldsymbol{w}\|^2 - \|\boldsymbol{y} - \boldsymbol{u}\|^2 \geq \varepsilon \|\boldsymbol{w} - \boldsymbol{u}\|$$

for all $\boldsymbol{x} \in \mathbb{V}$, $\boldsymbol{u} = P_{\boldsymbol{y}}(\boldsymbol{x})$, and $\boldsymbol{w} \in \mathcal{F}(\boldsymbol{x})$.

Now shrink $\mathbb{V}$ if necessary so that all f_i are L-Lipschitz on $\mathbb{V}$. Also, let

$$B := \sup_{\boldsymbol{x} \in \mathbb{V},\, 1 \leq i \leq n} \|\boldsymbol{y} - f_i(\boldsymbol{x})\| < \infty.$$

Take arbitrary $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{V}$. Let

$$\boldsymbol{u} = P_{\boldsymbol{y}}(\boldsymbol{x}), \qquad \boldsymbol{v} = P_{\boldsymbol{y}}(\boldsymbol{z}).$$

Choose labels i, j such that

$$\boldsymbol{u} = f_i(\boldsymbol{x}), \qquad \boldsymbol{v} = f_j(\boldsymbol{z}).$$

Set

$$\boldsymbol{r} := f_i(\boldsymbol{z}), \qquad \boldsymbol{w} := f_j(\boldsymbol{x}).$$

By L-Lipschitzness,

$$\|\boldsymbol{u} - \boldsymbol{r}\| \leq L\|\boldsymbol{x} - \boldsymbol{z}\|, \qquad \|\boldsymbol{v} - \boldsymbol{w}\| \leq L\|\boldsymbol{x} - \boldsymbol{z}\|.$$

Apply the margin estimate at $\boldsymbol{x}$ with competitor $\boldsymbol{w}$, and at $\boldsymbol{z}$ with competitor $\boldsymbol{r}$:

$$\|\boldsymbol{y} - \boldsymbol{w}\|^2 - \|\boldsymbol{y} - \boldsymbol{u}\|^2 \geq \varepsilon \|\boldsymbol{w} - \boldsymbol{u}\|,$$
$$\|\boldsymbol{y} - \boldsymbol{r}\|^2 - \|\boldsymbol{y} - \boldsymbol{v}\|^2 \geq \varepsilon \|\boldsymbol{r} - \boldsymbol{v}\|.$$

Adding,

$$\varepsilon(\|\boldsymbol{w} - \boldsymbol{u}\| + \|\boldsymbol{r} - \boldsymbol{v}\|) \leq \big(\|\boldsymbol{y} - \boldsymbol{w}\|^2 - \|\boldsymbol{y} - \boldsymbol{v}\|^2\big) + \big(\|\boldsymbol{y} - \boldsymbol{r}\|^2 - \|\boldsymbol{y} - \boldsymbol{u}\|^2\big).$$

The right-hand side is bounded by $4BL\|\boldsymbol{x} - \boldsymbol{z}\|$ because

$$\begin{aligned} \|\boldsymbol{y} - \boldsymbol{w}\|^2 - \|\boldsymbol{y} - \boldsymbol{v}\|^2 &= \langle (\boldsymbol{y} - \boldsymbol{w}) + (\boldsymbol{y} - \boldsymbol{v}), \boldsymbol{w} - \boldsymbol{v} \rangle \\ &\leq \Big(\|\boldsymbol{y} - \boldsymbol{w}\| + \|\boldsymbol{y} - \boldsymbol{v}\| \Big) \|\boldsymbol{w} - \boldsymbol{v}\| \leq 2B\|\boldsymbol{w} - \boldsymbol{v}\| \leq 2BL\|\boldsymbol{x} - \boldsymbol{z}\|, \end{aligned}$$

and similarly for $\big(\|\boldsymbol{y} - \boldsymbol{r}\|^2 - \|\boldsymbol{y} - \boldsymbol{u}\|^2\big)$.

On the other hand,

$$\|\boldsymbol{w} - \boldsymbol{u}\| \geq \|\boldsymbol{u} - \boldsymbol{v}\| - L\|\boldsymbol{x} - \boldsymbol{z}\|,$$

and

$$\|\boldsymbol{r} - \boldsymbol{v}\| \geq \|\boldsymbol{u} - \boldsymbol{v}\| - L\|\boldsymbol{x} - \boldsymbol{z}\|.$$

Hence

$$\varepsilon(2\|\boldsymbol{u} - \boldsymbol{v}\| - 2L\|\boldsymbol{x} - \boldsymbol{z}\|) \leq 4BL\|\boldsymbol{x} - \boldsymbol{z}\|.$$

Therefore

$$\|\boldsymbol{u} - \boldsymbol{v}\| \leq L\left(1 + \frac{2B}{\varepsilon}\right) \|\boldsymbol{x} - \boldsymbol{z}\|.$$

Since $\boldsymbol{u} = P_{\boldsymbol{y}}(\boldsymbol{x})$ and $\boldsymbol{v} = P_{\boldsymbol{y}}(\boldsymbol{z})$, this proves the local Lipschitz of $P_{\boldsymbol{y}}$ at $\boldsymbol{x}_0$. Thus $P_{\boldsymbol{y}}$ is locally Lipschitz on $\mathcal{K} \setminus \mathcal{J}_{\boldsymbol{y}}$, which concludes the proof. □

A.2 Bad sets

Here we restate Definition 2.1 and provide a more detailed version below.

Definition A.4 (Stable and unstable collapse pattern)**.** Let $f_1, \dots, f_n : \mathcal{K} \to \mathcal{Y}$ be branch maps, and fix $\boldsymbol{x}_0 \in \mathcal{K}$. We say that the labeled branches have a stable collapse pattern near $\boldsymbol{x}_0$ if there exists a relative neighborhood $\mathbb{U} \subset \mathcal{K}$ of $\boldsymbol{x}_0$ and a partition

$$\{1, \dots, n\} = C_1 \cup \cdots \cup C_q$$

such that, for every $\boldsymbol{x} \in \mathbb{U}$,

$$f_i(\boldsymbol{x}) = f_j(\boldsymbol{x}) \quad \Longleftrightarrow \quad i, j \in C_\alpha \text{ for some } \alpha.$$

Thus labels in the same cluster remain collapsed throughout $\mathbb{U}$, while labels in different clusters remain distinct throughout $\mathbb{U}$. Let

$$\mathcal{S} := \{\boldsymbol{x} \in \mathcal{K} : \text{the branches have a stable collapse pattern near } \boldsymbol{x}\}.$$

Define the unstable collapse set

$$\mathcal{U} := \mathcal{K} \setminus \mathcal{S}.$$

Lemma A.5 (Closedness of the unstable collapse set)**.** *The set $\mathcal{U}$ is relatively closed in $\mathcal{K}$.*

Proof. The stable set $\mathcal{S}$ is relatively open. Indeed, if a stable collapse partition holds on a relative neighborhood $\mathbb{U}$ of $\boldsymbol{x}_0$, then the same partition holds on a smaller relative neighborhood of every point in $\mathbb{U}$. Therefore $\mathcal{U} = \mathcal{K} \setminus \mathcal{S}$ is relatively closed. □

Definition A.6 (Switching set.)**.** For each fixed $\boldsymbol{y} \in \mathcal{Y}$, define the switching-in-$\boldsymbol{x}$ set

$$\mathcal{J}_{\boldsymbol{y}} := \{\boldsymbol{x} \in \mathcal{K} : \boldsymbol{y} \in \widetilde{\Sigma}(\boldsymbol{x})\}.$$

Lemma A.7 (Closedness of switching set)**.** *For each $\boldsymbol{y} \in \mathcal{Y}$, the set $\mathcal{J}_{\boldsymbol{y}}$ is relatively closed in $\mathcal{K}$.*

Proof. Since $\mathrm{graph}(\widetilde{\Sigma})$ is closed in $\mathcal{K} \times \mathcal{Y}$, its section $\mathcal{J}_{\boldsymbol{y}}$ is relatively closed in $\mathcal{K}$. □

Definition A.8 (Extended domain and discontinuity points)**.** First we extend $\mathcal{F}$ to $\bar{\boldsymbol{x}} \in \overline{\mathcal{K}} \backslash \mathcal{K}$ by the limit relative to $\mathcal{K}$. For each branch $i = 1, 2, \cdots, n$, we define

$$f_i(\bar{\boldsymbol{x}}) = \begin{cases} \lim\limits_{\mathcal{K} \ni \boldsymbol{x} \to \bar{\boldsymbol{x}}} f_i(\boldsymbol{x}), & \text{if } \lim\limits_{\mathcal{K} \ni \boldsymbol{x} \to \bar{\boldsymbol{x}}} f_i(\boldsymbol{x}) \text{ exists,} \\ 0, & \text{otherwise.} \end{cases}$$

In addition, the nearest branch selector $P_{\boldsymbol{y}}(\boldsymbol{x})$ can also be extended to $\overline{\mathcal{K}}$ accordingly.

Note that even if f_i is continuously extendable to $\bar{\boldsymbol{x}}$, it is still possible that f_i is not locally Lipschitz continuous at the point $\bar{\boldsymbol{x}}$. An example is $\sqrt{x}$ when it is extended from $(0, 1]$ to 0. We collect all these points (where f_i is not locally Lipschitz) into the set $\mathcal{D}$:

$$\mathcal{D} := \Big\{\boldsymbol{x} \in \overline{\mathcal{K}} : f_i \text{ is not locally Lipschitz continuous at } \boldsymbol{x} \text{ for some } i \in \{1, 2, \cdots, n\}.\Big\}$$

Lemma A.9 (Closedness of discontinuity points)**.** *$\mathcal{D}$ is closed in $\overline{\mathcal{K}}$ and $\mathcal{D} \subset \overline{\mathcal{K}} \backslash \mathcal{K}$.*

Proof. The conclusion $\mathcal{D} \subset \overline{\mathcal{K}} \backslash \mathcal{K}$ directly follows from the definition of $\mathcal{D}$ and the local Lipschitz property of f_i. To show its closedness, we define:

$$\mathcal{D}_i := \Big\{\boldsymbol{x} \in \overline{\mathcal{K}} : f_i \text{ is not locally Lipschitz continuous at } \boldsymbol{x}.\Big\}$$

and hence $\mathcal{D} = \cup_{i=1}^n \mathcal{D}_i$. Thanks to [27, Lemma A.2], each $\mathcal{D}_i$ is closed. The union of finitely many closed sets is closed, then $\mathcal{D}$ is closed. □

Definition A.10 (Bad set)**.** Equip $\mathcal{Y} \times \overline{\mathcal{K}}$ with the product Euclidean norm. Define $\overline{\mathcal{U}}^{\overline{\mathcal{K}}}$ as the closure of $\mathcal{U}$ relative to $\overline{\mathcal{K}}$. The joint bad set is

$$\mathcal{B} := \mathcal{Y} \times \left(\mathcal{D} \cup \overline{\mathcal{U}}^{\overline{\mathcal{K}}}\right) \cup \Big\{(\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{Y} \times \overline{\mathcal{K}} : \boldsymbol{y} \in \widetilde{\Sigma}(\boldsymbol{x})\Big\}.$$

Lemma A.11 (Closedness of the joint bad set)**.** *The set $\mathcal{B}$ is closed in $\mathcal{Y} \times \overline{\mathcal{K}}$.*

Proof. By Lemma A.9, $\mathcal{D}$ is closed in $\overline{\mathcal{K}}$. By definition, $\overline{\mathcal{U}}^{\overline{\mathcal{K}}}$ is closed in $\overline{\mathcal{K}}$. Hence

$$\mathcal{Y} \times \left(\mathcal{D} \cup \overline{\mathcal{U}}^{\overline{\mathcal{K}}}\right)$$

is closed in $\mathcal{Y} \times \overline{\mathcal{K}}$.

Moreover, $\mathrm{graph}(\widetilde{\Sigma})$ is closed in $\overline{\mathcal{K}} \times \mathcal{Y}$. Therefore its coordinate-swapped version

$$\{(\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{Y} \times \overline{\mathcal{K}} : \boldsymbol{y} \in \widetilde{\Sigma}(\boldsymbol{x})\}$$

is closed in $\mathcal{Y} \times \overline{\mathcal{K}}$. Thus $\mathcal{B}$, being the union of two closed sets, is closed. □

Based on the bad set, we can provide a joint-space analogue of Lemma A.3.

Lemma A.12 (Joint local Lipschitzness of the nearest selector)**.** *Define*

$$P(\boldsymbol{y}, \boldsymbol{x}) := P_{\boldsymbol{y}}(\boldsymbol{x}).$$

Then P is locally Lipschitz on

$$(\mathcal{Y} \times \overline{\mathcal{K}}) \setminus \mathcal{B}.$$

Proof. Fix $(\boldsymbol{y}_0, \boldsymbol{x}_0) \notin \mathcal{B}$. Then

$$\boldsymbol{x}_0 \notin \mathcal{D} \cup \overline{\mathcal{U}}^{\overline{\mathcal{K}}}, \qquad \boldsymbol{y}_0 \notin \widetilde{\Sigma}(\boldsymbol{x}_0).$$

Therefore the nearest distinct branch value to $\boldsymbol{y}_0$ at $\boldsymbol{x}_0$ is unique. Let C_* be the corresponding stable collapse cluster, and choose its priority representative i_*.

Since $\boldsymbol{x}_0 \notin \overline{\mathcal{U}}^{\overline{\mathcal{K}}}$, after shrinking a relative neighborhood $V \subset \overline{\mathcal{K}}$ of $\boldsymbol{x}_0$, the collapse pattern of the branches is stable on $V \cap \mathcal{K}$. Choose one representative i_α for each stable collapse cluster C_α. Since C_* is the unique nearest cluster at $(\boldsymbol{y}_0, \boldsymbol{x}_0)$, there is a positive margin

$$\gamma := \min_{\alpha \neq *} \left(\|\boldsymbol{y}_0 - f_{i_\alpha}(\boldsymbol{x}_0)\| - \|\boldsymbol{y}_0 - f_{i_*}(\boldsymbol{x}_0)\|\right) > 0.$$

By continuity of the branch maps and of the functions

$$(\boldsymbol{y}, \boldsymbol{x}) \mapsto \|\boldsymbol{y} - f_{i_\alpha}(\boldsymbol{x})\|,$$

after further shrinking V and taking $\epsilon > 0$ sufficiently small, we have

$$\|\boldsymbol{y} - f_{i_*}(\boldsymbol{x})\| < \|\boldsymbol{y} - f_{i_\alpha}(\boldsymbol{x})\| \qquad \text{for every } \alpha \neq *,$$

for all $(\boldsymbol{y}, \boldsymbol{x}) \in B_\epsilon(\boldsymbol{y}_0) \times V$. Hence the same stable collapse cluster C_* remains the unique nearest cluster throughout this product neighborhood. Therefore,

$$P(\boldsymbol{y}, \boldsymbol{x}) = f_{i_*}(\boldsymbol{x}) \qquad \text{for all } (\boldsymbol{y}, \boldsymbol{x}) \in B_\epsilon(\boldsymbol{y}_0) \times V.$$

Since $\boldsymbol{x}_0 \notin \mathcal{D}$, the branch f_i is locally Lipschitz near $\boldsymbol{x}_0$. Therefore, for all $(\boldsymbol{y}, \boldsymbol{x}), (\boldsymbol{y}', \boldsymbol{x}') \in B_\epsilon(\boldsymbol{y}_0) \times V$,

$$\|P(\boldsymbol{y}, \boldsymbol{x}) - P(\boldsymbol{y}', \boldsymbol{x}')\| = \|f_{i_*}(\boldsymbol{x}) - f_{i_*}(\boldsymbol{x}')\| \leq L\|\boldsymbol{x} - \boldsymbol{x}'\| \leq L\|(\boldsymbol{y}, \boldsymbol{x}) - (\boldsymbol{y}', \boldsymbol{x}')\|.$$

Hence P is locally Lipschitz near $(\boldsymbol{y}_0, \boldsymbol{x}_0)$. Since $(\boldsymbol{y}_0, \boldsymbol{x}_0)$ was arbitrary, P is locally Lipschitz on $(\mathcal{Y} \times \overline{\mathcal{K}}) \setminus \mathcal{B}$. □

A.3 Construction of dynamics

Distance-to-bad-set safety factor. For $(\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{Y} \times \overline{\mathcal{K}}$, set

$$\rho(\boldsymbol{y}, \boldsymbol{x}) := \mathrm{dist}((\boldsymbol{y}, \boldsymbol{x}), \mathcal{B}), \qquad \omega(\boldsymbol{y}, \boldsymbol{x}) := \frac{\rho(\boldsymbol{y}, \boldsymbol{x})}{1 + \rho(\boldsymbol{y}, \boldsymbol{x})}.$$

Because $\mathcal{B}$ is closed,

$$\omega(\boldsymbol{y}, \boldsymbol{x}) = 0 \quad \Longleftrightarrow \quad (\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{B}.$$

and we consider $(\boldsymbol{y}, \boldsymbol{x})$ to be a safe point if this factor ω is large enough. Moreover, for every fixed $\boldsymbol{y} \in \mathcal{Y}$, the map $\boldsymbol{x} \mapsto \omega(\boldsymbol{y}, \boldsymbol{x})$ is 1-Lipschitz on $\mathcal{K}$. Indeed,

$$|\rho(\boldsymbol{y}, \boldsymbol{x}) - \rho(\boldsymbol{y}, \boldsymbol{z})| \leq \|(\boldsymbol{y}, \boldsymbol{x}) - (\boldsymbol{y}, \boldsymbol{z})\|_{\mathcal{Y} \times \overline{\mathcal{K}}} = \|\boldsymbol{x} - \boldsymbol{z}\|,$$

and $r \mapsto r/(1 + r)$ is 1-Lipschitz on $[0, \infty)$. Hence

$$|\omega(\boldsymbol{y}, \boldsymbol{x}) - \omega(\boldsymbol{y}, \boldsymbol{z})| \leq \|\boldsymbol{x} - \boldsymbol{z}\| \qquad \text{for all } \boldsymbol{x}, \boldsymbol{z} \in \mathcal{K}.$$

Safe tubes and safe-set profiles. Define the safe tube (i.e., a clear gap to the bad set):

$$\mathcal{S}(R,s) := \Big\{(\boldsymbol{y},\boldsymbol{x}) \in \mathcal{Y} \times \overline{\mathcal{K}} : \ \|\boldsymbol{y}\| \le R,\ \omega(\boldsymbol{y},\boldsymbol{x}) \ge s.\Big\}$$

And then define safe-set profiles:

$$H_1(R,s) := \sup_{\substack{(\boldsymbol{y},\boldsymbol{x})\in\mathcal{S}(R,s)\\(\boldsymbol{y},\boldsymbol{z})\in\mathcal{S}(R,s)\\ \boldsymbol{x}\neq\boldsymbol{z}}} \frac{\|P_{\boldsymbol{y}}(\boldsymbol{x}) - P_{\boldsymbol{y}}(\boldsymbol{z})\|}{\|\boldsymbol{x}-\boldsymbol{z}\|}.$$

and

$$H_2(R,s) := \sup_{(\boldsymbol{y},\boldsymbol{x})\in\mathcal{S}(R,s)} \|P_{\boldsymbol{y}}(\boldsymbol{x})\|.$$

Lemma A.13. *It holds that*

$$H_1(R,s) < \infty, \qquad H_2(R,s) < \infty$$

for every $R > 0$ and $s > 0$.

Proof. If $s \ge 1$, then the safe sets are empty since $\omega < 1$, and the claim is trivial. Hence assume $0 < s < 1$. Since $\overline{B}_R^{\mathcal{Y}} \times \overline{\mathcal{K}}$ is compact and ω is continuous, $\mathcal{S}(R,s)$ is compact. Moreover,

$$\mathcal{S}(R,s) \subset (\mathcal{Y} \times \overline{\mathcal{K}}) \setminus \mathcal{B}.$$

Thanks to Lemma A.12, $P(\boldsymbol{y},\boldsymbol{x})$ is joint local Lipschitz on $(\mathcal{Y} \times \overline{\mathcal{K}}) \setminus \mathcal{B}$ and the restriction of P to the compact set $\mathcal{S}(R,s)$ is globally Lipschitz. Hence there exists $L_{R,s} < \infty$ such that

$$\|P_{\boldsymbol{y}}(\boldsymbol{x}) - P_{\boldsymbol{y}}(\boldsymbol{z})\| = \|P(\boldsymbol{y},\boldsymbol{x}) - P(\boldsymbol{y},\boldsymbol{z})\| \le L_{R,s}\|(\boldsymbol{y},\boldsymbol{x}) - (\boldsymbol{y},\boldsymbol{z})\| = L_{R,s}\|\boldsymbol{x}-\boldsymbol{z}\|$$

for all $(\boldsymbol{y},\boldsymbol{x}), (\boldsymbol{y},\boldsymbol{z}) \in \mathcal{S}(R,s)$. Therefore $H_1(R,s) \le L_{R,s} < \infty$.

Since P is continuous on the compact set $\mathcal{S}(R,s)$, it is bounded there. Thus there exists $M_{R,s} < \infty$:

$$\|P_{\boldsymbol{y}}(\boldsymbol{x})\| \le M_{R,s} \qquad \text{for all } (\boldsymbol{y},\boldsymbol{x}) \in \mathcal{S}(R,s).$$

Therefore, $H_2(R,s) \le M_{R,s} < \infty$. This completes the proof. □

Damping factor and recurrent operator. Define

$$\lambda_R(s) := \frac{1}{H_1(R,s) + H_2(R,s) + 1}, \qquad s > 0.$$

Define

$$\Theta_R(r) := \int_0^r \lambda_R(s)\, ds, \qquad r \ge 0.$$

Define

$$\eta(\boldsymbol{y},\boldsymbol{x}) := \frac{\Theta_{1+\|\boldsymbol{y}\|}(\omega(\boldsymbol{y},\boldsymbol{x}))}{1 + \Theta_{1+\|\boldsymbol{y}\|}(\omega(\boldsymbol{y},\boldsymbol{x}))}.$$

Finally define

$$g(\boldsymbol{y},\boldsymbol{x}) := (1 - \eta(\boldsymbol{y},\boldsymbol{x}))\boldsymbol{y} + \eta(\boldsymbol{y},\boldsymbol{x})P_{\boldsymbol{y}}(\boldsymbol{x}). \tag{6}$$

Lemma A.14 (Monotonicity)**.** *For fixed $R > 0$, $H_1(R,s)$ and $H_2(R,s)$ are nonincreasing in s, and hence $\lambda_R(s)$ is nondecreasing in s. Moreover, for fixed $s > 0$, $H_1(R,s)$ and $H_2(R,s)$ are nondecreasing in R, and hence $\lambda_R(s)$ is nonincreasing in R.*

Proof. Since the safe tube $\mathcal{S}(R,s)$ shrinks as s increases, $H_1(R,s)$ and $H_2(R,s)$ are nonincreasing in s, $\lambda_R(s)$ is nondecreasing in s. Similar argument for R. □

Lemma A.15 (Positivity and freezing)**.** *For $(\boldsymbol{y},\boldsymbol{x}) \in \mathcal{Y} \times \overline{\mathcal{K}}$,*

$$\eta(\boldsymbol{y},\boldsymbol{x}) = 0 \iff (\boldsymbol{y},\boldsymbol{x}) \in \mathcal{B} \iff \boldsymbol{x} \in \mathcal{D} \cup \mathcal{U}^{\overline{\mathcal{K}}} \ \text{ or } \ \boldsymbol{y} \in \widetilde{\Sigma}(\boldsymbol{x}).$$

In particular, for $\boldsymbol{x} \in \mathcal{K}$, since $\mathcal{D} \cap \mathcal{K} = \varnothing$ and $\mathcal{U}^{\overline{\mathcal{K}}} \cap \mathcal{K} = \mathcal{U}$, we have

$$\boldsymbol{x} \in \mathcal{U} \implies g(\boldsymbol{y},\boldsymbol{x}) = \boldsymbol{y}.$$

Proof. By definition,

$$\eta(\boldsymbol{y}, \boldsymbol{x}) = \frac{\Theta_{1+\|\boldsymbol{y}\|}(\omega(\boldsymbol{y}, \boldsymbol{x}))}{1 + \Theta_{1+\|\boldsymbol{y}\|}(\omega(\boldsymbol{y}, \boldsymbol{x}))}.$$

Since $\lambda_R(s) > 0$ for every $s > 0$, we have $\Theta_R(r) = 0$ if and only if $r = 0$. Hence

$$\eta(\boldsymbol{y}, \boldsymbol{x}) = 0 \iff \omega(\boldsymbol{y}, \boldsymbol{x}) = 0.$$

Moreover,

$$\omega(\boldsymbol{y}, \boldsymbol{x}) = 0 \iff \rho(\boldsymbol{y}, \boldsymbol{x}) = 0 \iff (\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{B},$$

where the last equivalence uses the closedness of $\mathcal{B}$. By the definition of $\mathcal{B}$, this is equivalent to

$$\boldsymbol{x} \in \mathcal{D} \cup \mathcal{U}^{\overline{\mathcal{K}}} \quad \text{or} \quad \boldsymbol{y} \in \widetilde{\Sigma}(\boldsymbol{x}).$$

Finally, if $\boldsymbol{x} \in \mathcal{U}$, then $\boldsymbol{x} \in \mathcal{U}^{\overline{\mathcal{K}}}$, so $(\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{B}$ for every $\boldsymbol{y}$. Thus $\eta(\boldsymbol{y}, \boldsymbol{x}) = 0$, and therefore

$$g(\boldsymbol{y}, \boldsymbol{x}) = (1 - \eta(\boldsymbol{y}, \boldsymbol{x}))\boldsymbol{y} + \eta(\boldsymbol{y}, \boldsymbol{x}) P_{\boldsymbol{y}}(\boldsymbol{x}) = \boldsymbol{y}.$$

This completes the proof. □

A.4 Convergence given fixed x

Lemma A.16 (Graph-closure switching agrees with reduced switching at stable points)**.** *Fix $\boldsymbol{x} \in \mathcal{K} \backslash \mathcal{U}$. Then*

$$\widetilde{\Sigma}(\boldsymbol{x}) = \Sigma^{\circ}(\boldsymbol{x}).$$

Proof. The inclusion $\Sigma^{\circ}(\boldsymbol{x}) \subseteq \widetilde{\Sigma}(\boldsymbol{x})$ is immediate from the definition of graph closure. We prove the reverse inclusion.

Let $\boldsymbol{q} \in \widetilde{\Sigma}(\boldsymbol{x})$. By definition, there exist $\boldsymbol{x}_\ell \to \boldsymbol{x}$ and $\boldsymbol{q}_\ell \to \boldsymbol{q}$ such that

$$\boldsymbol{q}_\ell \in \Sigma^{\circ}(\boldsymbol{x}_\ell).$$

For each ℓ, choose labels $i_\ell \neq j_\ell$ witnessing the switching, namely

$$f_{i_\ell}(\boldsymbol{x}_\ell) \neq f_{j_\ell}(\boldsymbol{x}_\ell),$$

and

$$\|\boldsymbol{q}_\ell - f_{i_\ell}(\boldsymbol{x}_\ell)\| = \|\boldsymbol{q}_\ell - f_{j_\ell}(\boldsymbol{x}_\ell)\| = \min_{1 \le k \le n} \|\boldsymbol{q}_\ell - f_k(\boldsymbol{x}_\ell)\|.$$

Since there are only finitely many label pairs, after passing to a subsequence we may assume

$$i_\ell = i, \qquad j_\ell = j$$

for fixed $i \neq j$.

Because $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$, the collapse pattern is stable near $\boldsymbol{x}$. Hence, after shrinking to a neighborhood of $\boldsymbol{x}$, whether two labels are collapsed is constant. Since

$$f_i(\boldsymbol{x}_\ell) \neq f_j(\boldsymbol{x}_\ell)$$

for infinitely many ℓ, the labels i and j cannot belong to the same stable collapse cluster. Therefore

$$f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x}).$$

Now let $\ell \to \infty$. By continuity of the branch maps,

$$\|\boldsymbol{q} - f_i(\boldsymbol{x})\| = \|\boldsymbol{q} - f_j(\boldsymbol{x})\| = \min_{1 \le k \le n} \|\boldsymbol{q} - f_k(\boldsymbol{x})\|.$$

Together with $f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x})$, this implies $\boldsymbol{q} \in \Sigma^{\circ}(\boldsymbol{x})$. Therefore, $\widetilde{\Sigma}(\boldsymbol{x}) \subseteq \Sigma^{\circ}(\boldsymbol{x})$, and hence $\widetilde{\Sigma}(\boldsymbol{x}) = \Sigma^{\circ}(\boldsymbol{x})$. □

Lemma A.17 (Voronoi-segment invariance)**.** *Fix $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$. Let $\boldsymbol{y}_0 \notin \Sigma^{\circ}(\boldsymbol{x})$, and define $\boldsymbol{u} := P_{\boldsymbol{y}_0}(\boldsymbol{x})$. Then for every $\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]$ (Here, $[\boldsymbol{y}_0, \boldsymbol{u}]$ represents the line segment connecting the endpoints $\boldsymbol{y}_0$ and $\boldsymbol{u}$.),*

$$P_{\boldsymbol{z}}(\boldsymbol{x}) = \boldsymbol{u}.$$

In particular,

$$[\boldsymbol{y}_0, \boldsymbol{u}] \cap \widetilde{\Sigma}(\boldsymbol{x}) = \varnothing.$$

Proof. Since $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$, Lemma A.16 gives $\widetilde{\Sigma}(\boldsymbol{x}) = \Sigma^{\circ}(\boldsymbol{x})$ and hence

$$\boldsymbol{y}_0 \notin \widetilde{\Sigma}(\boldsymbol{x}).$$

Hence the selected value $\boldsymbol{u} = P_{\boldsymbol{y}_0}(\boldsymbol{x})$ is the unique nearest branch value to $\boldsymbol{y}_0$.

Let $\boldsymbol{v} \in \mathcal{F}(\boldsymbol{x})$ be any distinct value with $\boldsymbol{v} \neq \boldsymbol{u}$. Then

$$\|\boldsymbol{y}_0 - \boldsymbol{u}\| < \|\boldsymbol{y}_0 - \boldsymbol{v}\|.$$

For $t \in [0, 1]$, set

$$\boldsymbol{z}_t := (1 - t)\boldsymbol{y}_0 + t\boldsymbol{u}.$$

Then

$$\|\boldsymbol{z}_t - \boldsymbol{u}\| = (1 - t)\|\boldsymbol{y}_0 - \boldsymbol{u}\|.$$

By the triangle inequality,

$$\|\boldsymbol{z}_t - \boldsymbol{v}\| \geq \|\boldsymbol{y}_0 - \boldsymbol{v}\| - \|\boldsymbol{y}_0 - \boldsymbol{z}_t\| = \|\boldsymbol{y}_0 - \boldsymbol{v}\| - t\|\boldsymbol{y}_0 - \boldsymbol{u}\|.$$

Since $\|\boldsymbol{y}_0 - \boldsymbol{v}\| > \|\boldsymbol{y}_0 - \boldsymbol{u}\|$, we obtain

$$\|\boldsymbol{z}_t - \boldsymbol{v}\| > (1 - t)\|\boldsymbol{y}_0 - \boldsymbol{u}\| = \|\boldsymbol{z}_t - \boldsymbol{u}\|.$$

Therefore $\boldsymbol{u}$ remains the unique nearest branch value for every $\boldsymbol{z}_t \in [\boldsymbol{y}_0, \boldsymbol{u}]$. Hence

$$P_{\boldsymbol{z}_t}(\boldsymbol{x}) = \boldsymbol{u}$$

for every $t \in [0, 1]$, and then $\boldsymbol{z}_t \notin \Sigma^{\circ}(\boldsymbol{x})$ for every $t \in [0, 1]$.

Using again $\widetilde{\Sigma}(\boldsymbol{x}) = \Sigma^{\circ}(\boldsymbol{x})$, we conclude that

$$\boldsymbol{z}_t \notin \widetilde{\Sigma}(\boldsymbol{x})$$

for every $t \in [0, 1]$. Therefore $[\boldsymbol{y}_0, \boldsymbol{u}] \cap \widetilde{\Sigma}(\boldsymbol{x}) = \varnothing$, which concludes the proof. □

Lemma A.18 (Uniform positive bad-set distance along the Voronoi segment)**.** *Fix $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$. Let $\boldsymbol{y}_0 \notin \Sigma^{\circ}(\boldsymbol{x})$ and define $\boldsymbol{u} := P_{\boldsymbol{y}_0}(\boldsymbol{x})$. Then*

$$\inf_{\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]} \rho(\boldsymbol{z}, \boldsymbol{x}) > 0.$$

Proof. Define

$$\Gamma := \{(\boldsymbol{z}, \boldsymbol{x}) : \boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]\} \subset \mathcal{Y} \times \overline{\mathcal{K}}.$$

The set Γ is compact because $[\boldsymbol{y}_0, \boldsymbol{u}]$ is compact.

We claim that

$$\Gamma \cap \mathcal{B} = \varnothing.$$

Indeed, since $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$ and $\mathcal{D} \subset \overline{\mathcal{K}} \setminus \mathcal{K}$, we have $\boldsymbol{x} \notin \mathcal{U} \cup \mathcal{D}$. Moreover, by Lemma A.17, $[\boldsymbol{y}_0, \boldsymbol{u}] \cap \widetilde{\Sigma}(\boldsymbol{x}) = \varnothing$. Therefore, for every $\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]$, the point $(\boldsymbol{z}, \boldsymbol{x})$ belongs to none of the components defining $\mathcal{B}$. Hence $\Gamma \cap \mathcal{B} = \varnothing$.

Since $\mathcal{B}$ is closed and Γ is compact, the continuous function

$$(\boldsymbol{z}, \boldsymbol{x}) \mapsto \operatorname{dist}((\boldsymbol{z}, \boldsymbol{x}), \mathcal{B})$$

attains a strictly positive minimum on Γ. Therefore

$$\inf_{\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]} \rho(\boldsymbol{z}, \boldsymbol{x}) = \operatorname{dist}(\Gamma, \mathcal{B}) > 0.$$

This proves the claim. □

Lemma A.19 (Positive damping)**.** *Fix $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$. Let $\boldsymbol{y}_0 \notin \Sigma^{\circ}(\boldsymbol{x})$ and define $\boldsymbol{u} := P_{\boldsymbol{y}_0}(\boldsymbol{x})$. Then*

$$c := \inf_{\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]} \eta(\boldsymbol{z}, \boldsymbol{x}) > 0.$$

Proof. By Lemma A.18, there exists $\rho_0 > 0$ such that

$$\rho(\boldsymbol{z}, \boldsymbol{x}) \geq \rho_0$$

for every $\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]$. Hence

$$\omega(\boldsymbol{z}, \boldsymbol{x}) = \frac{\rho(\boldsymbol{z}, \boldsymbol{x})}{1 + \rho(\boldsymbol{z}, \boldsymbol{x})} \geq \frac{\rho_0}{1 + \rho_0} =: \omega_0 > 0.$$

Let

$$R_0 := 1 + \max_{\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]} \|\boldsymbol{z}\|.$$

Then

$$1 + \|\boldsymbol{z}\| \leq R_0$$

for every $\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]$. Since $H_1(R, s)$ and $H_2(R, s)$ are nondecreasing in R, the function $\lambda_R(s)$ is nonincreasing in R. Therefore

$$\Theta_{1+\|\boldsymbol{z}\|}(\omega_0) \geq \Theta_{R_0}(\omega_0).$$

Thus, for every $\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]$,

$$\eta(\boldsymbol{z}, \boldsymbol{x}) \geq \frac{\Theta_{R_0}(\omega_0)}{1 + \Theta_{R_0}(\omega_0)} =: c.$$

Since $\omega_0 > 0$ and $\lambda_{R_0}(s) > 0$ for every $s > 0$,

$$\Theta_{R_0}(\omega_0) = \int_0^{\omega_0} \lambda_{R_0}(s)\, ds > 0.$$

Hence $c > 0$. □

Theorem A.20 (Convergence with fixed $\boldsymbol{x}$)**.** *Fix $\boldsymbol{x} \in \mathcal{K} \backslash \mathcal{U}$. Let $\boldsymbol{y}_0 \notin \Sigma^\circ(\boldsymbol{x})$ and define $\boldsymbol{u} := P_{\boldsymbol{y}_0}(\boldsymbol{x})$. Consider the iteration*

$$\boldsymbol{y}_{t+1} = g(\boldsymbol{y}_t, \boldsymbol{x}).$$

Then

$$\boldsymbol{y}_t \to \boldsymbol{u}.$$

More precisely, there exists $c > 0$ such that

$$\|\boldsymbol{y}_t - \boldsymbol{u}\| \leq (1 - c)^t \|\boldsymbol{y}_0 - \boldsymbol{u}\|.$$

where c may depend on $\boldsymbol{x}$ and $\boldsymbol{y}_0$.

Proof. By Lemma A.19,

$$c := \inf_{\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]} \eta(\boldsymbol{z}, \boldsymbol{x}) > 0.$$

By Lemma A.17,

$$P_{\boldsymbol{z}}(\boldsymbol{x}) = \boldsymbol{u}$$

for every $\boldsymbol{z} \in [\boldsymbol{y}_0, \boldsymbol{u}]$.

We prove by induction that

$$\boldsymbol{y}_t \in [\boldsymbol{y}_0, \boldsymbol{u}]$$

for every t. This is true for $t = 0$. If $\boldsymbol{y}_t \in [\boldsymbol{y}_0, \boldsymbol{u}]$, then

$$\boldsymbol{y}_{t+1} = (1 - \eta(\boldsymbol{y}_t, \boldsymbol{x}))\boldsymbol{y}_t + \eta(\boldsymbol{y}_t, \boldsymbol{x})\boldsymbol{u}.$$

Since

$$0 \leq \eta(\boldsymbol{y}_t, \boldsymbol{x}) \leq 1,$$

we have

$$\boldsymbol{y}_{t+1} \in [\boldsymbol{y}_t, \boldsymbol{u}] \subset [\boldsymbol{y}_0, \boldsymbol{u}].$$

Thus $\boldsymbol{y}_t \in [\boldsymbol{y}_0, \boldsymbol{u}]$ for all t.

Therefore

$$\boldsymbol{y}_{t+1} - \boldsymbol{u} = (1 - \eta(\boldsymbol{y}_t, \boldsymbol{x}))(\boldsymbol{y}_t - \boldsymbol{u}),$$

and so

$$\|\boldsymbol{y}_{t+1} - \boldsymbol{u}\| \leq (1 - c)\|\boldsymbol{y}_t - \boldsymbol{u}\|.$$

Iterating gives

$$\|\boldsymbol{y}_t - \boldsymbol{u}\| \leq (1 - c)^t \|\boldsymbol{y}_0 - \boldsymbol{u}\|.$$

Therefore $\boldsymbol{y}_t \to \boldsymbol{u}$. □

A.5 Input regularity

Lemma A.21. *Fix $\boldsymbol{y} \in \mathcal{Y}$. The map $\boldsymbol{x} \mapsto \eta(\boldsymbol{y}, \boldsymbol{x}) P_{\boldsymbol{y}}(\boldsymbol{x})$ is globally Lipschitz on $\overline{\mathcal{K}}$.*

Proof. Fix $\boldsymbol{y}$. Set $R := 1 + \|\boldsymbol{y}\|$. Then $\|\boldsymbol{y}\| \le R$, and

$$\eta(\boldsymbol{y}, \boldsymbol{x}) = \frac{\Theta_R(\omega(\boldsymbol{y}, \boldsymbol{x}))}{1 + \Theta_R(\omega(\boldsymbol{y}, \boldsymbol{x}))}.$$

For simplicity, we write

$$\omega(\boldsymbol{x}) := \omega(\boldsymbol{y}, \boldsymbol{x}), \qquad P(\boldsymbol{x}) := P_{\boldsymbol{y}}(\boldsymbol{x}), \qquad \eta(\boldsymbol{x}) := \eta(\boldsymbol{y}, \boldsymbol{x}).$$

Since $\boldsymbol{x} \mapsto \omega(\boldsymbol{x})$ is 1-Lipschitz, we have

$$|\omega(\boldsymbol{x}) - \omega(\boldsymbol{z})| \le \|\boldsymbol{x} - \boldsymbol{z}\|.$$

Case 1: If $(\boldsymbol{y}, \boldsymbol{x}), (\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{B}$, according to Lemma A.15, we have $\eta(\boldsymbol{x}) = \eta(\boldsymbol{z}) = 0$ and hence

$$\eta(\boldsymbol{x}) P(\boldsymbol{x}) = \eta(\boldsymbol{z}) P(\boldsymbol{z}) = 0.$$

Case 2: If $(\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{B}$ and $(\boldsymbol{y}, \boldsymbol{z}) \notin \mathcal{B}$, set

$$r := \omega(\boldsymbol{z}) > 0.$$

Then $(\boldsymbol{y}, \boldsymbol{z}) \in \mathcal{S}(R, r)$, so

$$\|P(\boldsymbol{z})\| \le H_2(R, r).$$

Also,

$$\eta(\boldsymbol{z}) \le \Theta_R(r) \le r \lambda_R(r).$$

Hence

$$\|\eta(\boldsymbol{z}) P(\boldsymbol{z})\| \le r \lambda_R(r) H_2(R, r) \le r.$$

Since

$$r = \omega(\boldsymbol{z}) - \omega(\boldsymbol{x}) \le \|\boldsymbol{z} - \boldsymbol{x}\|,$$

we get

$$\|\eta(\boldsymbol{x}) P(\boldsymbol{x}) - \eta(\boldsymbol{z}) P(\boldsymbol{z})\| \le \|\boldsymbol{x} - \boldsymbol{z}\|.$$

Case 3: Now suppose $(\boldsymbol{y}, \boldsymbol{x}), (\boldsymbol{y}, \boldsymbol{z}) \notin \mathcal{B}$. Without loss of generality, assume

$$r_1 := \omega(\boldsymbol{x}) \le r_2 := \omega(\boldsymbol{z}).$$

Then $(\boldsymbol{y}, \boldsymbol{z}), (\boldsymbol{y}, \boldsymbol{x}) \in \mathcal{S}(R, r_1)$, so

$$\|P(\boldsymbol{x}) - P(\boldsymbol{z})\| \le H_1(R, r_1) \|\boldsymbol{x} - \boldsymbol{z}\|.$$

Also,

$$\|P(\boldsymbol{z})\| \le H_2(R, r_2).$$

Therefore

$$\big\|\eta(\boldsymbol{x}) P(\boldsymbol{x}) - \eta(\boldsymbol{z}) P(\boldsymbol{z})\big\| \le \eta(\boldsymbol{x}) \|P(\boldsymbol{x}) - P(\boldsymbol{z})\| + |\eta(\boldsymbol{x}) - \eta(\boldsymbol{z})| \big\|P(\boldsymbol{z})\big\|.$$

For the first term,

$$\eta(\boldsymbol{x}) \le \Theta_R(r_1) \le r_1 \lambda_R(r_1),$$

so

$$\eta(\boldsymbol{x}) \|P(\boldsymbol{x}) - P(\boldsymbol{z})\| \le r_1 \lambda_R(r_1) H_1(R, r_1) \|\boldsymbol{x} - \boldsymbol{z}\| \le \|\boldsymbol{x} - \boldsymbol{z}\|.$$

For the second term, since λ_R is nondecreasing,

$$|\eta(\boldsymbol{x}) - \eta(\boldsymbol{z})| \le |\Theta_R(r_1) - \Theta_R(r_2)| \le (r_2 - r_1) \lambda_R(r_2).$$

Thus

$$|\eta(\boldsymbol{x}) - \eta(\boldsymbol{z})| \big\|P(\boldsymbol{z})\big\| \le (r_2 - r_1) \lambda_R(r_2) H_2(R, r_2) \le r_2 - r_1.$$

Since ω is 1-Lipschitz,

$$r_2 - r_1 \le \|\boldsymbol{x} - \boldsymbol{z}\|.$$

Combining the two estimates gives

$$\|\eta(\boldsymbol{x}) P(\boldsymbol{x}) - \eta(\boldsymbol{z}) P(\boldsymbol{z})\| \le 2 \|\boldsymbol{x} - \boldsymbol{z}\|.$$

Therefore $\boldsymbol{x} \mapsto \eta(\boldsymbol{y}, \boldsymbol{x}) P_{\boldsymbol{y}}(\boldsymbol{x})$ is globally Lipschitz on $\overline{\mathcal{K}}$. □

Theorem A.22 (Global Lipschitzness of the damped operator)**.** *For every fixed $\boldsymbol{y} \in \mathcal{Y}$, the map*

$$\boldsymbol{x} \mapsto g(\boldsymbol{y}, \boldsymbol{x})$$

is globally Lipschitz on $\overline{\mathcal{K}}$.

Proof. We write

$$g(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{y} - \eta(\boldsymbol{y}, \boldsymbol{x})\boldsymbol{y} + \eta(\boldsymbol{y}, \boldsymbol{x})P_{\boldsymbol{y}}(\boldsymbol{x}).$$

By Lemma A.21, $\boldsymbol{x} \mapsto \eta(\boldsymbol{y}, \boldsymbol{x})P_{\boldsymbol{y}}(\boldsymbol{x})$ is globally Lipschitz.

It remains to control $\boldsymbol{x} \mapsto \eta(\boldsymbol{y}, \boldsymbol{x})\boldsymbol{y}$. Since $\boldsymbol{y}$ is fixed, it is enough to control $\boldsymbol{x} \mapsto \eta(\boldsymbol{y}, \boldsymbol{x})$. Because $r \mapsto r/(1+r)$ is 1-Lipschitz and

$$\Theta_R'(r) = \lambda_R(r) \leq 1$$

for a.e. r, we have

$$|\eta(\boldsymbol{y}, \boldsymbol{x}) - \eta(\boldsymbol{y}, \boldsymbol{z})| \leq |\omega(\boldsymbol{y}, \boldsymbol{x}) - \omega(\boldsymbol{y}, \boldsymbol{z})| \leq \|\boldsymbol{x} - \boldsymbol{z}\|.$$

Therefore

$$\|\eta(\boldsymbol{y}, \boldsymbol{x})\boldsymbol{y} - \eta(\boldsymbol{y}, \boldsymbol{z})\boldsymbol{y}\| \leq \|\boldsymbol{y}\|\|\boldsymbol{x} - \boldsymbol{z}\|.$$

Combining the two estimates, $\boldsymbol{x} \mapsto g(\boldsymbol{y}, \boldsymbol{x})$ is globally Lipschitz on $\overline{\mathcal{K}}$. □

A.6 Final proof of Theorem 2.2

Finally, we restate Theorem 2.2 in a more complete version and prove it.

Theorem A.23. *Let $\mathcal{K} \subset \mathbb{R}^d$ be bounded and let $\mathcal{Y} = \mathbb{R}^m$ with the Euclidean norm. Assume $f_1, \ldots, f_n : \mathcal{K} \to \mathcal{Y}$ be locally Lipschitz. These branches can collapse. Then there exists $g : \mathcal{Y} \times \mathcal{K} \to \mathcal{Y}$ satisfying:*

1. *For every $\boldsymbol{y} \in \mathcal{Y}$, the map $\boldsymbol{x} \mapsto g(\boldsymbol{y}, \boldsymbol{x})$ is globally Lipschitz on $\mathcal{K}$.*
2. *For every $\boldsymbol{x} \in \mathcal{U}$, $g(\boldsymbol{y}, \boldsymbol{x}) = \boldsymbol{y}$ for every $\boldsymbol{y} \in \mathcal{Y}$. Hence the iteration freezes at unstable collapse points.*
3. *For every $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$ and every $\boldsymbol{y}_0 \notin \Sigma^\circ(\boldsymbol{x})$, the iteration $\boldsymbol{y}_{t+1} = g(\boldsymbol{y}_t, \boldsymbol{x})$ converges to one of the branch in $\mathcal{F}(\boldsymbol{x})$:*
   $$\boldsymbol{y}_t \to P_{\boldsymbol{y}_0}(\boldsymbol{x}) \in \mathcal{F}(\boldsymbol{x})$$
   Moreover, $\Sigma^\circ(\boldsymbol{x})$ has Lebesgue measure zero in $\mathcal{Y}$. Therefore, for every $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$, the convergence statement holds for Lebesgue-almost every initialization $\boldsymbol{y}_0 \in \mathcal{Y}$.
4. *For every $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$, it holds that*
   $$\mathcal{A}_g(\boldsymbol{x}) = \left\{ \lim_{t \to \infty} \boldsymbol{y}_t(\boldsymbol{x}; \boldsymbol{y}_0) : \boldsymbol{y}_0 \in \mathcal{Y} \setminus \Sigma^\circ(\boldsymbol{x}) \right\} = \mathcal{F}(\boldsymbol{x}).$$
   In other words, every branch in $\mathcal{F}(\boldsymbol{x})$ can be achieved

Proof. The first claim follows from Theorem A.22.

The second claim follows from Lemma A.15.

The third claim follows from Theorem A.20. In addition, For a stable $\boldsymbol{x}$, the graph-closure switching set agrees locally with the ordinary reduced Voronoi switching set (Lemma A.16). Since $\mathcal{F}(\boldsymbol{x})$ is finite, $\Sigma^\circ(\boldsymbol{x})$ is contained in a finite union of pairwise bisector hyperplanes. Hence $\Sigma^\circ(\boldsymbol{x})$ has Lebesgue measure zero in $\mathcal{Y}$.

For the fourth claim, fix $\boldsymbol{x} \in \mathcal{K} \setminus \mathcal{U}$. By the third claim, every initialization $\boldsymbol{y}_0 \notin \Sigma^\circ(\boldsymbol{x})$ converges to some element of $\mathcal{F}(\boldsymbol{x})$. Hence

$$\left\{ \lim_{t \to \infty} \boldsymbol{y}_t(\boldsymbol{x}; \boldsymbol{y}_0) : \boldsymbol{y}_0 \in \mathcal{Y} \setminus \Sigma^\circ(\boldsymbol{x}) \right\} \subset \mathcal{F}(\boldsymbol{x}).$$

Conversely, take any $\boldsymbol{u} \in \mathcal{F}(\boldsymbol{x})$. Since $\mathcal{F}(\boldsymbol{x})$ is finite, $\boldsymbol{u}$ is the unique nearest branch value throughout some ball centered at $\boldsymbol{u}$. Every initialization in this ball lies outside $\Sigma^\circ(\boldsymbol{x})$ and converges to $\boldsymbol{u}$ by Theorem A.20; moreover, $g(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{u}$. Thus each branch is a locally attracting fixed point, proving the reverse inclusion. No point outside $\mathcal{F}(\boldsymbol{x})$ can attract a neighborhood, since every neighborhood contains initializations outside the measure-zero set $\Sigma^\circ(\boldsymbol{x})$, whose limits lie in $\mathcal{F}(\boldsymbol{x})$. Therefore $\mathcal{A}_g(\boldsymbol{x}) = \mathcal{F}(\boldsymbol{x})$ as defined in (2). □

B Proof of Theorem 2.3

We first describe Theorem 2.3 in a more precise version and then provide its complete proof.

Theorem B.1. *Let $\mathcal{K} \subset \mathbb{R}^d$ be bounded and measurable, let $\mathcal{Y} = \mathbb{R}^m$, $m \geq 1$, and let $f_1, \ldots, f_n : \mathcal{K} \to \mathcal{Y}$ be locally Lipschitz branch maps. Let $\mathcal{K}^* := \mathcal{K} \setminus \mathcal{U}$. Let $g : \mathcal{Y} \times \mathcal{K} \to \mathcal{Y}$ be the recurrent operator constructed in Theorem 2.2. For $\boldsymbol{y}_0 \in \mathcal{Y}$, define the induced solution selector*

$$u(\boldsymbol{y}_0, \boldsymbol{x}) := \lim_{t \to \infty} g^t(\boldsymbol{y}_0, \boldsymbol{x}),$$

whenever the limit exists.

Then there exists a Lebesgue-null set $\mathcal{E} \subset \mathcal{Y}$ such that for every $\boldsymbol{y}_0 \in \mathcal{Y} \setminus \mathcal{E}$, the fixed-initialization switching set

$$\mathcal{N}(\boldsymbol{y}_0) := \{\boldsymbol{x} \in \mathcal{K}^\star : \boldsymbol{y}_0 \in \Sigma^\circ(\boldsymbol{x})\}$$

has zero d-dimensional Lebesgue measure:

$$\mathcal{L}^d(\mathcal{N}(\boldsymbol{y}_0)) = 0.$$

Moreover, for every $\boldsymbol{y}_0 \in \mathcal{Y} \setminus \mathcal{E}$, the limit $u(\boldsymbol{y}_0, \boldsymbol{x})$ exists for all $\boldsymbol{x} \in \mathcal{K}^\star \setminus \mathcal{N}(\boldsymbol{y}_0)$, and the map $\boldsymbol{x} \mapsto u(\boldsymbol{y}_0, \boldsymbol{x})$ is locally Lipschitz on $\mathcal{K}^\star \setminus \mathcal{N}(\boldsymbol{y}_0)$.

Note: In Appendix B, we write $\mathcal{L}^d$ for d-dimensional Lebesgue measure to avoid ambiguity in the Fubini argument. Elsewhere, we use $|\cdot|$ for Lebesgue measure in the relevant ambient space.

Proof. We first show that for almost every fixed initialization $\boldsymbol{y}_0$, the switching set $\mathcal{N}(\boldsymbol{y}_0)$ has zero measure. For fixed $\boldsymbol{x} \in \mathcal{K}^\star$, the switching set $\Sigma^\circ(\boldsymbol{x})$ is contained in a finite union of affine hyperplanes in $\mathcal{Y}$. Indeed, for each $i \neq j$ with $f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x})$, expanding $\|\boldsymbol{y} - f_i(\boldsymbol{x})\|^2 = \|\boldsymbol{y} - f_j(\boldsymbol{x})\|^2$ gives

$$2\langle \boldsymbol{y}, f_j(\boldsymbol{x}) - f_i(\boldsymbol{x})\rangle = \|f_j(\boldsymbol{x})\|^2 - \|f_i(\boldsymbol{x})\|^2.$$

Since $f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x})$, the vector $f_j(\boldsymbol{x}) - f_i(\boldsymbol{x})$ is nonzero, and hence this equation defines a genuine affine hyperplane in $\mathcal{Y} = \mathbb{R}^m$. Therefore it has m-dimensional Lebesgue measure zero. Since there are only finitely many pairs (i, j), we obtain

$$\mathcal{L}^m(\Sigma^\circ(\boldsymbol{x})) = 0 \qquad \text{for every } \boldsymbol{x} \in \mathcal{K}^\star.$$

Now define the switching incidence set in the joint space

$$\Gamma := \{(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{K}^\star \times \mathcal{Y} : \ \boldsymbol{y} \in \Sigma^\circ(\boldsymbol{x})\}.$$

Equivalently,

$$\Gamma = \bigcup_{1 \leq i < j \leq n} \Gamma_{ij},$$

where Γ_{ij} is the set of all $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{K}^\star \times \mathcal{Y}$ such that

$$f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x}), \qquad \|\boldsymbol{y} - f_i(\boldsymbol{x})\| = \|\boldsymbol{y} - f_j(\boldsymbol{x})\| = \min_{1 \leq k \leq n} \|\boldsymbol{y} - f_k(\boldsymbol{x})\|,$$

Each Γ_{ij} is measurable because it is described by finitely many equalities and inequalities involving continuous functions of $(\boldsymbol{x}, \boldsymbol{y})$. Hence Γ is measurable.

For $R > 0$, define the closed radius-R ball in the initialization space $\mathcal{Y}$ by

$$\mathbb{B}_R^{\mathcal{Y}} := \{\boldsymbol{y} \in \mathcal{Y} : \|\boldsymbol{y}\| \leq R\}.$$

We use $\mathbb{B}_R^{\mathcal{Y}}$ only to localize the argument to a finite-measure set: since $\mathcal{K}$ is bounded and measurable, $\mathcal{K}^\star \subset \mathcal{K}$ has finite Lebesgue measure, and $\mathbb{B}_R^{\mathcal{Y}}$ also has finite Lebesgue measure. Thus the indicator function

$$\mathbf{1}_{\Gamma \cap (\mathcal{K}^\star \times \mathbb{B}_R^{\mathcal{Y}})}$$

is integrable on $\mathcal{K}^\star \times \mathbb{B}_R^{\mathcal{Y}}$, so Fubini's theorem applies.

For fixed $\boldsymbol{x} \in \mathcal{K}^\star$, define the vertical fiber

$$\Gamma_{\boldsymbol{x}}^R := \{\boldsymbol{y} \in \mathbb{B}_R^{\mathcal{Y}} : (\boldsymbol{x}, \boldsymbol{y}) \in \Gamma\}.$$

By definition,

$$\Gamma_{\boldsymbol{x}}^R = \Sigma^\circ(\boldsymbol{x}) \cap \mathbb{B}_R^{\mathcal{Y}}.$$

Therefore

$$\mathcal{L}^m(\Gamma_{\boldsymbol{x}}^R) = \mathcal{L}^m(\Sigma^\circ(\boldsymbol{x}) \cap \mathbb{B}_R^{\mathcal{Y}}) = 0 \qquad \text{for every } \boldsymbol{x} \in \mathcal{K}^\star.$$

Applying Fubini's theorem [8, Theorem 1.22] in the order y-then-x, we get

$$\mathcal{L}^{d+m}\big(\Gamma \cap (\mathcal{K}^\star \times \mathbb{B}_R^{\mathcal{Y}})\big) = \int_{\mathcal{K}^\star} \left(\int_{\mathbb{B}_R^{\mathcal{Y}}} \mathbf{1}_\Gamma(\boldsymbol{x}, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{y} \right) \mathrm{d}\boldsymbol{x} = \int_{\mathcal{K}^\star} \mathcal{L}^m(\Gamma_{\boldsymbol{x}}^R) \, \mathrm{d}\boldsymbol{x} = 0$$

Next apply Fubini's theorem in the other order. For fixed $\boldsymbol{y} \in \mathbb{B}_R^{\mathcal{Y}}$, define the horizontal fiber

$$\Gamma_R^{\boldsymbol{y}} := \{\boldsymbol{x} \in \mathcal{K}^\star : (\boldsymbol{x}, \boldsymbol{y}) \in \Gamma\}.$$

By definition,

$$\Gamma_R^{\boldsymbol{y}} = \{\boldsymbol{x} \in \mathcal{K}^\star : \boldsymbol{y} \in \Sigma^\circ(\boldsymbol{x})\} = \mathcal{N}(\boldsymbol{y}).$$

Therefore,

$$\begin{aligned} 0 = \mathcal{L}^{d+m}\big(\Gamma \cap (\mathcal{K}^\star \times \mathbb{B}_R^{\mathcal{Y}})\big) &= \int_{\mathbb{B}_R^{\mathcal{Y}}} \left(\int_{\mathcal{K}^\star} \mathbf{1}_\Gamma(\boldsymbol{x}, \boldsymbol{y}) \, \mathrm{d}\boldsymbol{x} \right) \mathrm{d}\boldsymbol{y} \\ &= \int_{\mathbb{B}_R^{\mathcal{Y}}} \mathcal{L}^d(\Gamma_R^{\boldsymbol{y}}) \, \mathrm{d}\boldsymbol{y} = \int_{\mathbb{B}_R^{\mathcal{Y}}} \mathcal{L}^d(\mathcal{N}(\boldsymbol{y})) \, \mathrm{d}\boldsymbol{y}. \end{aligned}$$

The function $\boldsymbol{y} \mapsto \mathcal{L}^d(\mathcal{N}(\boldsymbol{y}))$ is nonnegative. Hence the last identity implies that

$$\mathcal{L}^d(\mathcal{N}(\boldsymbol{y})) = 0$$

for Lebesgue-a.e. $\boldsymbol{y} \in \mathbb{B}_R^{\mathcal{Y}}$.

Let $\mathcal{E}_R \subset \mathbb{B}_R^{\mathcal{Y}}$ be the exceptional set

$$\mathcal{E}_R := \{\boldsymbol{y} \in \mathbb{B}_R^{\mathcal{Y}} : \mathcal{L}^d(\mathcal{N}(\boldsymbol{y})) > 0\}.$$

Then

$$\mathcal{L}^m(\mathcal{E}_R) = 0.$$

Now define

$$\mathcal{E} := \bigcup_{\ell=1}^{\infty} \mathcal{E}_\ell,$$

where $\mathcal{E}_\ell$ is the exceptional set corresponding to the ball $\mathbb{B}_\ell^{\mathcal{Y}}$. Since $\mathcal{E}$ is a countable union of Lebesgue-null sets,

$$\mathcal{L}^m(\mathcal{E}) = 0.$$

Take any $\boldsymbol{y}_0 \in \mathcal{Y} \setminus \mathcal{E}$. Choose an integer ℓ large enough so that $\boldsymbol{y}_0 \in \mathbb{B}_\ell^{\mathcal{Y}}$. Since $\boldsymbol{y}_0 \notin \mathcal{E}_\ell$, we have

$$\mathcal{L}^d(\mathcal{N}(\boldsymbol{y}_0)) = 0.$$

This proves the desired almost-everywhere statement for fixed initializations.

It remains to prove the local Lipschitz property. Fix $\boldsymbol{y}_0 \in \mathcal{Y} \setminus \mathcal{E}$ and $\boldsymbol{x}_0 \in \mathcal{K}^\star \setminus \mathcal{N}(\boldsymbol{y}_0)$. Since $\boldsymbol{x}_0 \notin \mathcal{N}(\boldsymbol{y}_0)$, we have $\boldsymbol{y}_0 \notin \Sigma^\circ(\boldsymbol{x}_0)$. Thus, among the distinct branch values in

$$\mathcal{F}(\boldsymbol{x}_0) = \{f_1(\boldsymbol{x}_0), \ldots, f_n(\boldsymbol{x}_0)\},$$

there is a unique branch value closest to $\boldsymbol{y}_0$.

Because $\boldsymbol{x}_0 \in \mathcal{K}^\star$, the branch collapse pattern is stable near $\boldsymbol{x}_0$. Therefore there exist a relative neighborhood $\mathbb{V}_0 \subset \mathcal{K}$ of $\boldsymbol{x}_0$ and a partition

$$\{1, \ldots, n\} = C_1 \cup \cdots \cup C_q$$

such that, for every $\boldsymbol{x} \in \mathbb{V}_0$,

$$f_i(\boldsymbol{x}) = f_j(\boldsymbol{x}) \quad \Longleftrightarrow \quad i, j \in C_\alpha \text{ for some } \alpha.$$

Choose one representative $i_\alpha \in C_\alpha$ for each cluster C_α. Since $\boldsymbol{y}_0 \notin \Sigma^\circ(\boldsymbol{x}_0)$, there is a unique cluster $C_{\alpha_\star}$ whose branch value is closest to $\boldsymbol{y}_0$ at $\boldsymbol{x}_0$. That is,

$$\|\boldsymbol{y}_0 - f_{i_{\alpha_\star}}(\boldsymbol{x}_0)\| < \|\boldsymbol{y}_0 - f_{i_\beta}(\boldsymbol{x}_0)\| \qquad \text{for every } \beta \neq \alpha_\star.$$

If $q = 1$, then there is only one distinct branch value near $\boldsymbol{x}_0$, and the conclusion is immediate. Otherwise, define the positive margin

$$\delta := \min_{\beta \neq \alpha_\star} \left[\|\boldsymbol{y}_0 - f_{i_\beta}(\boldsymbol{x}_0)\| - \|\boldsymbol{y}_0 - f_{i_{\alpha_\star}}(\boldsymbol{x}_0)\|\right] > 0.$$

By continuity of the finitely many branch maps, after shrinking $\mathbb{V}_0$ if necessary, we may ensure that for every $\boldsymbol{x} \in \mathbb{V}_0$ and every $\beta \neq \alpha_\star$,

$$\|\boldsymbol{y}_0 - f_{i_{\alpha_\star}}(\boldsymbol{x})\| < \|\boldsymbol{y}_0 - f_{i_\beta}(\boldsymbol{x})\|.$$

Thus the same cluster $C_{\alpha_\star}$ remains the unique nearest branch cluster throughout $\mathbb{V}_0$. Consequently,

$$P_{\boldsymbol{y}_0}(\boldsymbol{x}) = f_{i_{\alpha_\star}}(\boldsymbol{x}) \qquad \text{for every } \boldsymbol{x} \in \mathbb{V}_0.$$

In particular, it holds that

$$\mathbb{V}_0 \cap \mathcal{N}(\boldsymbol{y}_0) = \emptyset,$$

and equivalently

$$\boldsymbol{y}_0 \notin \Sigma^\circ(\boldsymbol{x}), \qquad \text{for every } \boldsymbol{x} \in \mathbb{V}_0.$$

By Theorem A.20, for every $\boldsymbol{x} \in \mathbb{V}_0$,

$$u(\boldsymbol{y}_0, \boldsymbol{x}) = \lim_{t\to\infty} g^t(\boldsymbol{y}_0, \boldsymbol{x}) = P_{\boldsymbol{y}_0}(\boldsymbol{x}) = f_{i_{\alpha_\star}}(\boldsymbol{x}).$$

Since $f_{i_{\alpha_\star}}$ is locally Lipschitz, the map $\boldsymbol{x} \mapsto u(\boldsymbol{y}_0, \boldsymbol{x})$ is locally Lipschitz in a neighborhood of $\boldsymbol{x}_0$. Because $\boldsymbol{x}_0 \in \mathcal{K}^\star \backslash \mathcal{N}(\boldsymbol{y}_0)$ was arbitrary, hence $\boldsymbol{x} \mapsto u(\boldsymbol{y}_0, \boldsymbol{x})$ is locally Lipschitz on $\mathcal{K}^\star \backslash \mathcal{N}(\boldsymbol{y}_0)$. □

An example to understand Theorem B.1. Consider the one-dimensional set-valued map

$$F(x) = \{x, -x\}, \qquad x \in \mathcal{K} \subset \mathbb{R},$$

with branches $f_1(x) = x$ and $f_2(x) = -x$. The two branches collapse at $x = 0$, and this collapse is not stable: in every neighborhood of 0, the branches are equal at 0 but distinct away from 0. Hence the stable domain is

$$\mathcal{K}^\star = \mathcal{K} \setminus \{0\}.$$

Now fix an initialization y_0. If $y_0 = 0$, then for every $x \in \mathcal{K}^\star$,

$$|y_0 - x| = |y_0 + x|,$$

so y_0 lies exactly on the Voronoi switching boundary for every stable input. Therefore

$$\mathcal{N}(0) = \mathcal{K}^\star.$$

Thus $y_0 = 0$ is a bad initialization: it is switching for a positive-measure, indeed full-measure, set of stable inputs. However, this bad initialization is exceptional. If $y_0 \neq 0$, then

$$|y_0 - x| = |y_0 + x|$$

implies

$$(y_0 - x)^2 = (y_0 + x)^2,$$

and hence

$$4y_0 x = 0.$$

Since $y_0 \neq 0$, this forces $x = 0$, which is not contained in $\mathcal{K}^\star$. Therefore

$$\mathcal{N}(y_0) = \varnothing \qquad \text{for every } y_0 \neq 0.$$

This example shows why the theorem must exclude a null set of initializations: some special initialization, here $y_0 = 0$, can be bad for many inputs, but the set of such initializations has Lebesgue measure zero in $Y = \mathbb{R}$. For almost every fixed initialization y_0, the induced selector is stable and locally Lipschitz on the whole stable domain $\mathcal{K}^\star$.

C Proof of Theorem 2.4

We now restate Theorem 2.4 to facilitate readability and proceed to prove it.

Theorem C.1. *Let $\mathcal{K}^\star \subset \mathbb{R}^d$ be bounded and measurable, and let $\mathcal{Y} = \mathbb{R}^m$. Let $f_1, \dots, f_n : \mathcal{K}^\star \to \mathcal{Y}$ be locally Lipschitz branch maps. Define the genuinely multi-branch region*

$$\mathcal{M} := \{\boldsymbol{x} \in \mathcal{K}^\star : |\mathcal{F}(\boldsymbol{x})| \geq 2\} = \{\boldsymbol{x} \in \mathcal{K}^\star : \exists i \neq j \text{ such that } f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x})\}.$$

Then, for every

$$a \in [0, |\mathcal{M}|],$$

there exists a measurable solution selector

$$u_a : \mathcal{K}^\star \to \mathcal{Y}, \qquad u_a(\boldsymbol{x}) \in \mathcal{F}(\boldsymbol{x}),$$

such that

$$|\mathrm{Disc}_{\mathcal{K}^\star}(u_a)| = a.$$

Here $\mathrm{Disc}_{\mathcal{K}^\star}(u_a)$ denotes the set of discontinuity points of u_a relative to the domain $\mathcal{K}^\star$.

Proof. For each pair $1 \leq i < j \leq n$, define

$$\mathcal{O}_{ij} := \{\boldsymbol{x} \in \mathcal{K}^\star : f_i(\boldsymbol{x}) \neq f_j(\boldsymbol{x})\}.$$

Since f_i and f_j are continuous relative to $\mathcal{K}^\star$, the set $\mathcal{O}_{ij}$ is relatively open in $\mathcal{K}^\star$. Moreover,

$$\mathcal{M} = \bigcup_{1 \leq i < j \leq n} \mathcal{O}_{ij}.$$

Thus $\mathcal{M}$ is relatively open in $\mathcal{K}^\star$ and is Lebesgue measurable.

Step 1: Scope of discontinuity. We first note that no selector can be discontinuous outside $\mathcal{M}$. Indeed, let $\boldsymbol{x}_0 \in \mathcal{K}^\star \setminus \mathcal{M}$. Then all branch values agree:

$$f_1(\boldsymbol{x}_0) = \cdots = f_n(\boldsymbol{x}_0) =: \boldsymbol{y}_0.$$

Let u be any selector satisfying $u(\boldsymbol{x}) \in \mathcal{F}(\boldsymbol{x})$. For any sequence $\boldsymbol{x}_k \to \boldsymbol{x}_0$ in $\mathcal{K}^\star$, there exists an index $i_k \in \{1, \dots, n\}$ such that

$$u(\boldsymbol{x}_k) = f_{i_k}(\boldsymbol{x}_k).$$

Hence

$$\|u(\boldsymbol{x}_k) - \boldsymbol{y}_0\| = \|f_{i_k}(\boldsymbol{x}_k) - f_{i_k}(\boldsymbol{x}_0)\| \leq \max_{1 \leq i \leq n} \|f_i(\boldsymbol{x}_k) - f_i(\boldsymbol{x}_0)\| \to 0.$$

Therefore u is continuous at $\boldsymbol{x}_0$, and consequently

$$\mathrm{Disc}_{\mathcal{K}^\star}(u) \subset \mathcal{M}$$

for every solution selector u.

Step 2: Construction of disjoint open cover. Now fix $a \in [0, |\mathcal{M}|]$. If $a = 0$, choose

$$u_a(\boldsymbol{x}) = f_1(\boldsymbol{x}).$$

Since f_1 is locally Lipschitz, it is continuous, and hence

$$|\mathrm{Disc}_{\mathcal{K}^\star}(u_a)| = 0.$$

It remains to consider $a > 0$. Since $\mathcal{M}$ is relatively open in $\mathcal{K}^\star$, for every $\boldsymbol{x} \in \mathcal{M}$ there exists a pair $i < j$ and an open ball $B(\boldsymbol{x}, r)$ such that

$$B(\boldsymbol{x}, r) \cap \mathcal{K}^\star \subset \mathcal{O}_{ij}.$$

The family of all such ambient balls is a Vitali cover of $\mathcal{M}$. By the Vitali covering theorem [8, Theorem 1.24], there exists a countable pairwise disjoint collection of open balls

$$B_1, B_2, \dots$$

such that, for each ℓ, there is a pair (i_ℓ, j_ℓ) with

$$G_\ell := B_\ell \cap \mathcal{K}^\star \subset \mathcal{O}_{i_\ell j_\ell},$$

and

$$\left|\mathcal{M} \setminus \bigcup_{\ell=1}^{\infty} G_\ell\right| = 0. \tag{7}$$

Since the balls B_ℓ are pairwise disjoint, the sets G_ℓ are pairwise disjoint. Therefore

$$\sum_{\ell=1}^{\infty} |G_\ell| = \left|\bigcup_{\ell=1}^{\infty} G_\ell\right| = |\mathcal{M}|.$$

We now construct a relatively open set $G_a \subset \mathcal{M}$ with $|G_a| = a$.

If $0 < a < |\mathcal{M}|$, define partial sums

$$S_N := \sum_{\ell=1}^{N} |G_\ell|, \qquad S_0 := 0.$$

Since $S_N \to |\mathcal{M}| > a$, there exists a first index N such that $S_N > a$. Hence

$$S_{N-1} \le a < S_N.$$

Set $r := a - S_{N-1}$. Then $0 \le r < |G_N|$. If $r = 0$, just define

$$G_a := \bigcup_{\ell=1}^{N-1} G_\ell.$$

If $r > 0$, let

$$H_t := \{\boldsymbol{x} \in \mathbb{R}^d : x_1 < t\}.$$

The function $\phi(t) := |G_N \cap H_t|$ is continuous, satisfies $\phi(t) \to 0$ as $t \to -\infty$, and satisfies $\phi(t) \to |G_N|$ as $t \to +\infty$. Hence there exists t_0 such that $|G_N \cap H_{t_0}| = r$. Now define

$$G_a := \left(\bigcup_{\ell=1}^{N-1} G_\ell\right) \cup (G_N \cap H_{t_0}).$$

Then $G_a \subset \mathcal{M}$, G_a is relatively open in $\mathcal{K}^\star$, and

$$|G_a| = a.$$

Moreover, its relative boundary satisfies

$$|\partial_{\mathcal{K}^\star} G_a| = 0,$$

because $\partial_{\mathcal{K}^\star} G_a$ is contained in a finite union of sets of the form $\mathcal{K}^\star \cap \partial B_\ell$ and possibly one hyperplane slice $\mathcal{K}^\star \cap \{x_1 = t_0\}$, all of which have d-dimensional Lebesgue measure zero.

If $a = |\mathcal{M}|$, define

$$G_a := \bigcup_{\ell=1}^{\infty} G_\ell.$$

Then $G_a \subset \mathcal{M}$, G_a is relatively open in $\mathcal{K}^\star$, and

$$|\mathcal{M} \setminus G_a| = 0, \qquad |G_a| = |\mathcal{M}| = a.$$

In this case, $\partial_{\mathcal{K}^\star} G_a \cap \mathcal{M} \subset \mathcal{M} \setminus G_a$ because G_a is relatively open and $G_a \subset \mathcal{M}$. Then by (7), we have

$$|\partial_{\mathcal{K}^\star} G_a \cap \mathcal{M}| = 0.$$

Step 3: Construction of solution selector. We next define the selector u_a. The set G_a is a finite or countable union of disjoint selected pieces. Each selected piece E is of one of the forms

$$E = G_\ell \qquad \text{or} \qquad E = G_N \cap H_{t_0},$$

and is contained in some $\mathcal{O}_{i_E j_E}$ for a pair $i_E < j_E$. For each such selected piece E, choose a countable dense subset $Q_E \subset E$ with respect to the relative topology of E. Define

$$u_a(\boldsymbol{x}) = \begin{cases} f_{i_E}(\boldsymbol{x}), & \boldsymbol{x} \in Q_E, \\ f_{j_E}(\boldsymbol{x}), & \boldsymbol{x} \in E \setminus Q_E, \end{cases} \qquad \boldsymbol{x} \in E.$$

Outside G_a, define

$$u_a(\boldsymbol{x}) = f_1(\boldsymbol{x}).$$

Because the selected pieces are measurable and at most countably many, and because each Q_E is countable, the map u_a is measurable. Moreover,

$$u_a(\boldsymbol{x}) \in \{f_1(\boldsymbol{x}), \dots, f_n(\boldsymbol{x})\} = \mathcal{F}(\boldsymbol{x})$$

for every $\boldsymbol{x} \in \mathcal{K}^\star$.

Step 4: Discontinuity set. We now estimate the discontinuity set of u_a.

First, let E be one selected piece. Since E is measurable, the Lebesgue density theorem [8, Theorem 1.35] implies that almost every point of E is a density point of E. Let $\boldsymbol{x}_0$ be such a density point. Because Q_E is dense in E, there exists a sequence

$$\boldsymbol{x}_k \in Q_E, \qquad \boldsymbol{x}_k \to \boldsymbol{x}_0.$$

Also, since Q_E is countable and $\boldsymbol{x}_0$ is a density point of E, every sufficiently small relative neighborhood of $\boldsymbol{x}_0$ contains points of $E \setminus Q_E$. Hence there exists a sequence

$$\boldsymbol{z}_k \in E \setminus Q_E, \qquad \boldsymbol{z}_k \to \boldsymbol{x}_0.$$

Along these two sequences,

$$u_a(\boldsymbol{x}_k) = f_{i_E}(\boldsymbol{x}_k) \to f_{i_E}(\boldsymbol{x}_0),$$

while

$$u_a(\boldsymbol{z}_k) = f_{j_E}(\boldsymbol{z}_k) \to f_{j_E}(\boldsymbol{x}_0).$$

Since $E \subset \mathcal{O}_{i_E j_E}$, we have

$$f_{i_E}(\boldsymbol{x}_0) \neq f_{j_E}(\boldsymbol{x}_0).$$

Thus u_a is discontinuous at $\boldsymbol{x}_0$. Therefore u_a is discontinuous at almost every point of every selected piece E, and hence $|G_a \setminus \mathrm{Disc}_{\mathcal{K}^\star}(u_a)| = 0$. Consequently,

$$|\mathrm{Disc}_{\mathcal{K}^\star}(u_a)| \geq |G_a| = a.$$

Conversely, we already showed that no selector can be discontinuous outside $\mathcal{M}$. Moreover, if

$$\boldsymbol{x}_0 \in \mathcal{M} \setminus (G_a \cup \partial_{\mathcal{K}^\star} G_a),$$

then $\boldsymbol{x}_0$ has a relative neighborhood in $\mathcal{K}^\star$ disjoint from G_a. On that neighborhood, $u_a = f_1$, so u_a is continuous at $\boldsymbol{x}_0$. Therefore

$$\mathrm{Disc}_{\mathcal{K}^\star}(u_a) \subset G_a \cup (\partial_{\mathcal{K}^\star} G_a \cap \mathcal{M}).$$

If $0 < a < |\mathcal{M}|$, then $|\partial_{\mathcal{K}^\star} G_a| = 0$, so

$$|\mathrm{Disc}_{\mathcal{K}^\star}(u_a)| \leq |G_a| = a.$$

Together with the lower bound, this gives

$$|\mathrm{Disc}_{\mathcal{K}^\star}(u_a)| = a.$$

If $a = |\mathcal{M}|$, then $|\partial_{\mathcal{K}^\star} G_a \cap \mathcal{M}| = 0$, so again

$$|\mathrm{Disc}_{\mathcal{K}^\star}(u_a)| = |G_a| = |\mathcal{M}| = a.$$

This completes the proof. □

D Experiment Details for Toy Examples

This appendix records the implementation details for the toy-example experiments in Section 2. In particular, it specifies the parameterization of the weight-tied dynamics, the construction of the training/validation/test data, and the staged training procedure used in the numerical verifications for Theorems 2.2–2.4.

Model. In all toy-example experiments, we parameterize the weight-tied dynamics g_θ by a two-layer MLP with embedding width 256. We use the same parameterization across the two toy examples. The model is trained in the unrolled form used throughout the paper, with shared parameters across all iterations.

Data. The first toy example **(Algebraic)** is constructed from the two branches of the map $x \mapsto \{\pm 1/|x|\}$. Its training inputs are sampled on a geometric grid

$$\mathcal{X}_{\text{train}}^{(1)} = \{\pm(1+\delta)^s\},$$

with $\delta = 0.1$, truncated to the range determined by $x_{\max} = 10$ and $y_{\max} = 10$; in particular, the grid excludes $x = 0$. The validation inputs are generated independently on a different geometric grid with step size 0.2, so the validation points do not coincide with the training points. For visualization and testing, we use denser grids that are again generated independently from the training grid.

The second toy example **(Double-well)** is constructed from the two stable branches of the double-well equation

$$4y^3 - 4y + x = 0.$$

Its training inputs are sampled uniformly on $[-2.5, 2.5]$ with 500 points, while the validation inputs are sampled independently on the same interval with 128 points. For visualization and testing, we again use denser independent grids. Therefore, in both toy examples, the train and test data are independent.

Training. We largely follow Algorithm 1, with one practical modification: we use staged training with increasing unroll lengths. For the first toy example, the three stages use unroll lengths

$$[10, 20, 30],$$

and for the second toy example, they use

$$[3, 6, 9].$$

All runs use learning rate 10^{-4}, 5000 epochs per stage, and 3 stages in total. The initialization distribution is

$$y_0 \sim \text{Unif}[-5, 5],$$

and for each input we sample 10 initializations during training. In the supervised toy-example experiments, when multiple target branches are available for the same input, we assign each sampled initialization y_0 to the branch that is closest to y_0 in Euclidean distance.

Optimization and model selection. We use the same optimizer settings across the two toy examples, together with the stage-wise continuation described above. Within each stage, model selection is based on the validation loss, and the best checkpoint in that stage is used to initialize the next stage. This produces the final supervised checkpoints used in the toy-example visualizations.

Branch selector construction. For the branch-selection experiments in Section 2.2, we derive single-valued supervised datasets from the original two-branch toy examples by imposing an interval-wise alternating selector. Specifically, we partition the input domain into N_{int} intervals and assign one branch to each interval, with adjacent intervals always selecting different branches. In the first toy example, the partition is uniform in $\log|x|$, so that branch switching is distributed evenly across scales near the singular region. In the second toy example, the partition is uniform in x. We consider interval counts $N_{\text{int}} \in \{2, 4, 8, 16, 32, 64\}$. This construction keeps the underlying solution set unchanged while producing increasingly irregular single-valued targets as N_{int} grows, thereby isolating the effect of arbitrary human branch selection from the effect of the original multi-solution geometry.

Hardware. All neural training and evaluation runs in this paper were performed on a single workstation with four NVIDIA Quadro RTX 6000 GPUs (24 GB each).

E Ising-GNN Experiments: Details and Addtional Results

E.1 Experiment Details

This subsection provides more experiment details as a supplement to the main text.

Dataset construction. Each graph is a triangular lattice generated by python package `networkx`, with all couplings antiferromagnetic: $J_{ij} \sim -\mathrm{Unif}[0.5, 1.5]$. This choice makes every edge prefer opposite spins, while the triangular geometry prevents all constraints from being satisfied simultaneously. The result is a geometrically frustrated family of Ising instances with many competing low-energy patterns. The stored dataset contains 2500 training graphs and 500 held-out test graphs. The graph sizes vary substantially: the number of nodes ranges from 6 to 55 with mean 27.37, and the number of undirected edges ranges from 9 to 135 with mean about 62.08.

For supervised comparisons, we use Gurobi on (4) to compute spin label $\boldsymbol{s} \in \{-1, +1\}^{|\mathcal{V}|}$. This label is only one branch among many physically admissible low-energy branches.

Graph models. We largely adopt the implicit GNN architecture from [27], unrolling the dynamics for $T = 10$ iterations. Node states and edge couplings J_{ij} are encoded by a two-layer ReLU MLP of width 64. The update uses three weighted 64-to-64 GraphConv layers with sum aggregation, no degree normalization, and ReLU activations, followed by a two-layer spin readout with a final $\tanh$. The output range $(-1, 1)$ is therefore enforced smoothly by $\tanh$, not by clipping. However, when training the supervised baseline against exact Gurobi labels, we observed that the standard update was highly unstable and struggled to minimize the loss. To achieve reliable convergence, we modify the supervised model to use a damped recurrent update:

$$\boldsymbol{s}_{t+1} = (1 - \alpha)\boldsymbol{s}_t + \alpha\, g_{\mathrm{GNN}}(\boldsymbol{s}_t, \mathcal{G}), \qquad \alpha \in (0, 1].$$

This damping mitigates the training instability associated with forcing the weight-tied dynamics to fit a single target branch. We treat α as a hyperparameter and tune it over the grid $\{0.01, 0.05, 0.1, 0.5, 1\}$.

Other details. All models use hidden width $h = 64$. For the original small-scale dataset, models are trained with Adam, learning rate 10^{-3}, batch size 32, and a maximum of $10{,}000$ epochs. The energy-trained recurrent GNN is evaluated with 20 distinct random initializations, whereas the vanilla GNNs have no initialization-dependent dynamics and produce one deterministic output per graph. For the single-target supervised recurrent GNN, we use a fixed zero initialization, consistent with its intended role of predicting one solver-selected target; its identical repeated outputs therefore reflect this evaluation protocol, rather than establishing uniqueness of its attractor under arbitrary initializations.

E.2 Evaluation Metrics and Reference Pools

Count and spread. With $K = 20$ outputs per graph, we report the count and spread as follows. For two rounded configurations $\boldsymbol{s}, \boldsymbol{r} \in \{-1, +1\}^N$, define the normalized, global-sign-adjusted Hamming distance

$$d_{\mathbb{Z}_2}(\boldsymbol{s}, \boldsymbol{r}) = \min_{\eta \in \{-1,+1\}} \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}\{s_j \neq \eta r_j\} \in [0, 1/2]. \tag{8}$$

For outputs $\bar{\boldsymbol{s}}^{(1)}, \ldots, \bar{\boldsymbol{s}}^{(K)}$, #Sols is the number of distinct equivalence classes under $\boldsymbol{s} \sim -\boldsymbol{s}$, using exact equality of rounded spins and no energy filter. It therefore counts configurations, not certified ground states. Spread is

$$\mathrm{Spread} = \frac{2}{K(K-1)} \sum_{1 \leq a < b \leq K} d_{\mathbb{Z}_2}(\bar{\boldsymbol{s}}^{(a)}, \bar{\boldsymbol{s}}^{(b)}). \tag{9}$$

All 20 outputs enter this average, including duplicates; we do not first deduplicate the set. Consequently, repeated copies of one configuration, even with global sign flips, have zero Spread.

Reference pools and recall. The archived Gurobi solution pools use `PoolSearchMode=2`, `PoolSolutions=200`, `PoolGap=0.05`, and a 60-second per-graph time limit. Returned solutions are processed in solver objective order and greedily filtered for separation modulo global sign: a solution is retained only if its distance in (8) from every retained representative is at least 0.05, with at most 64 representatives retained. The retained configurations form finite low-energy reference pools, not enumerations of all optimal solutions. In the 200-graph evaluation cohort their sizes range from 1 to 60, with mean 19.79. For the reference pool $\mathcal{R}_{\mathcal{G}}$, the graphwise recall is

$$\mathrm{Recall@10\%} = \frac{1}{|\mathcal{R}_{\mathcal{G}}|} \sum_{\boldsymbol{r} \in \mathcal{R}_{\mathcal{G}}} \mathbf{1}\left\{ \min_{1 \leq k \leq K} d_{\mathbb{Z}_2}(\bar{\boldsymbol{s}}^{(k)}, \boldsymbol{r}) < 0.1 \right\}. \tag{10}$$

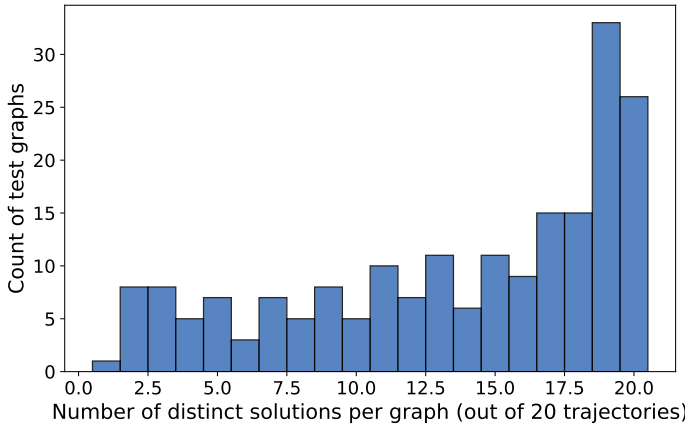

Figure 5: Histogram of the number of distinct rounded configurations modulo global sign per test graph for the weight-tied GNN dynamic. Evaluation uses 200 test graphs and $k = 20$ initializations per graph.

E.3 Additional Results: Distribution of Configuration Counts

While Table 2 summarizes solution counts, spread, and reference-pool coverage, Figure 5 gives the distribution of the exact rounded-configuration counts modulo global sign. The broad multiplicity is not driven solely by a few outliers. These counts have no energy filter and should be interpreted jointly with the energy and coverage metrics, rather than as counts of certified optimal solutions.

E.4 Additional Results: Scaling up the Graph Size

This subsection reports the larger-scale Ising experiments used to assess how the models behave as the graph size increases. All datasets in this subsection use the same triangular antiferromagnetic construction as the base dataset in the main text. Model, training, and energy-evaluation protocols are also the same. The only difference is the graph size.

Dataset levels. We generated eight dataset levels, denoted $L1$ through $L8$, by increasing the target graph-size range in the same data-generation script used for the small dataset. Each level contains 2500 training graphs and 500 test graphs. Table 5 reports the realized graph-size statistics.

Interpretation. Table 6 shows that energy-based training scales much more reliably than Gurobi-label fitting. Across all levels $L1$–$L8$, the energy-based models remain trainable on increasingly large frustrated graphs, and the recurrent GNN consistently gives the lowest rounded-spin energies; even at $L8$, where test graphs have about 1.2×10^4 nodes and 3.6×10^4 edges, its test energy remains around -1.16.

In contrast, Gurobi-label training is available only for $L1$–$L5$ due to labeling cost, and its performance degrades rapidly as graph size grows. This is consistent with the main-text interpretation: fitting one selected branch becomes increasingly brittle in large multi-solution landscapes, whereas energy-based training only asks the model to find any low-energy valid branch.

E.5 Initialization Sensitivity and Trajectory Budget

We vary the graph-level bimodal Gaussian initialization $\boldsymbol{s}_0 = \sigma\xi + bs_g$, where $\xi \sim \mathcal{N}(0, I)$ and one sign $s_g \in \{-1, +1\}$ is shared across all nodes of a graph. The default is $(\sigma, b) = (1, 2)$. Table 7 separates within-mode dispersion from mode separation. Broader dispersion increases the number of outputs, while a larger mode separation reduces exploration and slightly improves average energy. Convergence times change little: the first iteration satisfying $\|\boldsymbol{s}_{t+1} - \boldsymbol{s}_t\| < 10^{-4}$ averages about 18 steps throughout the sweep.

Level	Split	# Graphs	Node Range	Mean Nodes	Edge Range	Mean Edges
L1	Train	2500	66–120	86.64	165–315	222.05
L1	Test	500	66–120	86.05	165–315	220.40
L2	Train	2500	136–253	186.28	360–693	502.65
L2	Test	500	136–253	186.40	360–693	503.01
L3	Train	2500	276–528	388.27	759–1488	1083.06
L3	Test	500	276–496	387.77	759–1395	1081.62
L4	Train	2500	561–990	751.35	1584–2838	2139.65
L4	Test	500	561–946	756.40	1584–2709	2154.40
L5	Train	2500	990–1953	1476.19	2838–5673	4267.92
L5	Test	500	990–1953	1474.49	2838–5673	4262.84
L6	Train	2500	2016–3916	2979.14	5859–11484	8708.41
L6	Test	500	2016–3916	2989.31	5859–11484	8738.53
L7	Train	2500	4005–7875	5961.49	11748–23250	17559.91
L7	Test	500	4005–7875	5876.83	11748–23250	17308.27
L8	Train	2500	8001–15931	11975.45	23625–47259	35465.78
L8	Test	500	8001–15931	12218.36	23625–47259	36189.83

Table 5: Realized graph-size statistics for the eight triangular antiferromagnetic dataset levels used in the scaling experiments. Edge counts are undirected counts.

Increasing the trajectory budget also improves coverage of the finite reference pool: Recall@10% increases from 0.743 to 0.891 when the trajectory budget increases from 20 to 50. Average energy remains near -0.98, and pairwise spread remains near 0.287. Thus, additional starts recover more reference configurations without materially changing average energy or separation. Recall uses the Hamming-distance threshold in (10), not a 10% energy gap.

E.6 Multi-Solution Learning Baselines

Fixed-output multi-head prediction. We generate four distinct labels within 5% of the Gurobi optimum per graph and evaluate on 384 test graphs with at least four such labels. Every method returns four candidates. We compare fixed canonical label order, permutation-invariant Hungarian matching, random label order, and an energy-only four-head control. Table 8 shows that Hungarian matching improves over a fixed ordering, but the energy-only recurrent model attains substantially lower energies. The energy-only four-head control nearly collapses to one output. Counts alone do not establish quality: distinct rounded outputs need not be low-energy solutions.

Proximal operator learning. We compare with POL [24] using the same GNN backbone, graph-coherent initialization, and one-dimensional spin parameterization, with 20 starts on 200 test graphs. This matches the parameterization of our Ising model rather than using POL's original rank-two MaxCut relaxation. Smaller proximal weight λ prioritizes energy minimization, whereas larger values penalize particle movement more strongly. Table 9 shows that larger λ increases the count of distinct outputs at the cost of substantially worse energy. Under this matched parameterization, our model improves both energy and count over the quality-oriented $\lambda = 0$ setting.

E.7 Robustness to Spin-Flip Perturbations

We retain the original checkpoints without retraining or noise augmentation. These experiments test inference-time robustness of the zero-temperature solver; they are not calibrated finite-temperature Gibbs sampling. For persistent perturbations, each deterministic update $\hat{\boldsymbol{s}}_{t+1} = g_\theta(\boldsymbol{s}_t, \mathcal{G})$ is followed by $s_{t+1,i} = \xi_{t,i}\hat{s}_{t+1,i}$, where $\Pr(\xi_{t,i} = -1) = p$, independently across nodes, trajectories, and steps. For a one-time perturbation, we flip each component of a noise-free terminal state once and then apply 10 deterministic recovery steps. Tables 10–11 report both settings.

The energy-trained model maintains nearly unchanged energy through $p = 0.1$ under persistent flips, and recovers low energy after every tested one-time perturbation. The single-branch baseline does

Table 6: Ising results across dataset levels $L1$–$L8$. Supervised results are available only for $L1$–$L5$, since Gurobi labeling takes > 50 hrs for L6 labeling.

Dataset Level	Objective	Model	Train energy	Test energy
L1	Gurobi labeled	Vanilla GNN	0.4413 ± 0.5470	0.4474 ± 0.5096
L1	Gurobi labeled	Recurrent GNN	-0.1735 ± 0.2357	-0.1634 ± 0.2294
L1	Energy based	Vanilla GNN	-0.8761 ± 0.0751	-0.8765 ± 0.0701
L1	Energy based	Recurrent GNN	-1.0612 ± 0.0407	-1.0638 ± 0.0422
L2	Gurobi labeled	Vanilla GNN	0.7460 ± 0.3787	0.7644 ± 0.4004
L2	Gurobi labeled	Recurrent GNN	0.3858 ± 0.2314	0.3875 ± 0.2295
L2	Energy based	Vanilla GNN	-0.9003 ± 0.0532	-0.9017 ± 0.0497
L2	Energy based	Recurrent GNN	-1.1185 ± 0.0276	-1.1165 ± 0.0291
L3	Gurobi labeled	Vanilla GNN	2.2180 ± 0.2777	2.2055 ± 0.2773
L3	Gurobi labeled	Recurrent GNN	1.0121 ± 0.2629	0.9885 ± 0.2653
L3	Energy based	Vanilla GNN	-0.9181 ± 0.0389	-0.9167 ± 0.0390
L3	Energy based	Recurrent GNN	-1.1458 ± 0.0189	-1.1457 ± 0.0179
L4	Gurobi labeled	Vanilla GNN	2.5518 ± 0.1508	2.5516 ± 0.1463
L4	Gurobi labeled	Recurrent GNN	1.7608 ± 0.1986	1.7606 ± 0.1809
L4	Energy based	Vanilla GNN	-0.9352 ± 0.0259	-0.9356 ± 0.0264
L4	Energy based	Recurrent GNN	-1.1504 ± 0.0139	-1.1508 ± 0.0139
L5	Gurobi labeled	Vanilla GNN	2.5416 ± 0.0850	2.5427 ± 0.0817
L5	Gurobi labeled	Recurrent GNN	2.4534 ± 0.0923	2.4482 ± 0.0877
L5	Energy based	Vanilla GNN	-0.9448 ± 0.0199	-0.9435 ± 0.0202
L5	Energy based	Recurrent GNN	-1.1464 ± 0.0101	-1.1461 ± 0.0096
L6	Gurobi labeled	Vanilla GNN	–	–
L6	Gurobi labeled	Recurrent GNN	–	–
L6	Energy based	Vanilla GNN	-0.9540 ± 0.0139	-0.9531 ± 0.0137
L6	Energy based	Recurrent GNN	-1.1554 ± 0.0071	-1.1554 ± 0.0071
L7	Gurobi labeled	Vanilla GNN	–	–
L7	Gurobi labeled	Recurrent GNN	–	–
L7	Energy based	Vanilla GNN	-0.9583 ± 0.0097	-0.9577 ± 0.0100
L7	Energy based	Recurrent GNN	-1.1695 ± 0.0080	-1.1698 ± 0.0082
L8	Gurobi labeled	Vanilla GNN	–	–
L8	Gurobi labeled	Recurrent GNN	–	–
L8	Energy based	Vanilla GNN	-0.9616 ± 0.0067	-0.9617 ± 0.0068
L8	Energy based	Recurrent GNN	-1.1605 ± 0.0075	-1.1605 ± 0.0074

Table 7: Ising initialization sensitivity, using 20 trajectories per graph. Entries are means. Energy is per node.

Sweep	Value	Energy	#Sols	Iters
$\sigma, b = 2$	0.2	-0.9903	8.16	18.1
	0.5	-0.9900	12.12	18.3
	1.0	-0.9857	13.63	17.7
	2.0	-0.9785	14.36	17.8
$b, \sigma = 1$	1	-0.9791	14.18	17.9
	2	-0.9857	13.63	17.7
	4	-0.9914	12.00	18.4
	8	-0.9942	8.40	18.2

Table 8: Four-candidate Ising comparison. Energy is a per-node mean; #Sols is the mean number of distinct rounded outputs.

Model	Energy	#Sols
Energy-based recurrent GNN	-1.0059	3.802
Canonical label order	-0.2421	3.716
Hungarian matching	-0.3640	3.984
Random label order	0.2406	3.964
Energy-only four-head control	-0.7180	1.005
Gurobi optimum	-1.0862	–

Table 9: Matched-parameterization POL comparison. Entries are means; #Sols has no energy filter and must be interpreted together with energy.

Model	Proximal λ	Energy	#Sols
Energy-based recurrent GNN	–	-0.983	13.63
POL	0	-0.964	8.30
POL	0.1	-0.947	10.26
POL	1	-0.805	18.60
POL	10	-0.519	19.40
Gurobi optimum	–	-1.063	–

Table 10: Persistent spin flips after every recurrent update. Test energy is reported as mean $\pm$ standard deviation. The Gurobi optimum on this test set is -1.063 ± 0.107.

p	Energy-based recurrent GNN	Gurobi-label recurrent GNN
0.000	-0.983 ± 0.101	-0.092 ± 0.361
0.010	-0.986 ± 0.112	-0.044 ± 0.364
0.020	-0.987 ± 0.112	0.007 ± 0.377
0.050	-0.990 ± 0.112	0.110 ± 0.402
0.100	-0.986 ± 0.116	0.143 ± 0.398
0.200	-0.959 ± 0.127	0.071 ± 0.365
0.300	-0.919 ± 0.135	0.032 ± 0.351
0.400	-0.892 ± 0.139	0.011 ± 0.341
0.500	-0.878 ± 0.137	0.005 ± 0.337

Table 11: Energy immediately after one-time spin flips and after 10 recovery steps (mean $\pm$ standard deviation).

	Energy-based recurrent GNN		Gurobi-label recurrent GNN	
p	Perturbed	Recovered	Perturbed	Recovered
0.000	-0.983 ± 0.101	–	-0.092 ± 0.361	–
0.010	-0.941 ± 0.152	-0.985 ± 0.114	-0.088 ± 0.358	0.176 ± 0.479
0.020	-0.904 ± 0.180	-0.986 ± 0.113	-0.084 ± 0.359	0.187 ± 0.480
0.050	-0.793 ± 0.232	-0.988 ± 0.113	-0.074 ± 0.350	0.221 ± 0.485
0.100	-0.630 ± 0.283	-0.990 ± 0.113	-0.060 ± 0.350	0.279 ± 0.500
0.200	-0.350 ± 0.327	-0.993 ± 0.113	-0.034 ± 0.337	0.403 ± 0.519
0.300	-0.148 ± 0.341	-0.991 ± 0.114	-0.009 ± 0.333	0.546 ± 0.540
0.500	0.001 ± 0.332	-0.977 ± 0.118	0.001 ± 0.332	0.872 ± 0.571

not exhibit the same recovery behavior. These observations are empirical robustness results, not guarantees under stochastic dynamics. Noise strength is maximal at $p = 0.5$: for $p > 0.5$, a flip is equivalent in distribution to global reversal followed by flips of probability $1 - p$, and the Ising energy is invariant under global reversal.

E.8 Comparisons with Non-Neural Solvers

Metropolis. We initialize 20 independent Metropolis chains from the same spin states as the recurrent GNN on held-out triangular antiferromagnetic graphs. Table 12 compares equal wall-clock time (the full GNN rollout time) and time to the common energy-gap target 0.1, reached by both methods at every size. The GNN gives lower energy at the same time budget. It reaches the target faster from the second size onward, with comparable spread on the larger instances; Metropolis is slightly faster at the smallest size.

Table 12: Metropolis comparison. Paired entries are GNN / Metropolis. Speedup is Metropolis time divided by GNN time to the common energy-gap target; all energies are per node.

Mean nodes	Equal-time budget (s)	Best energy at equal time	Time to gap 0.1 (s)	Speedup	Spread at gap 0.1
86	0.407	$-1.149/-1.129$	$0.157/0.136$	$0.9\times$	$0.386/0.426$
186	0.427	$-1.175/-1.091$	$0.216/0.423$	$2.0\times$	$0.417/0.447$
388	0.454	$-1.180/-1.019$	$0.221/1.097$	$5.0\times$	$0.432/0.461$
756	0.515	$-1.182/-0.866$	$0.236/2.628$	$11.1\times$	$0.463/0.472$
1,474	0.635	$-1.185/-0.541$	$0.292/6.008$	$20.6\times$	$0.478/0.480$
2,989	0.939	$-1.192/-0.242$	$0.317/5.423$	$17.1\times$	$0.489/0.490$
5,877	1.534	$-1.187/-0.122$	$0.400/10.290$	$25.7\times$	$0.494/0.494$
12,218	2.365	$-1.189/-0.030$	$0.583/20.407$	$35.0\times$	$0.497/0.497$

Gurobi. Table 13 compares Gurobi 13.0.0 using 40 CPU cores with a recurrent GNN using one GPU. GNN runtime includes 10 iterations and all 20 initializations. These are joint speed–quality measurements: Gurobi attains lower energies, whereas GNN inference is faster on larger instances. They are not timings to matched energy or hardware budgets, and the evaluation cohorts differ from those in the Metropolis comparison.

Table 13: Gurobi versus recurrent GNN: mean runtime and energy per node on the size-scaling evaluation cohorts.

Mean nodes	Gurobi time (s)	GNN time (s)	Gurobi energy	GNN energy
88	0.15	0.226	-1.18	-1.08
719	1.13	0.228	-1.25	-1.15
2,884	30.99	0.224	-1.27	-1.17
12,252	289.81	0.283	-1.28	-1.17

F Allen-Cahn Experiments: More Details and Additional Results

F.1 Experiment Details

This subsection provides more experiment details as a supplement to the main text.

Dataset construction. All experiments in this section use the fixed interface parameter $\varepsilon = 0.01$. Each forcing field is generated on an $N \times N$ periodic grid over $\Omega = (0, 2\pi)^2$. Unless otherwise stated, $N = 64$. For each instance, we sample a periodic Gaussian random field (GRF). In particular, we draw independent Gaussian Fourier noise, impose the conjugate-symmetry condition needed to obtain a real-valued spatial field, set the zero mode to zero, and multiply each Fourier mode k by the filter

$$\exp(-\|k\|_2^2/(2\ell^2))(1 + \|k\|_2^2)^{-\alpha/2},$$

where the smoothness and spectral-scale parameters are sampled independently as $\alpha \sim \mathrm{Unif}[1.4, 2.8]$ and $\ell \sim \mathrm{Unif}[8.0, 14.0]$. After applying the inverse FFT, we subtract the empirical spatial mean and

divide by the empirical spatial standard deviation, obtaining a normalized field $\tilde{f}$. The final forcing is $f = A\tilde{f} + m$, where $A \sim \mathrm{Unif}[0.08, 0.20]$ controls the amplitude and $m \sim \mathrm{Unif}[-0.03, 0.03]$ controls the mean shift. The dataset contains 1000 training instances, 1000 validation instances, and 1000 held-out test instances, generated independently using the same procedure.

Model structure. We largely follow the implicit FNO architecture of [28]. In that framework, a neural operator is used as a shared update rule for a steady-state solver rather than as a one-shot feed-forward predictor. At each iteration, the model takes the current state estimate together with the fixed PDE input, applies an FNO-based update block, and reuses the same parameters across all unrolled iterations. This gives a weight-tied dynamics whose equilibrium is interpreted as the predicted steady solution. In our Allen–Cahn experiments, the input is the forcing field f, and the recurrent state is the phase-field estimate u_t. We use the residual update

$$u_{t+1} = g_\theta(u_t, f) = u_t + \eta \Phi_\theta(u_t, f), \qquad \eta = 0.2,$$

where Φ_θ is an FNO-based state-dependent update field shared across all iterations, following [28]. In particular, we choose such a block which has depth one and consists of one spectral convolution, one pointwise 1×1 convolution, additive forcing injection, and a GELU nonlinearity. For the main 64×64 experiments, all models use width 32, 32×32 retained Fourier modes, one recurrent update block, one internal layer inside that block, and one forcing input channel. The current state u_t and forcing f are lifted to 32 channels. The spectral and parallel 1×1 convolution paths are combined with additive forcing injection and GELU; the result is projected through a $32 \to 128 \to 1$ output head.

The initial recurrent state u_0 is sampled from a GRF distribution with scale 0.5, smoothness parameter $\alpha = 3.0$, and spectral scale $\ell = 8.0$. During training, each forcing instance is replicated with 8 independent GRF initializations, while validation uses 16 independent GRF initializations per instance. The final evaluation in Table 4 uses 20 initializations and 50 iterations per instance. This trains the same weight-tied operator to evolve from multiple candidate starting states for the same physical input.

Supervised label making. For supervised training, we construct labels by solving the steady Allen–Cahn equation with a classical periodic Implicit-Explicit (IMEX) solver [6] on the same spatial grid as the dataset. This is a commonly used method to solve Allen–Cahn. For each forcing instance, the classical solver is initialized from a GRF draw generated by the same initialization procedure used in model training, namely with scale 0.5, smoothness parameter $\alpha = 3.0$, and spectral scale $\ell = 8.0$. The solver tolerance is fixed at 10^{-5}. The resulting steady solutions are used as the supervised targets in all supervised Allen–Cahn experiments reported in the main text and appendix.

Solution counting. To count distinct output clusters, we cluster the M raw final states before IMEX refinement based on their pairwise root mean squared error (RMSE). Outputs are processed greedily: an output creates a new cluster only if its RMSE from every existing representative is at least a resolution-specific threshold; otherwise it joins an existing cluster. This threshold is calibrated on the energy-only model so that its outputs form one cluster for most test instances; it is 0.8 for the main 64×64 experiments. We report the mean and standard deviation of the cluster counts across test instances.

Diversity evaluation. The reported Diversity is the mean pairwise RMSE of the $M = 20$ final states,

$$D(\{\hat{u}_k\}_{k=1}^M) = \frac{1}{M(M-1)} \sum_{i \neq j} \frac{\|\hat{u}_i - \hat{u}_j\|_2}{N}.$$

Unlike the squared-distance training regularizer $\mathcal{D}_{\mathrm{train}}$, this evaluation metric uses RMSE. Refined diversity applies the same metric after refining each final state with the high-accuracy periodic IMEX solver. Both metrics are reported as mean $\pm$ standard deviation across test instances. The pure IMEX baseline uses the same 20 initial states; its refined diversity is measured after high-accuracy IMEX convergence. Residual MSE and energy in Table 4 are evaluated before refinement: for each trajectory, we record the best value over its 50 iterations, compute the mean and standard deviation across the 20 trajectories within each instance, and then average these two statistics separately across test instances.

Training. We largely follow Algorithm 2, but use a fixed six-stage curriculum in the unroll length. All models are trained for 120 epochs with the Adam optimizer, weight decay 10^{-4}, batch size 16, and initial learning rate 10^{-3}. The six curriculum stages start at epochs $0, 20, 40, 60, 80, 100$, with training unroll lengths $2, 4, 8, 12, 16, 20$, respectively. The learning rate is kept constant within each stage and decreased across stages according to the schedule $[10^{-3}, 8\times10^{-4}, 6\times10^{-4}, 4\times10^{-4}, 3\times10^{-4}, 2\times10^{-4}]$. During training, each forcing instance is paired with 8 independent GRF initial states. During validation, we use 16 independent GRF initial states per forcing instance. At each epoch, the model is evaluated on the validation set using the current curriculum stage, with the validation rollout length set to twice the training rollout length at that stage. We select the checkpoint with the smallest validation residual and report its final test performance using 20 initial states and 50 iterations per instance. For supervised runs, we keep the same optimizer, epoch budget, learning-rate schedule, and unroll curriculum, but replace the unsupervised objective by a pure supervised regression loss to the chosen labeled branch.

Diversity-weight curriculum. The diversity coefficient is stage-dependent rather than fixed throughout training. We use the same six rollout stages as the unroll-length curriculum, with rollout lengths $2, 4, 8, 12, 16, 20$ respectively. For each diversity-regularized run, we start with a relatively large diversity weight to encourage separation among trajectories from different initial states, and then decay the weight so that later stages focus more on physical refinement. In the last stage the diversity weight is set to zero, allowing the model to improve the Allen–Cahn energy without further forcing separation. We report each schedule in the main text by its initial value λ_{init}. Table 4 includes the full sweep over $\lambda_{\text{init}} \in \{0, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 2.0, 2.5, 3.0\}$. The schedules for the original settings are

$$\begin{aligned}
\lambda_{\text{init}} = 0 &: [0, 0, 0, 0, 0, 0], \\
\lambda_{\text{init}} = 0.5 &: [0.5, 0.2, 0.06, 0.02, 0.006, 0], \\
\lambda_{\text{init}} = 1.2 &: [1.2, 0.5, 0.16, 0.05, 0.016, 0], \\
\lambda_{\text{init}} = 2.0 &: [2.0, 0.9, 0.28, 0.09, 0.028, 0].
\end{aligned}$$

This curriculum separates the two roles of the objective: early stages encourage multiple basins of attraction, while later stages refine the discovered states toward lower-energy Allen–Cahn solutions.

F.2 Additional Experiments: ML-initialized Solver

We first compare three test-time solvers on the canonical 64×64 problem:

1. **pure ML**: the learned weight-tied dynamic trained with $\lambda = 0$ (energy only),
2. **numerical solver**: the classical periodic IMEX solver,
3. **hybrid solver**: a hybrid scheme that first applies the learned solver and then hands the iterate to the IMEX solver.

The hybrid solver uses the learned model for at most the first 50 steps. Let

$$r_t^{\text{fp}} = \left(\frac{1}{n^2} \sum_{p,q} \big(u_{t+1}(p,q) - u_t(p,q)\big)^2 \right)^{1/2}$$

denote the fixed-point residual of the learned recurrence on an $n \times n$ grid. The handoff criterion is $r_t^{\text{fp}} \le 10^{-2}$. If this never occurs, the switch is forced at step 50. After the switch, the IMEX solver continues from the learned state using the same total budget of 200 recorded iterations.

Result. Figure 6 reports the resulting energy and residual trajectories. The learned model decreases the variational objective rapidly, but its residual MSE plateaus. The numerical solver reduces the residual more steadily, but may remain at a higher energy for a substantial part of the trajectory. The hybrid method combines the two effects: it exploits the learned solver to reach a low-energy basin quickly and then uses the numerical solver to refine the state toward a smaller residual.

F.3 Additional Experiments: Resolution Scaling

We extend the main Allen–Cahn comparison beyond the 64×64 setting to 96×96 and 128×128 grids. The physical problem and other training hyperparameters are kept fixed. To fit memory, we

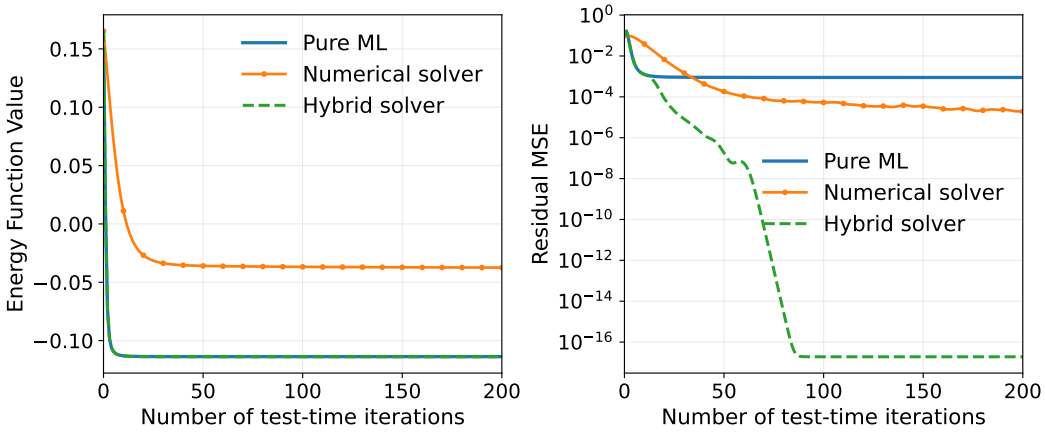

Figure 6: Convergence comparison for *pure ML*, the *numerical solver*, and the *hybrid solver*.

Grid	Training obj.	Train error	Test error	Train energy	Test energy	# Sols
96×96	IMEX label	9.51×10^{-2}	9.53×10^{-2}	1.38×10^{-1}	1.37×10^{-1}	1.00
96×96	Energy only	1.55×10^{-3}	1.54×10^{-3}	-1.15×10^{-1}	-1.16×10^{-1}	1.04
96×96	Div. $\lambda_{\text{init}} = 0.5$	1.47×10^{-3}	1.46×10^{-3}	-1.15×10^{-1}	-1.16×10^{-1}	1.11
96×96	Div. $\lambda_{\text{init}} = 1.2$	1.27×10^{-3}	1.26×10^{-3}	-1.15×10^{-1}	-1.16×10^{-1}	1.14
96×96	Div. $\lambda_{\text{init}} = 2.0$	1.45×10^{-2}	1.45×10^{-2}	-5.89×10^{-2}	-5.96×10^{-2}	19.74
128×128	IMEX label	9.26×10^{-2}	9.34×10^{-2}	1.37×10^{-1}	1.36×10^{-1}	1.00
128×128	Energy only	2.24×10^{-3}	2.22×10^{-3}	-1.13×10^{-1}	-1.14×10^{-1}	1.01
128×128	Div. $\lambda_{\text{init}} = 0.5$	1.88×10^{-3}	1.85×10^{-3}	-1.13×10^{-1}	-1.14×10^{-1}	1.05
128×128	Div. $\lambda_{\text{init}} = 1.2$	1.97×10^{-3}	1.94×10^{-3}	-1.12×10^{-1}	-1.14×10^{-1}	1.13
128×128	Div. $\lambda_{\text{init}} = 2.0$	2.69×10^{-3}	2.70×10^{-3}	-1.11×10^{-2}	-1.12×10^{-2}	19.98

Table 14: Resolution extension of the Allen–Cahn study. We report mean best residual MSE, mean best energy, and the average number of distinct final-state clusters over 20 trajectories per forcing instance. Train/test error denote residual MSE. Lower error/energy is better; larger # Sols indicates more output clusters.

use 32 retained Fourier modes at all resolutions and reduce the batch size from 16 at 64×64, to 8 at 96×96, and 4 at 128×128.

Result. Table 14 shows that the same qualitative pattern persists across resolutions. IMEX-label fitting produces one output cluster and gives the largest residual MSE and energy. Energy-only training also produces essentially one cluster. Increasing the initial diversity weight λ_{init} can recover many distinct final-state clusters, especially at larger weights, but this generally comes with worse residual MSE and energy. Thus the accuracy–diversity tradeoff observed in the main 64×64 experiment is not an artifact of a single grid resolution.

F.4 Additional Experiments: Initialization Sensitivity

We vary the smoothness of the initial field for the model trained with $\lambda_{\text{init}} = 1.2$, keeping its weights fixed. The initial state is $u_0 = c\widetilde{u}$, where $\widetilde{u}$ is a normalized, zero-mean GRF with the spectral filter specified above. We hold $c = 0.5$ and $\ell = 8$ fixed and vary α; larger α gives smoother fields. Table 15 reports means over 1000 test instances, with 20 initializations and 50 learned iterations per instance. Residual MSE and energy are averaged over final outputs in this sensitivity study, rather than minimized along trajectories as in Table 4. Smoother initial fields reduce the cluster count, while residual MSE and energy change more moderately. Thus initialization affects which basins are explored; the observed diversity is not independent of the initialization distribution.

Table 15: Allen–Cahn initialization smoothness at $\lambda_{\text{init}} = 1.2$. All entries are means. #Sols uses the raw-output RMSE threshold 0.8.

α	Residual MSE	Energy	#Sols
1	6.473×10^{-3}	-0.0553	19.36
2	5.998×10^{-3}	-0.0468	17.74
3	5.887×10^{-3}	-0.0425	16.16
4	5.865×10^{-3}	-0.0403	15.05

Table 16: Matched training-rule comparisons. Peak gradient is the largest logged gradient norm in a run. Ising entries are means over three training seeds; Allen–Cahn entries come from one training seed.

Domain	Training rule	Test objective	Peak gradient
Ising	Phantom unrolling	-0.9610	2.166×10^{1}
Ising	Equilibrium-style	0.8216	2.318×10^{12}
Allen–Cahn	BPTT	-0.1135	9.668×10^{-2}
Allen–Cahn	Equilibrium-style	0.2165	1.246×10^{2}

F.5 Additional Experiments: Training Rule and Computational Cost

We compare unrolled training with equilibrium-style implicit differentiation on the same transition architecture and objective within each domain, following the equilibrium-learning viewpoint of Bai et al. [4], Marwah et al. [28]. The reported unrolled method uses phantom gradients for Ising and backpropagation through time for Allen–Cahn. Table 16 summarizes these matched stress tests. The equilibrium-style baseline has larger peak gradient norms and worse final objectives in both domains. These observations concern the tested configurations, not an impossibility result for equilibrium learning. Global contraction would preclude multiple fixed points, but implicit differentiation can also be used at an individual noncontractive equilibrium when its backward linear system is well conditioned.

If one operator evaluation costs C, a T-step rollout costs $O(TC)$, compared with $O(C)$ for a single forward pass. An equilibrium solver using K operator evaluations costs $O(KC)$ plus solver overhead. Neither approach is inherently faster: the comparison depends on stopping tolerances, iteration counts, and convergence. Weight sharing keeps parameter count independent of rollout length, but does not remove the computational cost of repeated evaluations.

F.6 Additional Experiments: Quantitative IMEX Refinement

We extend the hybrid-solver comparison in Appendix F.2 to the diversity-regularized models. Table 17 reports the latest saved iterate within each estimated time budget, with no interpolation. These convergence runs use a separate bank of 100×5 trajectories and 200 recorded iterations. The time estimates use 0.08888 seconds per learned step and 0.41673 seconds per IMEX step; hybrid times are computed trajectory-wise from the saved handoff steps and averaged over the bank. Thus these are estimated convergence times, separate from the 20-initialization evaluation used for the diversity and cluster statistics in Table 4.

For $\lambda_{\text{init}} = 1.0$ and 1.1, the hybrid reaches residual MSE 3.98×10^{-16} at 34.8 estimated seconds and 4.73×10^{-12} at 34.9 estimated seconds, respectively. Pure IMEX is at 5.98×10^{-5} at the latter budget and ends at 1.95×10^{-5} after 83.3 estimated seconds. Together with the separate diversity evaluation, these results show a speed–accuracy–diversity tradeoff. The high-diversity setting $\lambda_{\text{init}} = 2.5$ attains refined diversity 1.358 and raw cluster count 19.51 in the separate evaluation of Table 4. The learned update is useful for basin discovery, while IMEX supplies subsequent high-accuracy refinement.

Relation to multi-solution PDE methods. Methods such as HomPINNs and ensemble PINNs [19, 43] study multiple branches of a fixed PDE configuration. Our evaluation instead trains one operator across forcing instances and returns multiple candidates for unseen instances without retraining. The equilibrium neural operator of Marwah et al. [28] provides a closer cross-instance comparison, but

Table 17: Hybrid and pure-IMEX convergence. Each cell shows residual MSE above energy at the latest iterate within the estimated budget. All values are averages over the convergence evaluation bank.

Method	10 s	20 s	50 s
Hybrid, $\lambda_{\text{init}} = 1.0$	2.67×10^{-4} -0.1115	1.32×10^{-6} -0.1116	1.90×10^{-17} -0.1116
Hybrid, $\lambda_{\text{init}} = 1.1$	4.42×10^{-4} -0.0996	1.17×10^{-6} -0.0998	2.47×10^{-13} -0.0998
Hybrid, $\lambda_{\text{init}} = 1.2$	6.00×10^{-4} -0.0434	9.73×10^{-5} -0.0441	3.65×10^{-5} -0.0449
Pure IMEX	4.08×10^{-3} -0.0301	2.29×10^{-4} -0.0358	3.75×10^{-5} -0.0370

targets a single-valued solution operator; the matched gradient-rule experiment above isolates the training-rule difference.

G Scope, Alternatives, and Limitations

Deterministic solution sets and noisy observations. Our target is the admissible set $\mathcal{F}(\boldsymbol{x})$ specified by the problem, rather than the frequency with which its elements appear as labels. Random initialization induces a distribution over recovered equilibria, but we do not claim that it estimates $p(\boldsymbol{y} \mid \boldsymbol{x})$. Genuine multiplicity is not identifiable from arbitrary noisy labels alone: observations a, b with probabilities $0.8, 0.2$ are compatible both with two valid branches and with one clean target a corrupted to b. These models have the same observed distribution. Distinguishing them requires additional information, such as a known noise model, trusted annotations, or an independent validity criterion. Noise-robust detection and calibrated branch probabilities remain outside our guarantees.

Alternative output representations. Concatenation or a fixed K-head predictor is appropriate when a small, fixed solution count and suitable training targets are available. Variable or large solution sets require selecting, padding, or repeating targets, and ordered outputs introduce a correspondence problem. Permutation-invariant matching removes the arbitrary ordering requirement, but does not remove the fixed output budget or the need for valid targets. Our representation instead explores the equilibria of one shared operator through multiple initializations; finite sampling does not guarantee exhaustive branch coverage.

A non-energy application direction. Image-specific ambiguity in 6D pose estimation can yield multiple valid poses because of symmetry, occlusion, or viewpoint. BOP-Distrib documents this on real images from T-LESS and YCB-V (Meden et al., *BOP-Distrib: Revisiting 6D Pose Estimation Benchmarks for Better Evaluation under Visual Ambiguities*, 2026). For finitely many admissible pose modes, a possible adaptation of Algorithm 1 would refine multiple initial pose hypotheses using one shared operator and match each output to an admissible target. This direction does not require a physical energy objective, but has not been empirically validated here.

Theoretical and empirical scope. The theory is nonparametric and architecture-independent; it is not a guarantee that a chosen neural parameterization or training algorithm learns every branch. Ising and Allen–Cahn offer complementary illustrations: discrete graph states versus continuous fields, GNN versus FNO updates, and spontaneous versus explicitly encouraged discovery. These two applications do not establish universal empirical performance. The convergence and selector results retain their stated regularity, stability, and exceptional-set restrictions.

H LLM Usage Statement

We employed Large Language Models (LLMs) to assist with editing for grammar, spelling, and word choice, as well as to help visualize results and facilitate experimental runs. We have carefully checked and verified all results presented in this paper.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our theories in Section 2 and experiments in Sections 3 and 4 support the claims in the abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: In Section 2, we clearly state: *Importantly, this does not mean multi-branch learning should unconditionally replace conventional supervised learning.*

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes]

Justification: Assumptions in Section 2, complete proof in Appendix.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All details provided in Appendix.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: Our codes are provided here: `https://github.com/liujl11git/bifurcation_models`

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: All details included in Appendix.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: See Table 2, Figure 5, and appendix details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: See Appendix D.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research in this paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This is a fundamental theory work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: This paper poses no such risks.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: Everything used in this paper has been properly credited and used following their licenses.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: This paper does not release new assets.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: We declared in the Appendix.

Guidelines:

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