Journal of Machine Learning **ISSN**: 2790-2048(e), 2790-203X(p)

Cluster-based classification with neural ODEs via control

Antonio Álvarez-López * [1,3], Rafael Orive-Illera † [1,2], and Enrique Zuazua ‡ [1,3,4]

[1]Departamento de Matemáticas, Universidad Autónoma de Madrid, Madrid, Spain.
[2]Instituto de Ciencias Matemáticas, CSIC-UAM-UC3M-UCM, Madrid, Spain.
[3]Chair for Dynamics, Control, Machine Learning, and Numerics, Alexander von Humboldt-Professorship, Department of Mathematics, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany.
[4]Chair of Computational Mathematics, Fundación Deusto, Bilbao, Basque Country, Spain.

Abstract. We address binary classification using neural ordinary differential equations from the perspective of simultaneous control of N data points. We consider a single-neuron architecture with parameters fixed as piecewise constant functions of time. In this setting, the model complexity can be quantified by the number of control switches. Previous work has shown that classification can be achieved using a point-by-point strategy that requires $O(N)$ switches. We propose a new control method that classifies any arbitrary dataset by sequentially steering clusters of d points, thereby reducing the complexity to $O(N/d)$ switches. The optimality of this result, particularly in high dimensions, is supported by some numerical experiments. Our complexity bound is sufficient but often conservative because same-class points tend to appear in larger clusters, simplifying classification. This motivates studying the probability distribution of the number of switches required. We introduce a simple control method that imposes a collinearity constraint on the parameters, and analyze a worst-case scenario where both classes have the same size and all points are i.i.d. Our results highlight the benefits of high-dimensional spaces, showing that classification using constant controls becomes more probable as d increases.

Keywords:
Classification,
complexity,
neural ODEs,
separability,
controllability.

Article Info.:
Volume: 1
Number: 2
Pages: 1- xx
Date: June/2022
doi.org/10.4208/jml.xxx

Article History:
Received: xx/xx/2022
Accepted: xx/xx/2022

Communicated by:
xxx

1 Introduction

At the heart of machine learning lies supervised learning [1], a framework that has been successfully applied in a vast number of domains [2, 3]. The main objective is to learn an unknown mapping $F : \mathscr{X} \to \mathscr{Y}$. To achieve this, a model $\hat{F} : \mathscr{X} \to \mathscr{Y}$ to approximate F is constructed by minimizing a loss function, using only the available—possibly noisy—values of F over a finite dataset $\mathscr{D} \subset \mathscr{X} \times \mathscr{Y}$. Our focus is on evaluating the minimal complexity required for $\hat{F}$ to fit the points in $\mathscr{D}$ without error.

In the context of data classification, the range of F is finite, and its elements are referred to as labels. Over the years, a wide variety of models have been developed, with notable examples including linear discriminants [4], support vector machines [5], random forests [6], and neural networks [7]. In [8], a methodology from a control perspective is proposed,

*Corresponding author. antonio.alvarezl@uam.es.
†rafael.orive@icmat.es.
‡enrique.zuazua@fau.de.

based on modeling deep residual networks (ResNets, [9]) as continuous-time dynamical systems known as neural ordinary differential equations (neural ODEs).

Neural ODEs have seen the development of several variants [10–12], yet the standard form remains as

$$\begin{cases} \dot{\mathbf{x}}(t) = W(t)\,\sigma(A(t)\,\mathbf{x}(t) + \mathbf{b}(t)), & t \in (0,T), \\ \mathbf{x}(0) = \mathbf{x}_0, & \end{cases} \tag{1.1}$$

where:

- $\mathbf{x}_0 \in \mathbb{R}^d$ is an input point;
- $(W, A, \mathbf{b}) \in L^\infty((0,T), \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times d} \times \mathbb{R}^p)$ are parameters to be trained;
- $d, p \geq 1$ are the state dimension and the width of the model, respectively;
- $\sigma : \mathbb{R}^p \to \mathbb{R}^p$ is a prefixed nonlinear Lipschitz function applied component-wise.

Existence and uniqueness of the solution to (1.1) is guaranteed by the Cauchy–Lipschitz theorem, ensuring the well-definedness of the flow map

$$\Phi_t(\cdot; W, A, \mathbf{b}) : \mathbf{x}_0 \in \mathbb{R}^d \longmapsto \mathbf{x}(t) \in \mathbb{R}^d, \qquad \text{for } t \in [0,T]. \tag{1.2}$$

The formulation of (1.1) naturally frames supervised learning as a control problem. Here, the input space is $\mathscr{X} = \mathbb{R}^d$, and the parameters $(W, A, \mathbf{b})$ serve as controls that simultaneously guide all input points toward their respective target positions in $\mathbb{R}^d$. To match the output space, a mapping $g : \mathbb{R}^d \to \mathscr{Y}$ is introduced as a final layer. The complete model is thus defined by the composition $\hat{F} = g \circ \Phi_T$.

We focus on binary classification with a hard classifier, whereby $\mathscr{Y} = \{1, 0\}$ and g is the characteristic function of a fixed set. Nonetheless, our results can be extended to any multiclass setting by fixing g as a weighted sum of predefined characteristic functions, each corresponding to a distinct label.

Neural ODEs were originally conceived as a tool for understanding deep ResNets, but they have since made a significant impact on machine learning. Their continuous-time framework facilitates mathematical analysis and provides practical benefits like incorporating structure or the design of new discrete schemes. For more details, we refer to [8].

Notation

- Scalars are denoted by plain letters, vectors by boldface letters, and matrices by uppercase letters. The scalar product of two vectors $\mathbf{u}$, $\mathbf{v}$ is written as $\mathbf{u} \cdot \mathbf{v}$.
- Subscripts identify elements of a set. Superscripts identify components of a vector.
- $\{\mathbf{e}_1, \ldots, \mathbf{e}_d\}$ denotes the canonical basis in $\mathbb{R}^d$.
- $\mathbb{S}^{d-1}$ denotes the $(d-1)-$dimensional sphere in $\mathbb{R}^d$.
- The cardinality of a set $\mathcal{X}$ is denoted by $|\mathcal{X}|$.
- For $x \in \mathbb{R}$, we write $\lceil x \rceil := \min\{n \in \mathbb{Z} : n \geq x\}$, $\lfloor x \rfloor := \max\{n \in \mathbb{Z} : n \leq x\}$.

2 Problem formulation and main results

Let $\mathscr{D} := \{(\mathbf{x}_n, y_n)\} \subset \mathbb{R}^d \times \{1, 0\}$ be a finite dataset such that $\mathbf{x}_i \neq \mathbf{x}_j$ for all $i \neq j$. We define the classes $\mathcal{R}$ (red circles) and $\mathcal{B}$ (blue crosses) by

$$\mathcal{R} = \{\mathbf{x}_n \in \mathbb{R}^d : \ (\mathbf{x}_n, 1) \in \mathscr{D}\}, \qquad \mathcal{B} = \{\mathbf{x}_n \in \mathbb{R}^d : \ (\mathbf{x}_n, 0) \in \mathscr{D}\}. \tag{2.1}$$

We adopt the simplified version of neural ODEs with one-neuron width. Namely, we set $p = 1$ in (1.1), which yields

$$\dot{\mathbf{x}}(t) = \mathbf{w}(t)\sigma(\mathbf{a}(t) \cdot \mathbf{x}(t) + b(t)), \qquad \text{for } t \in (0, T). \tag{2.2}$$

Here, $\sigma(\cdot) = (\cdot)_+$ is the rectified linear unit (ReLU), while $\theta = (\mathbf{w}, \mathbf{a}, b)$ belongs to

$$\Theta_T := L^\infty \left((0, T) \ ; \ \mathbb{S}^{d-1} \times \mathbb{R}^d \times \mathbb{R} \right).$$

and is assumed to be piecewise constant. Imposing this constraint reduces optimization to a finite-dimensional space [12], while also inducing a layered structure akin to that of a discrete ResNet [13, 14]. For any control $\theta \in \Theta_T$, we define the complexity of (2.2) as the number of finite-jump discontinuities (or switches) of θ over $(0, T)$, denoted by L.

Classification can essentially be interpreted as transforming data into representations in which different classes are separable. In the neural ODE framework, given a dataset $(\mathcal{R}, \mathcal{B})$, the goal is to find a control $\theta \in \Theta_T$ for (2.2) that induces a finite-time flow map Φ_T satisfying

$$\Phi_T(\mathcal{R}; \theta) \subset \tau_{\mathcal{R}} \qquad \text{and} \qquad \Phi_T(\mathcal{B}; \theta) \subset \tau_{\mathcal{B}},$$

where $(\tau_{\mathcal{R}}, \tau_{\mathcal{B}})$ is a pair of linearly separable regions of $\mathbb{R}^d$. For simplicity, we fix $(\tau_{\mathcal{R}}, \tau_{\mathcal{B}})$ to be half-spaces of the form

$$\left(\{x^{(i)} > 1\}, \{x^{(i)} \leq 1\} \right) \quad \text{or} \quad \left(\{x^{(i)} \leq 1\}, \{x^{(i)} > 1\} \right) \qquad \text{for } i \in \{1, \ldots, d\}, \tag{2.3}$$

and the hyperplane $\{x^{(i)} = 1\}$ as decision boundary. We make use of the dynamics of (2.2) in a constructive manner, by carefully defining each value of the piecewise constant control $\theta \in \Theta_T$. On the layer $t \in (t_{k-1}, t_k) \subset (0, T)$, the parameters $\mathbf{a}(t) \equiv \mathbf{a} \in \mathbb{R}^d$ and $b(t) \equiv b \in \mathbb{R}$ determine the hyperplane $H : \mathbf{a} \cdot \mathbf{x} + b = 0$ and the two half-spaces

$$H^+ := \{\mathbf{x} \in \mathbb{R}^d : \mathbf{a} \cdot \mathbf{x} + b > 0\}, \qquad H^- := \mathbb{R}^d \setminus H^+.$$

The half-space H^- remains fixed because $\sigma(\mathbf{a} \cdot \mathbf{x} + b) = 0$ for all $\mathbf{x} \in H^-$. in contrast, each point $\mathbf{x} \in H^+$ evolves according to the vector field $\mathbf{w}(\mathbf{a} \cdot \mathbf{x} + b)$. The direction of this field is constant and determined by $\mathbf{w} \in \mathbb{S}^{d-1}$, while its magnitude at each $\mathbf{x} \in H^+$ is equal to the distance from $\mathbf{x}$ to the hyperplane H.

This overview of the dynamics suggests that classifying data points using (2.2) is intrinsically related to identifying a set of hyperplanes that separate them by labels. All in all, there is a clear relationship between the complexity of the controls in (2.2) and the geometric structure of the data distribution. The following questions naturally arise:

Question 1. What is the minimum number of discontinuities required to ensure that any dataset with a fixed number of points can be classified?

Question 2. What is the probability that a given dataset can be classified using exactly $L = k$ discontinuities, for any $k \geq 0$?

Regarding Question 1, it was established in [13] that $L = 3\min\{|\mathcal{R}|, |\mathcal{B}|\}$ switches are sufficient to classify $(\mathcal{R}, \mathcal{B})$ with (2.2). The proof involves sequentially steering each point in $\mathcal{R} \cup \mathcal{B}$ individually to its corresponding target $\tau_{\mathcal{R}}$ or $\tau_{\mathcal{B}}$. Our first main result refines this bound by relying on the concept of general position, a classical notion in combinatorial geometry [15–18]. We consider the following definition, illustrated in Figure 2.1:

Definition 2.1. *A set $\mathcal{X} \subset \mathbb{R}^d$ is in* general position *if, for every $0 \leq k \leq d-1$, no affine subspace in $\mathbb{R}^d$ of dimension k contains more than $k+1$ points of $\mathcal{X}$.*

Assuming this mild condition—easily achieved by slight perturbations of the dataset—we construct a family of pairwise parallel hyperplanes that enclose all points of one class by subsets of size d. We can then define a simultaneous control method that classifies the points within these subsets rather than individually, thereby reducing the bound for L:

Theorem 2.1. *Let $d \geq 2$. For any dataset $(\mathcal{R}, \mathcal{B})$ defined as in* (2.1) *in general position and any pair of target sets $(\tau_{\mathcal{R}}, \tau_{\mathcal{B}})$ defined as in* (2.3)*, there exist $T > 0$ and a piecewise constant control $\theta \in \Theta_T$ whose number of discontinuities is*

$$L = 4\left\lceil \frac{\min\{|\mathcal{R}|, |\mathcal{B}|\}}{d} \right\rceil - 1,$$

such that the flow map of the neural ODE (2.2) *satisfies $\Phi_T(\mathcal{R}; \theta) \subset \tau_{\mathcal{R}}$ and $\Phi_T(\mathcal{B}; \theta) \subset \tau_{\mathcal{B}}$.*

Theorem 2.1 holds almost surely as long as the points are sampled from a non-singular measure. However, numerous data points sharing the same label may initially be clustered together, enabling their control with fewer parameters. This observation motivates the introduction of a probabilistic framework to address Question 2.

We consider the worst-case scenario where all points are i.i.d., and the sizes are fixed at $|\mathcal{R}| = |\mathcal{B}| = N$. To simplify the analysis, we introduce a new control method that restricts $\mathbf{a}(t)$ to be a constant, although optimally chosen. This constraint will allow us to determine the exact probability distribution of L for this method.

Theorem 2.2. *Let $d \geq 2$ and $N \geq 1$. Consider any dataset $(\mathcal{R}, \mathcal{B})$ defined as in* (2.1)*, with $|\mathcal{R}| = |\mathcal{B}| = N$. Assume every point $\mathbf{x} \in \mathcal{R} \cup \mathcal{B}$ is independently sampled from an absolutely continuous probability measure on $[0,1]^d$. Then, with probability 1, there exist $T > 0$, a pair of target sets $(\tau_{\mathcal{R}}, \tau_{\mathcal{B}})$ defined as in* (2.3)*, and $\theta = (\mathbf{w}, \mathbf{a}, b) \in \Theta_T$ with*

$$\mathbf{a}(t) \in \{\mathbf{e}_1, \dots, \mathbf{e}_d\} \qquad \textit{constant for all } t \in (0, T),$$

such that the flow map of the neural ODE (2.2) *satisfies $\Phi_T(\mathcal{R}; \theta) \subset \tau_{\mathcal{R}}$ and $\Phi_T(\mathcal{B}; \theta) \subset \tau_{\mathcal{B}}$.*

Furthermore, θ is piecewise constant with L discontinuities following, for $0 \leq k \leq 2N-2$,

$$\mathbb{P}(L \geq k) = \left(\sum_{p=\lceil \frac{k}{2}+1 \rceil}^{N} \binom{N-1}{p-1}^2 + \sum_{p=\lceil \frac{k+1}{2} \rceil}^{N-1} \binom{N-1}{p}\binom{N-1}{p-1} \right)^d 2^d \binom{2N}{N}^{-d}. \tag{2.4}$$

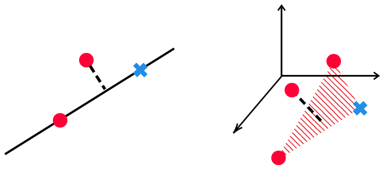

Figure 2.1: $\mathcal{X} \subset \mathbb{R}^2$ is in general position if no three points of $\mathcal{X}$ lie on the same line. $\mathcal{X} \subset \mathbb{R}^3$ is in general position if, additionally, no four points lie on the same plane.

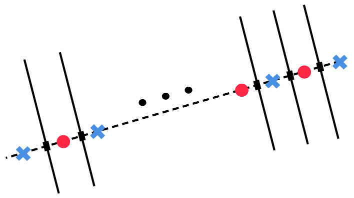

Figure 2.2: Configuration where the maximum number of $2N - 1$ hyperplanes is required to separate the points.

Remark 2.1. Stirling's approximation simplifies (2.4) for large values of d and N. If we set $k = 1$, the formula reduces to

$$\mathbb{P}(L \geq 1) = \left(1 - 2\binom{2N}{N}^{-1}\right)^d \sim \exp\left\{\frac{-\sqrt{\pi N}}{2^{2N-1}}\, d\right\}.$$

Assuming both d and N grow according to some relation $d = d(N)$, we can deduce that

$$\text{if} \qquad \lim_{N\to\infty} \frac{2^{2N}}{d(N)\sqrt{N}} = 0 \qquad \text{then} \qquad \lim_{N\to\infty} \mathbb{P}(L = 0) = 1.$$

This observation reveals an explicit trade-off between d and N that allows classifying all points employing a constant control, or equivalently, with an autonomous neural ODE.

The control method of Theorem 2.2 is designed so that distribution (2.4) is derived from the number of hyperplanes required to separate the points by labels. The maximum value $L = 2N - 2$, corresponding to $2N - 1$ hyperplanes, also improves upon the bound of [13].

In the following theorem, we take a geometric perspective on the problem to show that the maximum number of hyperplanes required to separate the two classes is indeed $2N - 1$. Moreover, we characterize the pathological point configurations that attain this maximum, as illustrated in Figure 2.2.

Theorem 2.3. *Let $d, N \geq 1$. For any dataset $(\mathcal{R}, \mathcal{B})$ defined as in* (2.1) *with $|\mathcal{R}| = |\mathcal{B}| = N$, the maximum number of hyperplanes in $\mathbb{R}^d$ required to separate the points by labels is $2N - 1$. Furthermore, this maximum is attained if and only if the points of $\mathcal{R}$ and $\mathcal{B}$ are collinear and alternating, i.e., there exist $\mathbf{u}, \mathbf{u}_0 \in \mathbb{R}^d$ and $-\infty < \lambda_1 < \cdots < \lambda_{2N} < +\infty$ such that*

$$\mathbf{u}_0 + \lambda_{2k-1}\mathbf{u} \in \mathcal{R} \qquad \textit{and} \qquad \mathbf{u}_0 + \lambda_{2k}\mathbf{u} \in \mathcal{B} \qquad \textit{for all } k \in \{1, \ldots, N\}.$$

We note that deriving the maximum is relatively straightforward; the main challenge lies in showing that it is attained only when all points are collinear and alternating.

Roadmap

In Section 3, we develop the mathematical framework, which mainly relies on hyperplane separability. We begin by proving Theorem 2.3 and then present combinatorial results—first for $d = 1$ and later in higher dimensions—that culminate in Theorem 2.2. In Section 4, assuming all points are in general position (see Definition 2.1), we construct a family of pairwise parallel hyperplanes that separate the dataset, with each pair enclosing exactly d points of the same class. Next, we prove Theorem 2.1 by defining a classification method that sequentially controls these subsets of size d, and analyze alternative activation functions that further reduce the value of L. Although most results are static and not specific to neural ODEs, Theorems 2.1 and 2.2 incorporate dynamic algorithms, distinguishing them from standard linear classifiers. In Section 5, we perform a computational test using gradient-based training to estimate the minimal complexity required for neural ODEs to classify datasets of fixed size. We then compare these results to the complexity obtained in Theorem 2.1. We conclude with a summary of our main contributions, a discussion on connections and extensions, and open questions for future work.

Related work

The theoretical study of neural ODEs from a control theory perspective has gained significant attention in recent years, opening several and promising research directions.

One line of research explores the approximation power of neural ODE flows, using either geometric techniques based on Lie brackets [19–21] or constructive methods for simultaneous control [13, 22]. In [23], the authors conducted a detailed study of controllability and its connection with density in L^p spaces for $p < +\infty$. Universal approximation in L^∞ was established for a specific class of diffeomorphisms in [24]. We also highlight the work of [25], which demonstrates density in $L^1(\mathbb{R}^d)$ for the family of ResNets with one neuron per hidden layer, corresponding to the forward Euler discretization of (2.2).

Another approach studies the training process as an optimal control problem. In [26], population risk minimization in deep learning is formulated as a mean-field optimal control problem, and Hamilton–Jacobi–Bellman and Pontryagin optimality conditions are derived. [27] proposes a modification of the successive approximations method by augmenting the Hamiltonian to solve Pontryagin's maximum principle, resulting in an alternative training algorithm with rigorous error estimates. [28] employs a continuity equation to study the mean-field dynamics, examining the existence and uniqueness of minimizers in the optimal control problem with L^2-regularization and establishing a mean-field maximum principle. In [29], the result is improved by introducing a kinetic regularization term in the loss function, and proving the existence of minimizers. [30] considers the classical empirical risk minimization, deriving a manifestation of the turnpike property through specific regularization terms. Additionally, [31] reduces the complexity for data classification by introducing an L^1-norm penalty, which promotes temporal sparsity in the control.

We follow a third direction that consists of estimating the complexity required to control N points. To the best of our knowledge, the only study addressing this problem in the one-neuron width model (2.2) is [13], where the bound of $O(N)$ switches is established. We build upon their results by reducing to $O(N/d)$ switches, and deriving a probabilistic result. In [14], the result of [13] is generalized to any finite width p, resulting in a complexity of $O(N/p)$. The problem is also explored in [32], which examines a neural ODE with a second-order time derivative to further enrich the dynamics, maintaining a complexity of $O(N)$. Furthermore, [33] investigates the controllability of probability measures for the continuity equation that extends (2.2) and considering errors in total variation.

3 Classification via canonical separability

Suppose that all points of the dataset (2.1) are independently sampled from a particular absolutely continuous probability measure on $[0,1]^d$, with $|\mathcal{R}| = |\mathcal{B}| = N$ fixed, and that they satisfy the following condition:

$$x_n^{(j)} \neq x_m^{(j)}, \quad \text{for all } j \in \{1,\dots,d\} \quad \text{and} \quad n \neq m. \tag{3.1}$$

Note that (3.1) is fulfilled with probability 1. We now introduce our concept of separability:

Definition 3.1. *A finite family $\mathcal{H}$ of affine $(d-1)$-dimensional hyperplanes in $\mathbb{R}^d$* separates *$(\mathcal{R},\mathcal{B})$ if they break the cube $[0,1]^d$ into polyhedra, each of them containing points of at most one of the two sets. We say that such $\mathcal{H}$ is a collection of* separating hyperplanes *for $(\mathcal{R},\mathcal{B})$.*

Note that some polyhedra might not contain any point from the two sets. Now, for any pair $(\mathcal{R},\mathcal{B})$ as in (2.1), and any collection of hyperplanes $\mathcal{H}$ in $\mathbb{R}^d$, let us consider

$$Z_{d,N}(\mathcal{R},\mathcal{B}) := \min\left\{|\mathcal{H}| \,:\, \mathcal{H} \text{ separates } (\mathcal{R},\mathcal{B})\right\}. \tag{3.2}$$

As defined, $Z_{d,N}$ is a random variable that maps each pair $(\mathcal{R},\mathcal{B})$ to the minimum cardinality of any collection of separating hyperplanes for $(\mathcal{R},\mathcal{B})$.

Theorem 2.2 is based on separating $(\mathcal{R},\mathcal{B})$ using hyperplanes that are orthogonal to an optimally chosen canonical vector. We refer to this approach as *canonical k-separability*, drawing on the concept of *k-separability* introduced in [34]. A finite set $\{\mathbf{x}_n\} \subset \mathbb{R}^d$ with binary labels is k-separable if there exists $\mathbf{w} \in \mathbb{S}^{d-1}$ such that the projections $\mathbf{w} \cdot \mathbf{x}_n$ can be divided into k disjoint intervals, each containing only projections with the same label.

Canonical k-separability is a similar concept, but it constrains $\mathbf{w}$ to the canonical basis. We will determine the exact probability that $(\mathcal{R},\mathcal{B})$ is k-separable under this constraint, which requires adapting the definition of $Z_{d,N}$ in (3.2).

For each $i = 1,\dots,d$, the usual projection $\pi^i : \mathbb{R}^d \to \mathbb{R}$ is defined by $\pi^i(\mathbf{x}) = x^{(i)}$ for all $\mathbf{x} \in \mathbb{R}^d$. Let Π^i be the pointwise extension of π^i, given by

$$\Pi^i(\mathcal{X}) = \left\{\pi^i(\mathbf{x}_1),\dots,\pi^i(\mathbf{x}_N)\right\} = \left\{x_1^{(i)},\dots,x_N^{(i)}\right\}$$

for all $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N \subset \mathbb{R}^d$ satisfying condition (3.1), which ensures $|\Pi^i(\mathcal{X})| = N$.

We introduce the random variable $Z_{d,N}^i$ such that

$$Z_{d,N}^i(\mathcal{R},\mathcal{B}) = Z_{1,N}\left(\Pi^i(\mathcal{R}),\Pi^i(\mathcal{B})\right) \tag{3.3}$$

for any pair $(\mathcal{R},\mathcal{B})$ defined as in (2.1). The value $Z_{d,N}^i(\mathcal{R},\mathcal{B})$ represents the minimum number of points required to separate the projections of both sets on the i-th Cartesian

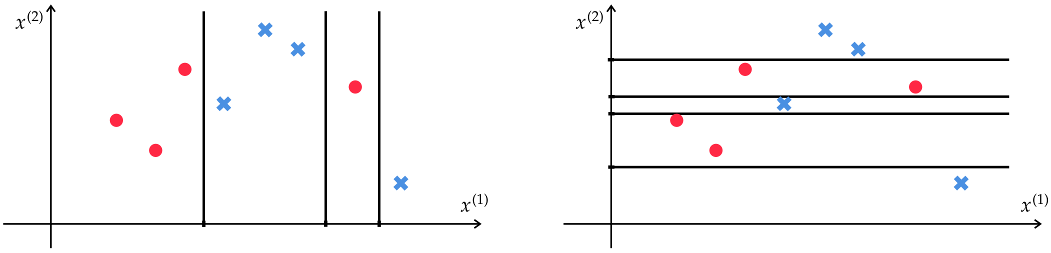

Figure 3.1: $Z^1_{d,N} = 3$ and $Z^2_{d,N} = 4$ computed by projecting the data on the respective axes $x^{(1)}$ and $x^{(2)}$.

axis. Those points determine a family of parallel hyperplanes in $\mathbb{R}^d$ given by the equations $x^{(i)} = H_j$ with $j = 1, \ldots, Z^i_{d,N}(\mathcal{R}, \mathcal{B})$, that separate $(\mathcal{R}, \mathcal{B})$ in $\mathbb{R}^d$, see Figure 3.1.

Given $d, N \geq 1$, we define

$$Z^{\perp}_{d,N} := \min \left\{ Z^1_{d,N}, \ldots, Z^d_{d,N} \right\}. \tag{3.4}$$

The variable $Z^{\perp}_{d,N}$ represents the minimum cardinality of any family of separating hyperplanes, all of which are perpendicular to some Cartesian axis. For the example shown in Figure 3.1, we would compute $Z^{\perp}_{d,N}(\mathcal{R}, \mathcal{B}) = 3$.

Now, we prove Theorem 2.3 to determine and characterize the maximum value of $Z_{d,N}$:

Proof of Theorem 2.3. By relabeling the points, with no loss of generality we can assume

$$\mathcal{R} = \{\mathbf{x}_n\}_{1 \leq n \leq N} \qquad \text{and} \qquad \mathcal{B} = \{\mathbf{x}_{N+n}\}_{1 \leq n \leq N}.$$

First, we prove

$$\max_{(\mathcal{R},\mathcal{B})} Z_{d,N}(\mathcal{R}, \mathcal{B}) = 2N - 1. \tag{3.5}$$

Case $d = 1$. When the points of the pair $(\mathcal{R}, \mathcal{B})$ are alternating, i.e.,

$$x_1 < x_{N+1} < x_2 < \cdots < x_N < x_{2N} \quad \text{or} \quad x_{N+1} < x_1 < x_{N+2} < \cdots < x_{2N} < x_N, \tag{3.6}$$

we have $Z_{1,N}(\mathcal{R}, \mathcal{B}) = 2N - 1$. Conversely, if the points of $\mathcal{R} \cup \mathcal{B}$ are not alternating, then the number of line segments connecting consecutive points with different labels is less than $2N - 1$ and prove (3.5) with $d = 1$.

Case $d > 1$. Since $Z_{d,N} \leq Z^{\perp}_{d,N}$ as defined in (3.4), we deduce that $\max Z_{d,N} \leq 2N - 1$ using (3.5) for $d = 1$. To show that $\max Z_{d,N} \geq 2N - 1$, assume the $2N$ points of $\mathcal{R} \cup \mathcal{B}$ are collinear and alternating. To separate $(\mathcal{R}, \mathcal{B})$, each of the $2N - 1$ line segments connecting two consecutive points of different labels must then be intersected by a hyperplane. Since any hyperplane that does not contain the entire line can intersect it in at most one point, we need at least $2N - 1$ hyperplanes to separate $(\mathcal{R}, \mathcal{B})$, as illustrated in Figure 2.2.

Necessity of being collinear and alternating (only if). If the points of $\mathcal{R} \cup \mathcal{B}$ are collinear but not alternating, then the number of line segments connecting consecutive points with

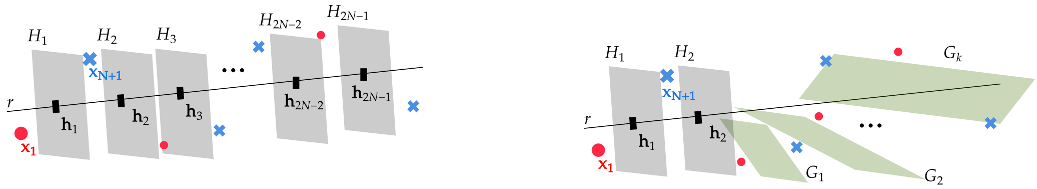

Figure 3.2: Figures supporting the argument presented in the proof of Theorem 2.3.

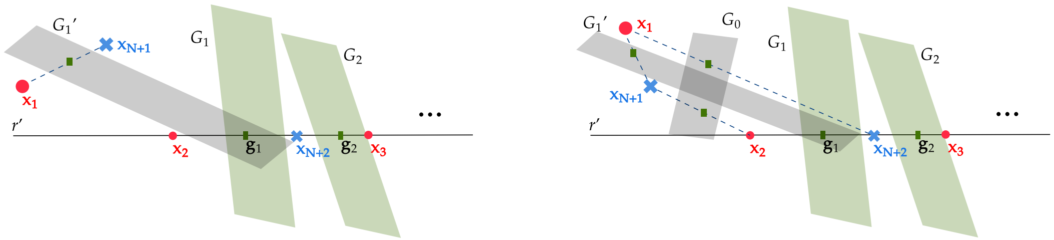

Figure 3.3: Figures supporting the argument presented in the proof of Theorem 2.3.

different labels is less than $2N-1$. Consequently, fewer than $2N-1$ hyperplanes suffice to separate them, that is, $Z_{d,N} < 2N-1$.

To show that collinearity is also a necessary condition to attain the maximum, we proceed by induction on N. For $N=1$ the claim is trivial. For $N=2$, if the four points are not collinear, their convex hull is at most a triangle, quadrilateral, or tetrahedron, which implies $Z_{d,N} \leq 2$.

Let $N > 2$ and suppose that $\mathcal{R} \cup \mathcal{B}$ is not contained in any line, yet $Z_{d,N}(\mathcal{R},\mathcal{B}) = 2N-1$. Since the dataset is finite, we can choose a line $r \subset \mathbb{R}^d$ with direction $\mathbf{v} \in \mathbb{S}^{d-1}$ such that the orthogonal projection of $\mathcal{R} \cup \mathcal{B}$ onto r is injective. The number of points of r required to separate the projected sets is then $2N-1$. Indeed, it is at most $2N-1$ (as per (3.5) for $d=1$), and on the other hand, these points correspond to a family of hyperplanes orthogonal to r that separate $(\mathcal{R},\mathcal{B})$, so it must be at least $Z_{d,N}(\mathcal{R},\mathcal{B}) = 2N-1$. All in all, we can choose hyperplanes $H_1,\dots,H_{2N-1}$ orthogonal to r that separate $(\mathcal{R},\mathcal{B})$, and moreover, the projected points $\{\mathbf{v}\cdot\mathbf{x}_i\}_{i=1}^{2N} \subset \mathbb{R}$ must be alternating as in (3.6). Without loss of generality, we may assume

$$\mathbf{v}\cdot\mathbf{x}_1 < h_1 < \mathbf{v}\cdot\mathbf{x}_{N+1} < \cdots < \mathbf{v}\cdot\mathbf{x}_N < h_{2N-1} < \mathbf{v}\cdot\mathbf{x}_{2N},$$

where $h_i = \mathbf{v}\cdot\mathbf{h}_i$ and $\mathbf{h}_i = H_i \cap r \subset \mathbb{R}^d$, as illustrated in Figure 3.2 (left). We then define $\mathcal{R}' = \mathcal{R} \setminus \{\mathbf{x}_1\}$ and $\mathcal{B}' = \mathcal{B} \setminus \{\mathbf{x}_{N+1}\}$, each containing $N-1$ points.

Now, we show that $Z_{d,N-1}(\mathcal{R}',\mathcal{B}') = 2N-3$. Suppose that k hyperplanes $G_1,\dots,G_k$ in $\mathbb{R}^d$ suffice to separate $(\mathcal{R}',\mathcal{B}')$ for some $1 \leq k \leq 2N-4$. Then the pair $(\mathcal{R},\mathcal{B})$ can be separated using $k+2$ hyperplanes $H_1, H_2, G_1, \dots, G_k$, as illustrated in Figure 3.2 (right). Since $k+2 \leq 2N-2 < 2N-1$, this contradicts the fact that $Z_{d,N}(\mathcal{R},\mathcal{B}) = 2N-1$.

The remainder of the proof consists of verifying that $(\mathcal{R}, \mathcal{B})$ can be separated by at most $2N-2$ hyperplanes, which would lead to a contradiction. Since $Z_{d,N-1}(\mathcal{R}', \mathcal{B}') = 2N-3$, the inductive hypothesis on $N-1$ ensures that $\mathcal{R}' \cup \mathcal{B}'$ must be collinear along some line $r' \subset \mathbb{R}^d$. By assumption, the points in $\mathcal{R} \cup \mathcal{B}$ are not collinear, which implies that $\mathbf{x}_1 \notin r'$ or $\mathbf{x}_{N+1} \notin r'$. Let $G_1, \dots, G_{2N-3}$ be hyperplanes that separate $(\mathcal{R}', \mathcal{B}')$ and each intersecting r' transversely, that is, we can set $\mathbf{g}_j = G_j \cap r' \in \mathbb{R}^d$. We distinguish two cases:

1. If $\mathbf{x}_1$, $\mathbf{x}_{N+1}$ and $\mathbf{g}_1$ are not collinear, we can consider any hyperplane G_1' that intersects the open segment $[\mathbf{x}_1, \mathbf{x}_{N+1}] = \{t\mathbf{x}_1 + (1-t)\mathbf{x}_{N+1} \,|\, 0 < t < 1\}$ transversely, and which also intersects the line r' transversely at $\mathbf{g}_1$. If $\mathbf{x}_1$ is on the same side of G_1' as $\mathbf{x}_2$, the family $\{G_1', G_2, \dots, G_{2N-3}\}$ separates $(\mathcal{R}, \mathcal{B})$—as illustrated in Figure 3.3 (left)—and $Z_{d,N}(\mathcal{R}, \mathcal{B}) = 2N-3$ (contradiction). Otherwise, we also consider any hyperplane G_0 that intersects transversely with the open segment $[\mathbf{x}_2, \mathbf{x}_{N+1}]$ and with the open segment $[\mathbf{x}_1, \mathbf{x}_{N+2}]$. Then, $\{G_0, G_1', G_2, \dots, G_{2N-3}\}$ separates $(\mathcal{R}, \mathcal{B})$, as represented in Figure 3.3 (right), and $Z_{d,N}(\mathcal{R}, \mathcal{B}) = 2N-2$ (contradiction).

2. If $\mathbf{x}_1$, $\mathbf{x}_{N+1}$ and $\mathbf{g}_1$ are collinear, we can slightly translate G_1 and perturb its intersection point with r' from $\mathbf{g}_1$ to another point $\mathbf{g}_1'$ of the open segment $[\mathbf{x}_2, \mathbf{x}_{N+2}]$. Thus, we can ensure that $\mathbf{x}_1$ and $\mathbf{x}_{N+1}$ and $\mathbf{g}_1'$ are not collinear and apply case 1.

□

To obtain the probability distribution of $Z_{d,N}^{\perp}$, we first solve the case $d=1$ employing combinatorial techniques:

Lemma 3.1. *For $N \geq 1$, let $Z_{1,N}$ be as defined in* (3.2). *For any $1 \leq k \leq 2N-1$, we have*

$$\mathbb{P}(Z_{1,N} = k) = \begin{cases} 2\binom{N-1}{p-1}^2 \binom{2N}{N}^{-1} & \text{if} \quad k = 2p-1, \\ 2\binom{N-1}{p}\binom{N-1}{p-1}\binom{2N}{N}^{-1} & \text{if} \quad k = 2p. \end{cases} \tag{3.7}$$

Proof. Since all points in $\mathcal{R} \cup \mathcal{B}$ are i.i.d., each possible ordering is equally likely. Therefore, to determine $\mathbb{P}(Z_{1,N} = k)$, we count the fraction of orderings that have exactly k gaps between consecutive subsets of points with different labels. We compute the number of such favorable configurations by dividing them into two cases based on the parity of k:

Case 1: $k = 2p-1$, for $1 \leq p \leq N$. Any ordering of the points in $\mathcal{R} \cup \mathcal{B}$ that leads to $Z_{1,N}(\mathcal{R}, \mathcal{B}) = 2p-1$ can be represented as the union of two partitions of $\mathcal{R}$ and $\mathcal{B}$, respectively constituted of subsets $R_i \subset \mathcal{R}$ and $B_i \subset \mathcal{B}$, for $i \in 1, \dots, p$, with lengths $|R_i| = r_i$ and $|B_i| = b_i$, that satisfy

$$\sum_{i=1}^{p} r_i = \sum_{i=1}^{p} b_i = N. \tag{3.8}$$

There are $\binom{N-1}{p-1}$ possible partitions $\{R_1, \dots, R_p\}$ of $\mathcal{R}$, determined by the choice of $p-1$ gaps between consecutive elements of $\mathcal{R}$, among the total $N-1$ possibilities. Analogously,

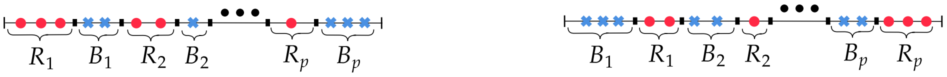

Figure 3.4: Examples for case 1 in the proof of Lemma 3.1.

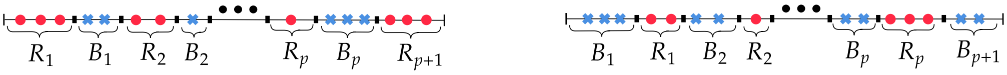

Figure 3.5: Examples for case 2 in the proof of Lemma 3.1.

there are $\binom{N-1}{p-1}$ possible partitions $\{B_1, \dots, B_p\}$ of $\mathcal{B}$, to insert in the p gaps and therefore obtain $Z_{1,N}(\mathcal{R}, \mathcal{B}) = k$. So there are $\binom{N-1}{p-1}^2$ possible configurations. On the other hand, we can repeat the argument but now considering the $\binom{N-1}{p-1}$ possible partitions of $\mathcal{B}$ and then inserting R_i into the p gaps. Both situations are symmetric; the only difference between them lies in whether the smallest point of $\mathcal{R} \cup \mathcal{B}$ (and thus the first subset of the two partitions) belongs to $\mathcal{R}$ or $\mathcal{B}$, as shown in Figure 3.4. Consequently, the total number of configurations for $(\mathcal{R}, \mathcal{B})$ that yield $Z_{1,N}(\mathcal{R}, \mathcal{B}) = 2p - 1$ is $2\binom{N-1}{p-1}^2$.

Case 2: $k = 2p$, for $1 \leq p \leq N - 1$. To obtain $Z_{1,N}(\mathcal{R}, \mathcal{B}) = 2p$, we need an even number of gaps between subsets of the same color, so now we must consider the partitions of $\mathcal{R}$ and $\mathcal{B}$ that are respectively constituted of subsets $R_i, R_{p+1} \subset \mathcal{R}$ and $B_i \subset \mathcal{B}$ for $i \in 1, \dots, p$ (or the reverse situation) with lengths $|R_i| = r_i$ and $|B_i| = b_i$, that satisfy

$$\sum_{i=1}^{p+1} r_i = \sum_{i=1}^{p} b_i = N. \tag{3.9}$$

Like in case 1, there exist $\binom{N-1}{p}$ possible partitions $\{R_1, \dots, R_{p+1}\}$ of $\mathcal{R}$ and $\binom{N-1}{p-1}$ possible partitions $\{B_1, \dots, B_p\}$ of $\mathcal{B}$. So there are $\binom{N-1}{p}\binom{N-1}{p-1}$ possibilities, each one depending on a choice of $(r_1, \dots, r_{p+1}) \in \mathbb{N}^{p+1}$ and $(b_1, \dots, b_p) \in \mathbb{N}^p$ satisfying (3.9). We take also into account the symmetric situation, when the partitions of $\mathcal{R}$ and $\mathcal{B}$ are respectively of the form $\{R_1, \dots, R_p\}$ and $\{B_1, \dots, B_{p+1}\}$, shown in Figure 3.5. Therefore, it follows that the number of configurations for $(\mathcal{R}, \mathcal{B})$ that yield $Z_{1,N}(\mathcal{R}, \mathcal{B}) = 2p$ is $2\binom{N-1}{p}\binom{N-1}{p-1}$.

Finally, the total number of possibilities is exactly the number of ways to choose N points out of $2N$, which is given by the central binomial coefficient $\binom{2N}{N}$. Indeed,

$$2\sum_{p=1}^{N} \binom{N-1}{p-1}^2 + 2\sum_{p=1}^{N-1} \binom{N-1}{p}\binom{N-1}{p-1} = \binom{2N}{N}.$$

□

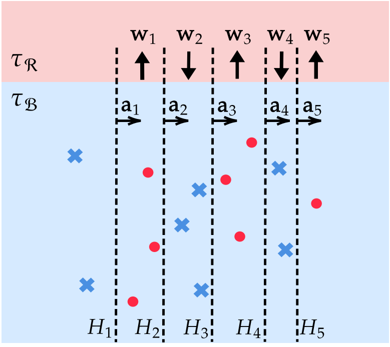

Figure 3.6: Schematic overview of the full control method used to prove Theorem 2.2. Parallel hyperplanes H_i separate the two classes into point clusters, which then move vertically in zigzag until all points are classified.

Remark 3.1. The probability mass function in (3.7) can be expressed in terms of the hypergeometric distribution, whose mass function is defined by

$$H(k; M, K, n) = \frac{\binom{K}{k}\binom{M-K}{n-k}}{\binom{M}{n}},$$

for all $0 \leq M$, $0 \leq K, n \leq M$, and $\max\{0, -M+K+n\} \leq k \leq \min\{K, n\}$. Thus, for each $1 \leq k \leq 2N-1$ we deduce

$$\mathbb{P}(Z_{1,N} = k) = \begin{cases} N(2N-1)^{-1}H(p-1; 2N-2, N-1, N-1), & \text{if } k = 2p-1, \\ (N-1)(2N-1)^{-1}H(p; 2N-2, N-1, N), & \text{if } k = 2p. \end{cases}$$

As a consequence of Lemma 3.1, we get the distribution of $Z^{\perp}_{d,N}$ for any $d \geq 1$:

Corollary 3.1. *Let $Z^{\perp}_{d,N}$ be as defined in* (3.4)*, for some $d, N \geq 1$. For $1 \leq k \leq 2N-1$, we have*

$$\mathbb{P}(Z^{\perp}_{d,N} \geq k) = \left(\sum_{p=\lceil \frac{k+1}{2} \rceil}^{N} \binom{N-1}{p-1}^2 + \sum_{p=\lceil \frac{k}{2} \rceil}^{N-1} \binom{N-1}{p}\binom{N-1}{p-1} \right)^d 2^d \binom{2N}{N}^{-d}. \tag{3.10}$$

Proof. Let $k \in \{1, \ldots, 2N-1\}$. By definition of $Z^{\perp}_{d,N}$, and the fact that $Z^{i}_{d,N}$ are independent and identically distributed to $Z_{1,N}$ for all i, we can compute

$$\mathbb{P}\left(Z^{\perp}_{d,N} \geq k\right) = \mathbb{P}\left(\min_{i=1,\ldots,d} Z^{i}_{d,N} \geq k\right) = \left(\mathbb{P}\left(Z_{1,N} \geq k\right)\right)^d.$$

We conclude the proof by applying Lemma 3.1 to deduce

$$\mathbb{P}\left(Z_{1,N} \geq k\right) = \left(\sum_{p=\lceil \frac{k+1}{2} \rceil}^{N} \binom{N-1}{p-1}^2 + \sum_{p=\lceil \frac{k}{2} \rceil}^{N-1} \binom{N-1}{p}\binom{N-1}{p-1} \right) 2\binom{2N}{N}^{-1}. \qquad \square$$

The results derived in this section are now used to prove Theorem 2.2. For clarity, the whole control method is represented in Figure 3.6 and formalized in Algorithm 1.

Proof of Theorem 2.2. By definition of (3.4), there exists a family of hyperplanes $H_1, \dots, H_z$, with $z = Z^{\perp}_{d,N}(\mathcal{R}, \mathcal{B})$, that are orthogonal to $\mathbf{e}_i$ for some $i \in \{1, \dots, d\}$ and separate $(\mathcal{R}, \mathcal{B})$. Moreover, $Z^{\perp}_{d,N}(\mathcal{R}, \mathcal{B})$ follows the distribution given by (3.10).

Without loss of generality we can assume $i = 1$, so the family of hyperplanes is

$$\left\{H_j \ : \ x^{(1)} = h_j\right\}_{j=1}^{z} \qquad \text{for some } 0 < h_1 < \cdots < h_z < 1.$$

For any fixed $k \in \{2, \dots, d\}$, we define $\tau_{\mathcal{R}} = \{x^{(k)} > 1\}$ and $\tau_{\mathcal{B}} = \{x^{(k)} \leq 1\}$. We then classify the points by clusters based on their $x^{(1)}$-coordinates in ascending order. We can assume that $\{\mathbf{x} \in \mathcal{R} \cup \mathcal{B} : x^{(1)} < h_1\} \subset \mathcal{B}$. If this is not the case, we swap the definitions of $\tau_{\mathcal{R}}$ and $\tau_{\mathcal{B}}$, and also the roles of $\mathcal{R}$ and $\mathcal{B}$ in this proof.

Taking $t_0 = 0$, we build the controls

$$\begin{aligned} (\mathbf{w}, \mathbf{a}, b)(t) &= \sum_{j=1}^{z} (\mathbf{w}_j, \mathbf{a}_j, b_j) \, \mathbb{1}_{[t_{j-1}, t_j)}(t), \\ \text{with} \quad & \mathbf{w}_j = (-1)^{j+1} \mathbf{e}_k, \ \mathbf{a}_j = \mathbf{e}_1, \ b_j = -h_j. \end{aligned} \tag{3.11}$$

Each time horizon $t_j \geq t_{j-1}$ is chosen so that every point $\mathbf{x} \in \mathcal{R} \cup \mathcal{B}$ with $h_j < x^{(1)} < h_{j+1}$ is mapped via $(\mathbf{w}_j, \mathbf{a}_j, b_j)$ to its corresponding target region $\tau_{\mathcal{R}}$ or $\tau_{\mathcal{B}}$. This is possible because the dataset is finite. Moreover, these movements do not affect points that have already been classified, since those satisfy $x^{(1)} < h_j$.

The described method classifies all points according to their labels, and the number of switches is $L = z - 1$. Consequently, L inherits the probability distribution of $Z^{\perp}_{d,N}$. □

Remark 3.2. Theorem 2.2 admits an interpretation in terms of the total variation seminorm, defined by

$$|\theta|_{\mathrm{TV}(0,T)} = \sup_{P} \sum_{i=1}^{|P|} \|\theta(t_i) - \theta(t_{i-1})\|,$$

where the supremum is taken over all finite partitions $P = \{0 = t_0 < t_1 < \cdots < t_{|P|} = T\}$. In particular, for piecewise constant functions we have that $|\cdot|_{\mathrm{TV}(0,T)}$ equals the sum of the magnitudes of its jump discontinuities. Thus, the control θ given by (3.11) will satisfy

$$|\theta|_{\mathrm{TV}(0,T)} \leq L \cdot \max_{1 \leq j \leq L} \sqrt{\|\mathbf{w}_{j+1} - \mathbf{w}_j\|^2 + |b_{j+1} - b_j|^2} \leq \sqrt{5} L.$$

From here, we can estimate the probability distribution of $|\theta|_{\mathrm{TV}(0,T)}$ via

$$\mathbb{P}\left(|\theta|_{\mathrm{TV}(0,T)} \geq \lambda\right) \leq \mathbb{P}\left(L \geq \frac{\lambda}{\sqrt{5}}\right), \qquad \text{for all } \lambda > 0$$

and then apply (2.4).

Algorithm 1 Classification of two N-point sets

```
Require: Two $N$-point sets $\mathcal{R}, \mathcal{B} \subset [0,1]^d$
Ensure: Classification of $\mathcal{R}$ and $\mathcal{B}$
$u \leftarrow e_i$    ▷ Optimal direction for canonical separability
$\tau_{\mathcal{R}} \leftarrow \{x^{(k)} > 1\}, \quad \tau_{\mathcal{B}} \leftarrow \{x^{(k)} < 1\}$    ▷ For any coordinate $k \neq i$
$\mathcal{R} \cup \mathcal{B} \leftarrow \mathrm{Reorder}(\mathcal{R} \cup \mathcal{B};\, i)$    ▷ Sort all points by ascending $i$-th coordinate
$\mathcal{S} \leftarrow \{x_1\}, \quad \mathrm{anchor} \leftarrow 0$
for $j \in \{2, \dots, 2N\}$ do
    if $y_{j-1} = y_j$ then
        $\mathcal{S} \leftarrow \mathcal{S} \cup \{x_j\}$
    else if $x_{j-1} \in \mathcal{B}$ and $x_j \in \mathcal{R}$ then
        while $\mathcal{S} \subset \tau_{\mathcal{R}}$ do
            $\mathcal{R} \cup \mathcal{B} \leftarrow \mathrm{neuralODE}_{T=1}\Big(\mathcal{R} \cup \mathcal{B};\, w = -e_k,\, a = u,\, b = \mathrm{anchor}\Big)$
        end while
        $\mathcal{S} \leftarrow \{x_j\}, \quad \mathrm{anchor} \leftarrow -u \cdot x_{j-1}$
    else if $x_{j-1} \in \mathcal{R}$ and $x_j \in \mathcal{B}$ then
        while $\mathcal{S} \subset \tau_{\mathcal{B}}$ do
            $\mathcal{R} \cup \mathcal{B} \leftarrow \mathrm{neuralODE}_{T=1}\Big(\mathcal{R} \cup \mathcal{B};\, w = e_k,\, a = u,\, b = \mathrm{anchor}\Big)$
        end while
        $\mathcal{S} \leftarrow \{x_j\}, \quad \mathrm{anchor} \leftarrow -u \cdot x_{j-1}$
    end if
end for
```

Remark 3.3. If we allow **a** to be any vector in $\mathbb{R}^d$, then the constraint (3.1) can be removed in Theorem 2.2, ensuring the existence of T, $(\tau_{\mathcal{R}}, \tau_{\mathcal{B}})$ and θ in every case, rather than almost surely. To accomplish this, we choose a new vector basis in which (3.1) holds, and note that the conclusion of Lemma 3.1 remains valid in the new coordinates.

4 Classification by separability in general position

Let $(\mathcal{R}, \mathcal{B})$ be as in (2.1), and assume for now that $|\mathcal{R}| = |\mathcal{B}| = N$. We recall Definition 2.1 and the random variable $Z_{d,N}$, defined in (3.2), to introduce the following quantity:

$$M(d,N) := \max\left\{Z_{d,N}(\mathcal{R}, \mathcal{B}) \,:\, (\mathcal{R}, \mathcal{B}) \text{ as (2.1) in general position, } |\mathcal{R}| = |\mathcal{B}| = N\right\}.$$

Observe that for $d = 1$, any set of points is naturally in general position, so the result from Theorem 2.3 applies directly. Consequently, we have $M(1,N) = 2N - 1$ for all N, as stated in (3.5). However, when $d > 1$, the situation changes significantly. Assuming general position eliminates the pathological configurations described in Theorem 2.3, thereby reducing the maximum possible value of $Z_{d,N}$.

For example, assume that $d = N = 2$. Here, we find $M(2,2) = 2$, which is less than $2 \cdot 2 - 1 = 3$. To show this, consider connecting the two red points with a line r. Since all points are in general position, none of the blue points can lie on r. There are two possible cases:

1. If the blue points are on the same side of r then we can separate $\mathcal{R}$ from $\mathcal{B}$ with one line r' parallel to r, as shown in Figure 4.1 (left).
2. If the blue points are on different sides of r then we can separate $\mathcal{R}$ from $\mathcal{B}$ with two lines r' and r'' parallel to r, as shown in Figure 4.1 (right).

In the example of Figure 4.1 (left), if the lines were restricted to be perpendicular to the canonical axes, two lines would be needed to separate $(\mathcal{R}, \mathcal{B})$. Moreover, if the lines also had to be parallel—as in the framework of Section 3—then three lines would be required.

The argument used in the simple case $(d, N) = (2, 2)$ can be extended to derive a general bound for $M(d, N)$ that shows the improvement over Theorem 2.3 when the points are in general position. To this end, we cover $\mathcal{R}$ with (possibly overlapping) subsets of size d, and then separate $(\mathcal{R}, \mathcal{B})$ by isolating these subsets using hyperplanes under a transversality condition.

Proposition 4.1. *Let $d, N \geq 1$, fix $i \in \{1, \ldots, d\}$ and let $(\mathcal{R}, \mathcal{B})$ be as* (2.1) *in general position, with $|\mathcal{R}| = |\mathcal{B}| = N$. Then, there exist hyperplanes $H_1', H_1'', \ldots, H_{\lceil N/d \rceil}', H_{\lceil N/d \rceil}''$ that separate $(\mathcal{R}, \mathcal{B})$. Moreover, for all $j = 1, \ldots, \lceil N/d \rceil$ the following hold:*

1. *H_j' and H_j'' are parallel;*
2. *H_j' and H_j'' enclose exactly* $\min\{d, |\mathcal{R}|\}$ *points of $\mathcal{R}$ and no points of $\mathcal{B}$.*
3. *H_j' and H_j'' are not orthogonal to $\mathbf{e}_i$;*

In particular, for all $d, N \geq 1$ it follows that

$$M(d, N) \leq 2 \left\lceil \frac{N}{d} \right\rceil . \tag{4.1}$$

Proof. If $d \leq N$, we can choose $\hat{R}_1, \ldots, \hat{R}_{\lceil N/d \rceil} \subset \mathcal{R}$ such that $\mathcal{R} = \hat{R}_1 \cup \cdots \cup \hat{R}_{\lceil N/d \rceil}$, with $|\hat{R}_j| = d$ for all j. Otherwise, build $\hat{R}_1$ by augmenting the set $\mathcal{R}$ with $d - N$ points chosen from $\mathbb{R}^d \setminus \mathcal{B}$ so that $\hat{R}_1 \cup \mathcal{B}$ remains in general position. Since $\mathcal{R} \cup \mathcal{B}$ is in general position, each $\hat{R}_j$ spans a unique hyperplane $\hat{H}_j \subset \mathbb{R}^d$ satisfying

$$(\mathcal{R} \cup \mathcal{B} \setminus \hat{R}_j) \cap \hat{H}_j = \varnothing \qquad \text{and} \qquad \hat{H}_j \neq \hat{H}_k \quad \text{for all } j \neq k. \tag{4.2}$$

For each j, choose two hyperplanes $\hat{H}_j'$ and $\hat{H}_j''$ that are parallel to $\hat{H}_j$ and lie in the two distinct connected components of $\mathbb{R}^d \setminus \hat{H}_j$. If these hyperplanes are chosen sufficiently close to $\hat{H}_j$, then the region between $\hat{H}_j'$ and $\hat{H}_j''$ contains the subset $\hat{R}_j$ and no other points of $\mathcal{R} \cup \mathcal{B}$. Consequently, $(\mathcal{R}, \mathcal{B})$ is separated by the family of pairwise parallel hyperplanes

$$\hat{H}_1', \hat{H}_1'', \ldots, \hat{H}_{\lceil N/d \rceil}', \hat{H}_{\lceil N/d \rceil}'' \subset \mathbb{R}^d.$$

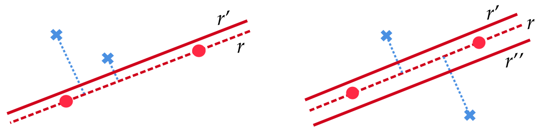

Figure 4.1: Separation of $(\mathcal{R}, \mathcal{B})$ in general position in $\mathbb{R}^2$ with $|\mathcal{R}| = |\mathcal{B}| = 2$ using at most two lines r', r''.

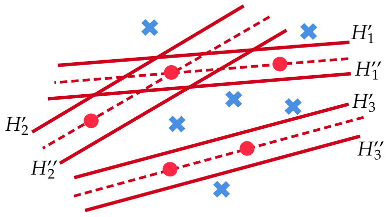

Figure 4.2: Hyperplanes constructed following the proof of Proposition 4.1, separating two unbalanced classes.

By construction, these hyperplanes meet the first two conditions of the statement; however, they may not satisfy the third condition, as some might be orthogonal to $\mathbf{e}_i$. Suppose that exactly $2p$ hyperplanes are orthogonal to $\mathbf{e}_i$ for some $1 \leq p \leq \lceil N/d \rceil$. Without loss of generality, assume these are

$$\hat{H}_1', \hat{H}_1'', \ldots, \hat{H}_p', \hat{H}_p''.$$

Since these hyperplanes are parallel, and different by (4.2), we get $\hat{R}_i \bigcap \hat{R}_j = \varnothing$ for $i \neq j$.

Let $1 \leq j \leq p$. Because $\mathcal{R} \cup \mathcal{B}$ is finite, we can slightly adjust the direction vector of $\hat{H}_j'$ and $\hat{H}_j''$. This yields new parallel hyperplanes H_j' and H_j'' in $\mathbb{R}^d$ that are no longer orthogonal to $\mathbf{e}_i$, yet still enclose exactly R_j, with no other points of $\mathcal{R} \cup \mathcal{B}$ in between. For $p < j \leq \lceil N/d \rceil$, set $H_j' = \hat{H}_j'$ and $H_j'' = \hat{H}_j''$. This yields a new family

$$H_1', H_1'', \ldots, H_p', H_p'', H_{p+1}', H_{p+1}'', \ldots, H_{\lceil N/d \rceil}', H_{\lceil N/d \rceil}'' \subset \mathbb{R}^d$$

that meets all the required conditions. Moreover, this proves (4.1).

□

In the unbalanced case, where $|\mathcal{R}| \neq |\mathcal{B}|$, the method can be similarly applied to isolate the smaller set using pairwise parallel hyperplanes, as shown in Figure 4.2. It follows:

Corollary 4.1. *Let $d \geq 1$. For any dataset $(\mathcal{R}, \mathcal{B})$ as in* (2.1) *in general position, there exists a family of* $2\left\lceil \frac{\min\{|\mathcal{R}|,|\mathcal{B}|\}}{d} \right\rceil$ *hyperplanes that separate the points by labels and satisfy the three conditions of Proposition 4.1, with $\mathcal{R}$ replaced by* $\operatorname{argmin}_{\mathcal{R},\mathcal{B}} \{|\mathcal{R}|, |\mathcal{B}|\}$.

Corollary 4.1 enables the separation of any finite dataset in general position in $\mathbb{R}^d$ into clusters of size d. We now describe an inductive approach to classify these clusters using the dynamics of neural ODEs. First, let us consider the system

$$\dot{\mathbf{x}}(t) = \mathbf{w}(t)\, \sigma_{\text{trun}}(\mathbf{a}(t) \cdot \mathbf{x}(t) + b(t)), \tag{4.3}$$

where the activation function is defined as

$$\sigma_{\text{trun}}(z) := \min\left\{1, (z)_+\right\} = (z)_+ - (z-1)_+, \qquad \text{for } z \in \mathbb{R}. \tag{4.4}$$

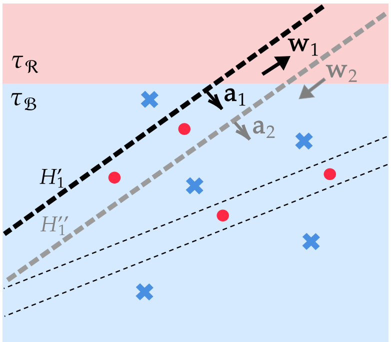

Figure 4.3: Schematic overview of an iteration in the control method used to prove Proposition 4.2. We aim to move the two red points that lie between the hyperplanes H_1' and H_1'' to the region $\tau_{\mathcal{R}}$ through a two-step process. First, we move all points within the half-space $H_1'^{+}$, determined by $\mathbf{a}_1$, in the direction of $\mathbf{w}_1$. Second, we move all points within the half-space $H_1''^{+}$, determined by $\mathbf{a}_2$, in the direction of $\mathbf{w}_1$. The controls are adjusted to ensure that the vector field in $H_1''^{+}$ maintains a unit norm at all times. Consequently, the points in $H_1''^{+}$ return exactly to their original positions after these two steps.

Observe that the flow map associated with (4.3) remains well-defined because σ_{trun} is Lipschitz-continuous. The introduction of σ_{trun} is motivated by its properties, which facilitate the inductive argument. Specifically, in the half-space defined by $\mathbf{a}(t) \cdot \mathbf{x} + b(t) > 1$, the system (4.3) simplifies to $\dot{\mathbf{x}} = \mathbf{w}(t)$. This property can be used to ensure that points already classified will remain so after each inductive step. For more details, see Figure 4.3, which serves as support for the proof, and the formalized control method in Algorithm 2.

Proposition 4.2. *Let $d \geq 2$. For any dataset $(\mathcal{R}, \mathcal{B})$ defined as in* (2.1) *in general position, and any pair of target sets $(\tau_{\mathcal{R}}, \tau_{\mathcal{B}})$ defined as in* (2.3)*, there exist $T > 0$ and a piecewise constant control $\theta \in \Theta_T$ whose number of discontinuities is*

$$L = 2\left\lceil \frac{\min\{|\mathcal{R}|, |\mathcal{B}|\}}{d} \right\rceil - 1,$$

such that the flow map of (4.3) *satisfies $\Phi_T(\mathcal{R}; \theta) \subset \tau_{\mathcal{R}}$ and $\Phi_T(\mathcal{B}; \theta) \subset \tau_{\mathcal{B}}$.*

Proof. Let $i \in \{1, \ldots, d\}$ be fixed and consider $\tau_{\mathcal{R}} = \{x^{(i)} > 1\}$, $\tau_{\mathcal{B}} = \{x^{(i)} \leq 1\}$. The strategy is to evolve $\mathcal{R}$ into the interior of $\tau_{\mathcal{R}}$ while keeping fixed $\mathcal{B}$. We first assume $|\mathcal{R}| = |\mathcal{B}| = N$ and will then extend to the general case $|\mathcal{R}| \neq |\mathcal{B}|$.

By Proposition 4.1, there exist $2\lceil N/d \rceil$ pairwise parallel hyperplanes separating $(\mathcal{R}, \mathcal{B})$. Assume these hyperplanes are defined by

$$H_j' : \mathbf{u}_j \cdot \mathbf{x} + h_j' = 0, \qquad \text{and} \qquad H_j'' : \mathbf{u}_j \cdot \mathbf{x} + h_j'' = 0, \qquad \text{for } j = 1, \ldots, \left\lceil \frac{N}{d} \right\rceil,$$

where $h_j' > h_j''$, and $\mathbf{u}_j \in \mathbb{S}^{d-1}$ satisfies $|\mathbf{u}_j \cdot \mathbf{e}_i| < 1$. Moreover, the region between each pair (H_j', H_j'') encloses a subset $R_j \subset \mathcal{R}$ such that $|R_j| = \min\{d, |\mathcal{R}|\}$, and no other points of $\mathcal{R} \cup \mathcal{B}$ lie between these hyperplanes, namely,

$$(H_j'^{+} \setminus H_j''^{+}) \cap (\mathcal{R} \cup \mathcal{B}) = R_j \subset \mathcal{R} \qquad \text{with } |R_j| = \min\{d, |\mathcal{R}|\},$$

where $H_j'^{+} = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{u}_j \cdot \mathbf{x} + h_j' > 0\}$ and $H_j''^{+} = \{\mathbf{x} \in \mathbb{R}^d : \mathbf{u}_j \cdot \mathbf{x} + h_j'' > 0\}$. Note that $H_j''^{+} \subset H_j'^{+}$ because $h_j' > h_j''$. Taking $t_0 = 0$, we define

$$(\mathbf{w}, \mathbf{a}, b) = \sum_{k=1}^{2\lceil N/d \rceil} (\mathbf{w}_k, \mathbf{a}_k, b_k)\, \mathbb{1}_{(t_{k-1}, t_k)}$$

such that, for $j = 1, \dots, \lceil N/d \rceil$:

1. $(\mathbf{w}_{2j-1}, \mathbf{a}_{2j-1}, b_{2j-1}) = (\mathbf{v}_j, \mathbf{u}_j/d_j', h_j'/d_j')$, where

   - $\mathbf{v}_j \in \mathbb{S}^{d-1}$ satisfies $\mathbf{v}_j \cdot \mathbf{u}_j = 0$ and $\mathbf{v}_j \cdot \mathbf{e}_i > 0$ (for instance, take $\mathbf{v}_j$ to be the normalized projection of $\mathbf{e}_i$ onto the orthogonal subspace $\langle \mathbf{u}_j \rangle^{\perp}$);
   - $d_j' = \min \left\{ \sigma_{\text{trun}}(\mathbf{u}_j \cdot \mathbf{x} + h_j') : \mathbf{x} \in R_j \right\} > 0$.

   Inside the half-space $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{u}_j \cdot \mathbf{x} + h_j' \geq d_j'\} \subset H_j'^{+}$ and over (t_{2j-2}, t_{2j-1}), equation (4.3) becomes $\dot{\mathbf{x}} = \mathbf{v}_j$ with $\mathbf{v}_j \cdot \mathbf{e}_i > 0$. Since the dataset is finite and contained in this half-space, we can choose $t_{2j-1} > t_{2j-2}$ such that $\Phi_{t_{2j-1}}(\mathbf{x}) \in \tau_{\mathcal{R}}$ for all $\mathbf{x} \in R_j$.

2. $(\mathbf{w}_{2j}, \mathbf{a}_{2j}, b_{2j}) = (-\mathbf{v}_j, \mathbf{u}_j/d_j'', h_j''/d_j'')$, where

   - $d_j'' = \min \left\{ \sigma_{\text{trun}}(\mathbf{u}_j \cdot \mathbf{x} + h_j'') : \mathbf{x} \in (\mathcal{R} \cup \mathcal{B}) \cap H_j''^{+} \right\} > 0$.

   Inside the half-space $\{\mathbf{x} \in \mathbb{R}^d : \mathbf{u}_j \cdot \mathbf{x} + h_j'' \geq d_j''\} \subset H_j''^{+}$ and over (t_{2j-1}, t_{2j}), equation (4.3) becomes $\dot{\mathbf{x}} = -\mathbf{v}_j$. Now, we set $t_{2j} = 2t_{2j-1} - t_{2j-2}$ so that

   $$\Phi_{t_{2j}}(\mathbf{x}) = \Phi_{t_{2j-2}}(\mathbf{x}) \qquad \text{for all } \mathbf{x} \in (\mathcal{R} \cup \mathcal{B}) \cap H_j''^{+}.$$

   All the while, we have

   $$\Phi_{t_{2j}}(\mathbf{x}) = \Phi_{t_{2j-1}}(\mathbf{x}) \qquad \text{for all } \mathbf{x} \in (\mathcal{R} \cup \mathcal{B}) \cap H_j''^{-}.$$

For $T = t_{2\lceil N/d \rceil}$, we conclude $\Phi_T(\mathcal{R}) \subset \tau_{\mathcal{R}}$ and $\Phi_T(\mathcal{B}) = \mathcal{B} \subset \tau_{\mathcal{B}}$, with $L = 2\lceil N/d \rceil - 1$.

If $|\mathcal{R}| < |\mathcal{B}|$, we apply the same argument to obtain $\Phi_T(\mathcal{R}) \subset \tau_{\mathcal{R}}$ and $\Phi_T(\mathcal{B}) = \mathcal{B} \subset \tau_{\mathcal{B}}$ with $L = 2\lceil |\mathcal{R}|/d \rceil - 1$, by virtue of Corollary 4.1. If $|\mathcal{R}| > |\mathcal{B}|$, we swap the roles of $\mathcal{R}$ and $\mathcal{B}$ and the definitions of $\tau_{\mathcal{R}}$ and $\tau_{\mathcal{B}}$, with $L = 2\lceil |\mathcal{B}|/d \rceil - 1$. □

The following lemma formalizes an idea from [33, section 2.2]. It shows that any flow of (4.3) can be represented as the composition of two flows of (2.2), if $\mathbf{w}$ and $\mathbf{a}$ are orthogonal:

Lemma 4.1. *Let $\theta = (\mathbf{w}, \mathbf{a}, b) \in \mathbb{S}^{d-1} \times \mathbb{R}^{d+1} \times \mathbb{R}$ with $\mathbf{w} \cdot \mathbf{a} = 0$. Then, $\theta_1 = (\mathbf{w}, \mathbf{a}, b)$ and $\theta_2 = (-\mathbf{w}, \mathbf{a}, b - 1)$ satisfy*

$$\Phi_t(\Phi_t(\cdot\,; \theta_1)\,; \theta_2) = \Phi_t^{\text{trun}}(\cdot\,; \theta) \qquad \textit{for all } t > 0,$$

where Φ_t and Φ_t^{trun} are the flow maps of (2.2) *and* (4.3), *respectively.*

Algorithm 2 Classification of two point sets via (4.3)

```
Require: Two finite sets $\mathcal{R}, \mathcal{B} \subset [0,1]^d$ in general position; Direction $e_i$
Ensure: Classification of $\mathcal{R}$ and $\mathcal{B}$
$\mathcal{S} \leftarrow \arg\min\{|\mathcal{R}|, |\mathcal{B}|\}, \quad \tau_{\mathcal{S}} \leftarrow \{x^{(i)} > 1\}$
$\mathcal{C} \leftarrow \mathrm{cover}(\mathcal{S}, d, \lceil |\mathcal{S}|/d \rceil)$    ▷ Covering for $\mathcal{S}$ by $\lceil |\mathcal{S}|/d \rceil$ subsets $\mathcal{S}_j$ with $|\mathcal{S}_j| = d$
for $(j, \mathcal{S}_j)$ in enumerate($\mathcal{C}$) do
    $H_j(a_j, b_j) = \{x : a_j \cdot x + b_j = 0, \ \|a_j\| = 1\} \leftarrow \mathrm{Span}(\mathcal{S}_j)$
    if $|a_j \cdot e_i| = 1$ then
        $a_j \leftarrow \mathrm{NormalizedPerturbation}(a_j)$    ▷ Achieves $|a_j \cdot e_i| < 1, \ \|a_j\| = 1$
    end if
    $\delta_j \leftarrow \min\{\mathrm{dist}(x, H_j) : x \in (\mathcal{R} \cup \mathcal{B}) \setminus \mathcal{S}_j\}$
    $b'_j \leftarrow b_j + 0.5\,\delta_j, \quad b''_j \leftarrow b_j - 0.5\,\delta_j$
    $w_j \leftarrow \mathrm{NormalizedProjection}_{\langle a_j \rangle^\perp}(e_i)$
    $d'_j = \min_{x \in \mathcal{S}_j} \sigma_{\mathrm{trun}}(a_j \cdot x + b'_j)$
    $d''_j = \min_{x \in (\mathcal{R} \cup \mathcal{B}) \cap \{x : a_j \cdot x + b''_j > 0\}} \sigma_{\mathrm{trun}}(a_j \cdot x + b''_j)$
    while $\mathcal{S}_j \not\subset \tau_{\mathcal{S}}$ do
        $\mathcal{R} \cup \mathcal{B} \leftarrow \mathrm{neuralODE}_{T=1}(\mathcal{R} \cup \mathcal{B}\,; w = w_j, \ a = a_j/d'_j, \ b = b'_j/d'_j)$
        $\mathcal{R} \cup \mathcal{B} \leftarrow \mathrm{neuralODE}_{T=1}(\mathcal{R} \cup \mathcal{B}\,; w = -w_j, \ a = a_j/d''_j, \ b = b''_j/d''_j)$
    end while
end for
```

Proof. Since $\mathbf{w} \cdot \mathbf{a} = 0$, it holds

$$\frac{d}{dt}(\mathbf{a} \cdot \mathbf{x}(t) + b) = \mathbf{a} \cdot \dot{\mathbf{x}}(t) = \mathbf{a} \cdot \mathbf{w}\sigma(\mathbf{a} \cdot \mathbf{x}(t) + b) = 0.$$

Thus, $\sigma(\mathbf{a} \cdot \mathbf{x}(t) + b)$ is constant for all t. Suppose $\mathbf{a} \cdot \mathbf{x} + b \leq 1$. Then

$$\Phi_t^{\mathrm{trun}}(\mathbf{x}\,;\,\theta) = \mathbf{x} + t\mathbf{w}(\mathbf{a} \cdot \mathbf{x} + b) = \Phi_t(\mathbf{x}\,;\,\theta_1) = \Phi_t(\Phi_t(\mathbf{x}\,;\,\theta_1)\,;\,\theta_2).$$

Otherwise, if $\mathbf{a} \cdot \mathbf{x} + b > 1$,

$$\begin{aligned} \Phi_t(\Phi_t(\mathbf{x}\,;\,\theta_1)\,;\,\theta_2) &= \Phi_t(\mathbf{x} + t\mathbf{w}(\mathbf{a} \cdot \mathbf{x} + b)\,;\,\theta_2) \\ &= \mathbf{x} + t\mathbf{w}(\mathbf{a} \cdot \mathbf{x} + b) - t\mathbf{w}(\mathbf{a} \cdot (\mathbf{x} + t\mathbf{w}(\mathbf{a} \cdot \mathbf{x} + b)) + b - 1) \\ &= \mathbf{x} + t\mathbf{w} = \Phi_t^{\mathrm{trun}}(\mathbf{x}\,;\,\theta). \end{aligned}$$

□

We can now use Lemma 4.1, together with Proposition 4.2, to demonstrate Theorem 2.1.

Proof of Theorem 2.1. By virtue of Proposition 4.2, there exist $\bar{T} > 0$ and a piecewise constant control $\bar{\theta} = (\bar{\mathbf{w}}, \bar{\mathbf{a}}, \bar{b}) \in \Theta_{\bar{T}}$, which presents $2\lceil \min\{|\mathcal{R}|, |\mathcal{B}|\}/d \rceil - 1$ discontinuities, such that

$$\Phi_{\bar{T}}^{\mathrm{trun}}(\mathcal{R}; \bar{\theta}) \subset \tau_{\mathcal{R}} \qquad \text{and} \qquad \Phi_{\bar{T}}^{\mathrm{trun}}(\mathcal{B}; \bar{\theta}) \subset \tau_{\mathcal{B}},$$

where Φ_t^{trun} is the flow map of (4.3). Moreover, $\bar{\mathbf{w}}(t) \cdot \bar{\mathbf{a}}(t) = 0$ for all t, so by Lemma 4.1 there exists a piecewise constant $\theta \in \Theta_{2\bar{T}}$ with $4\lceil \min\{|\mathcal{R}|, |\mathcal{B}|\}/d \rceil - 1$ discontinuities, such that $\Phi_{\bar{T}}^{\text{trun}}(\cdot\,; \bar{\theta}) = \Phi_{2\bar{T}}(\cdot\,; \theta)$. We conclude by defining $T = 2\bar{T}$. □

Remark 4.1. Similarly to Remark 3.2, we can extract from Theorem 2.1 a bound for the total variation of the controls as

$$|\theta|_{\text{TV}(0,T)} \leq 2L\|\theta\|_\infty \leq 2\sqrt{1 + \|(\mathbf{a}, b)\|_\infty^2} \left(4 \left\lceil \frac{\min\{|\mathcal{R}|, |\mathcal{B}|\}}{d} \right\rceil - 1 \right),$$

where $\|(\mathbf{a}, b)\|_\infty$ depends on the minimum separation between points of $\mathcal{R} \cup \mathcal{B}$ and the maximum distance of any point to the origin.

The structure of pairwise parallel hyperplanes defined in Proposition 4.1 can be exploited for control when combined with certain architectures. We propose *triangular neural ODEs*

$$\dot{\mathbf{x}}(t) = \mathbf{w}(t)\, \sigma_{\text{Tr}}(\mathbf{a}(t) \cdot \mathbf{x}(t) + b(t)), \tag{4.5}$$

where the activation σ_{Tr} is defined as

$$\sigma_{\text{Tr}}(z) = \max\{1 - |z - 1|, 0\}, \qquad \text{for } z \in \mathbb{R}. \tag{4.6}$$

The function σ_{Tr} remains globally Lipschitz, ensuring that the flow map is well-defined.

By replacing σ_{trun} with σ_{Tr}, we can improve upon Proposition 4.2. Specifically, we can halve the number of values taken by the control θ. Adapting the proof is straightforward: Step 2 can be omitted, as the support of σ_{Tr} is restricted to the interval $(0, 2)$.

Corollary 4.2. *Let $d \geq 2$. For any dataset $(\mathcal{R}, \mathcal{B})$ defined as in* (2.1) *in general position, and any pair of target sets $(\tau_{\mathcal{R}}, \tau_{\mathcal{B}})$ defined as in* (2.3)*, there exist $T > 0$ and a piecewise constant control $\theta \in \Theta_T$ whose number of discontinuities is*

$$L = \left\lceil \frac{\min\{|\mathcal{R}|, |\mathcal{B}|\}}{d} \right\rceil - 1,$$

such that the flow map of the neural ODE (4.5) *satisfies $\Phi_T(\mathcal{R}; \theta) \subset \tau_{\mathcal{R}}$ and $\Phi_T(\mathcal{B}; \theta) \subset \tau_{\mathcal{B}}$.*

Remark 4.2. Our separability-based methodology aligns with existing research in discrete geometry. In [35], the authors establish lower and upper bounds on the minimum number of hyperplanes required to individually separate N points in general position in $\mathbb{R}^d$. In [36] it is shown that finding such number for an arbitrary point set is an NP-complete problem. From a computational perspective, efficient algorithms for separability in low-dimensional spaces are proposed in [37,38].

Remark 4.3. The VC dimension of a classification model is defined as the size of the largest set of points that can be shattered, meaning any arbitrary binary assignment of labels is possible [39]. We can interpret Theorem 2.1 in terms of the VC dimension of our model. Specifically, we can establish a lower bound: by fixing the maximum number of discontinuities in the controls to $L \geq 1$, our method can shatter any set of Ld points in $\mathbb{R}^d$.

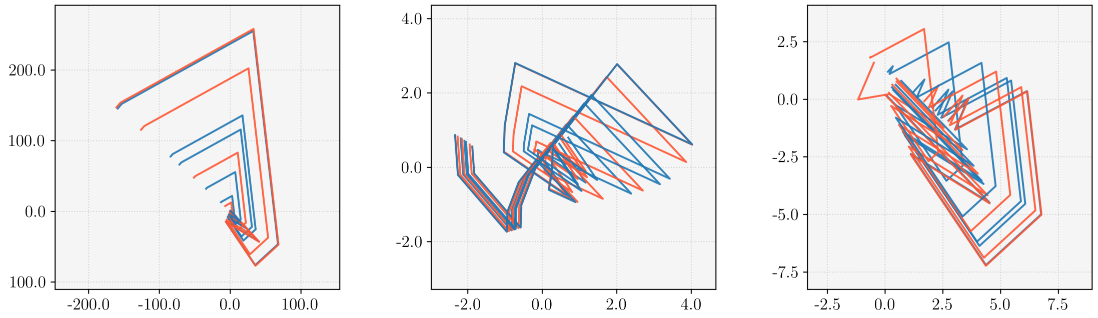

Figure 5.1: Trajectories for $\sigma = \text{ReLU}$ exhibit an exponential drift when $\mathbf{a}\cdot\mathbf{x}+b>1$ (left). If $\sigma = \tanh$ (center) or $\sigma = \sigma_{\text{trun}}$ (right), this drift is mitigated because $\|\sigma\|_\infty \leq 1$. Here, we have used $N=5$, $L=10$, and $T=60$.

5 Numerics

We present a computational test[1] designed to evaluate the capacity of neural ODEs for binary classification. Specifically, our objective is to numerically estimate the minimal complexity they require to classify an arbitrary dataset of a fixed size, and compare it with the complexity we used in Algorithm 1.

Let $\mathscr{D} = \{(\mathbf{x}_n, y_n)\} \subset \mathbb{R}^d \times \{1, 0\}$ be a finite dataset and $\mathcal{R} = \{\mathbf{x}_n : (\mathbf{x}_n, 1) \in \mathscr{D}\}$, $B = \{\mathbf{x}_n : (\mathbf{x}_n, 0) \in \mathscr{D}\}$ be such that $|\mathcal{R}| = |\mathcal{B}| = N$. For a fixed $L \geq 0$ and time $T > 0$, we aim to find piecewise constant controls $\theta \in \Theta_T$ with L discontinuities (or switches) such that the flow map of the neural ODE satisfies

$$\Phi_T(\mathcal{R};\theta) \subset \left\{x^{(1)} > 1\right\} \qquad \text{and} \qquad \Phi_T(\mathcal{B};\theta) \subset \left\{x^{(1)} \leq 1\right\}. \tag{5.1}$$

We opt for σ_{trun} as in (4.4) and equation (4.3) to better visualize the results (see Figure 5.1). In this case, Proposition 4.2 establishes that $L = 2\lceil N/d \rceil - 1$ switches ensure any dataset in general position can be classified. This condition is generically satisfied when all points are sampled from a non-singular probability measure, such as $U([0,1]^d)$.

We design our experimental setup in the following way:

Data. We use ten random datasets, each consisting of $N = 30$ red points ($\mathcal{R}$) and $N = 30$ blue points ($\mathcal{B}$), sampled from $U([0,1]^d)$ for various dimensions d ranging from 2 to $2N$.

Model. We consider equation (4.3) with controls that are piecewise constant over $L+1$ time intervals of length $\Delta t = T(L+1)^{-1}$, where $T = 60$. To approximate the solution, we use an explicit Euler discretization scheme with a step size of $0.25\Delta t$. The control values are initialized using Kaiming uniform initialization; that is, each component is sampled from $U([-\alpha,\alpha])$, where α is the inverse of the number of input units in the weight tensor.

Error. To enforce the correct classification of all points as defined by (5.1), we introduce

[1] The code, implemented in PyTorch [40], is available at `https://github.com/antonioalvarezl/2024-WCS-NODEs`

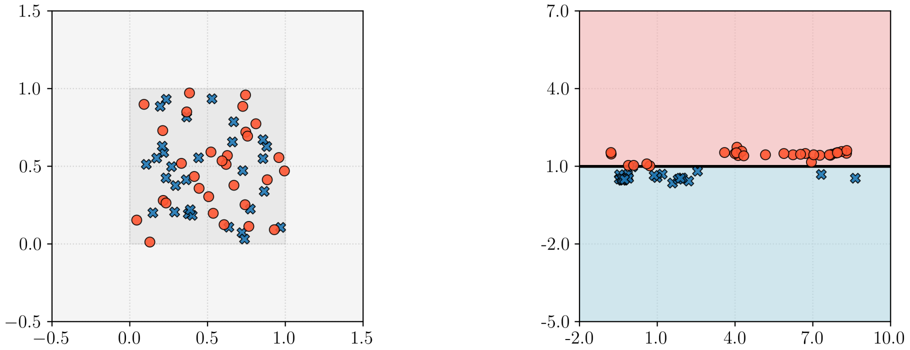

Figure 5.2: Left: Initial data comprising 30 red points and 30 blue points in $[0,1]^2$. Right: Final positions at time $T = 60$, having fixed $L = 37$ switches.

the following margin-based loss function:

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{R}|} \sum_{\mathbf{x}_n \in \mathcal{R}} \operatorname{dist}\left(\Phi_T(\mathbf{x}_n;\theta), \{x^{(1)} < 1.5\}\right)^2 + \frac{1}{|\mathcal{B}|} \sum_{\mathbf{x}_n \in \mathcal{B}} \operatorname{dist}\left(\Phi_T(\mathbf{x}_n;\theta), \{x^{(1)} > 0.5\}\right)^2$$

Training. Optimization is conducted using the Adam optimizer [41] with a learning rate of 0.01, which was determined to be optimal through a grid search over a range of values.
Procedure. First, for each d, we verify that setting $L = 2\lceil N/d \rceil - 1$ switches is sufficient to classify all ten datasets. Then, we gradually decrease the value of L until at least one of the datasets fails to be classified, according to the stopping criteria defined below.
Stopping criteria. If condition (5.1) is met, classification is successful and the training is stopped. Conversely, we consider the following three failure stopping criteria:

1. The maximum number of 70000 epochs is reached.

2. Slow convergence, if $\mathcal{L} \geq 0.15$ at 20000 epochs or $\mathcal{L} \geq 0.1$ at 40000 epochs, or if the minimum error does not decrease over 5000 consecutive epochs.

3. Local minima detection, if the maximum relative error over 50 consecutive epochs exceeds a threshold of 10^{-20}.

Close. We conclude that the model with L switches does not have the capacity to classify 60 points if any of the failure stopping criteria is met in 20 randomized initializations of the parameters for any of the ten datasets.

The experiments were conducted using specific seed settings to ensure reproducibility, and the results are shown in Figure 5.3. We observe that $L = 2\lceil N/d \rceil - 1$ is typically close to the optimal value L_* obtained through training. Specifically, for $d \in \{7, 10, 15, 20, 60\}$, we get $L_* = 2\lceil N/d \rceil - 1$. For $d \in \{2, 30\}$, we even find that $L_* > 2\lceil N/d \rceil - 1$.

We observed that training requires an increasing number of parameter initializations in lower dimensions. This tendency is particularly pronounced when $d = 2$, as shown

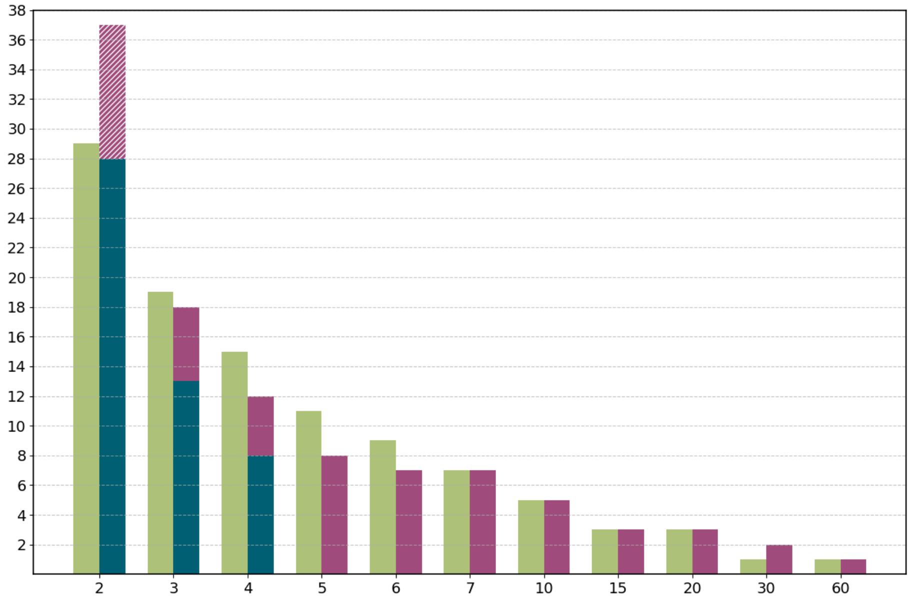

Figure 5.3: L versus d for fixed $N = 30$. Green bars represent the number of switches in Proposition 4.2, given by $2\lceil N/d \rceil - 1$. Purple bars show the minimum number of switches required by gradient-based training to successfully classify all datasets. Dark bars indicate the minimum number of switches found after data rescaling.

by the purple bar with a hatched pattern. In that case, only five out of ten datasets were successfully classified using $L = 37$ switches. We were unable to find any higher value of L that could classify the remaining five datasets.

Motivated by the difficulties found in low dimensions, we applied a preprocessing step to improve the separability of the data points. Specifically, for $d \in \{2, 3, 4\}$, we standardized the data by subtracting the mean and scaling to unit variance. This reduces the risk of the algorithm becoming trapped in poor local minima due to an overly dense point cloud. The results are represented by the blue bars of Figure 5.3.

In summary, we observed that the value of $L = 2\lceil N/d \rceil - 1$ switches given in Proposition 4.2 is a suboptimal bound that closely approximates the optimal value obtained via gradient descent, especially for large d with fixed N.

Conclusions

In this work, we have explored the capacity of neural ODEs to classify an arbitrary dataset, using the framework of simultaneous control of clusters of points and geometric separability techniques to define these clusters.

In our main result, we have established a new bound of $L = 4\lceil \min\{|\mathcal{R}|, |\mathcal{B}|\}/d \rceil$ control switches that guarantees classification of any dataset given by $(\mathcal{R}, \mathcal{B})$ in $\mathbb{R}^d$, using the single-neuron model (2.2). The only assumption is that all points are in general position (see Definition 2.1), a condition that is satisfied almost surely by any finite set sampled from a non-singular measure. Our result improves the previous bound of $L = 3\min\{|\mathcal{R}|, |\mathcal{B}|\}$, by leveraging the better separability that points present in higher dimensions. From a technical perspective, we demonstrate that any subset of d points can be isolated between two parallel hyperplanes within any finite dataset in general position. We have complemented this result with some numerical experiments that supported the optimality of our bound, particularly in high dimensions.

Our second result acknowledges that maximal complexity is rarely necessary, as data points of the same class are often initially close together, which facilitates their classification. We develop a new control method to derive the probability distribution of L when all data points are i.i.d. and both classes have fixed size $N \geq 1$. Our method once again emphasizes the advantage of high dimensionality. Specifically, we deduce that if $d \gtrsim 2^N / \sqrt{N}$ then classification using an autonomous neural ODE (characterized by $L = 0$) is possible with high probability. Additionally, we have characterized the pathological configurations in which the maximum value of $L = 2N - 2$ occurs.

Our results can be seen as a manifestation of the *blessing of dimensionality* [42], a phenomenon where machine learning problems become more tractable in high dimensions. This contrasts with the well-known curse of dimensionality, which occurs when data sparsity in higher dimensions leads to an exponential increase in computational complexity and data requirements.

We now discuss some connections and extensions of our results:

- **Multiclass classification**. For any $S \geq 2$, let $\mathcal{R}_1, \dots, \mathcal{R}_S \subset \mathbb{R}^d$ be finite subsets whose union is in general position. Suppose $|\mathcal{R}_S| = \max |\mathcal{R}_i|$. We first address binary classification between $\mathcal{R}_1$ and $\mathcal{R}_2 \cup \cdots \cup \mathcal{R}_S$ using Proposition 4.2. Once these two sets are linearly separable under the flow map Φ_T, we proceed to classify $\Phi_T(\mathcal{R}_2)$ against $\Phi_T(\mathcal{R}_1 \cup \mathcal{R}_3 \cup \cdots \cup \mathcal{R}_S)$ in a similar manner. Since Algorithm 2 consistently fixes the larger of the two sets, we can proceed inductively and ensuring that ultimately every pair of distinct subsets will be linearly separable under the flow map.

- **Alternative activation functions**. Theorem 2.2 applies broadly to any Lipschitz-continuous activation function $\sigma : \mathbb{R} \to \mathbb{R}$, as long as there exist $-\infty \leq a < b \leq \infty$ such that $\sigma(z) = 0$ for all $z \in (a, b)$ and $\sigma \neq 0$. The choice of σ would only affect the control time $T > 0$. Following the terminology of reference [22], such a σ would be referred to as a well function, and our proofs could be extended using an argument based on affine-invariance of system (2.2).

  Generalizing Theorem 2.1 to other activations presents additional challenges since they must represent any flow of the truncated ReLU (as in Lemma 4.1). A modification (and improvement) of Theorem 2.1 is Corollary 4.2, where we consider the triangular activation function. In fact, any activation function σ with $\operatorname{supp}(\sigma) \subset (a, b)$ for some $-\infty < a < b < +\infty$ allows for the same extension.

- **Alternative loss functions**. While we have used a simplified error function—the sum of the distances of all points to their respective target regions—cross-entropy loss ℓ_{CE} is the standard choice for classification problems in practice. In binary classification,

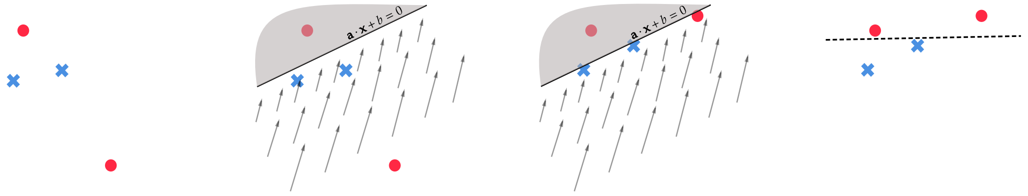

Figure 5.4: In the first picture, at least two hyperplanes are required to separate both classes ("static" classification). In the other three pictures, the same points evolve according to $\dot{\mathbf{x}} = \mathbf{w}(\mathbf{a} \cdot \mathbf{x} + b)_+$ (with constant $(\mathbf{w}, \mathbf{a}, b)$) and can eventually be separated using only one hyperplane ("dynamic" classification).

the cross entropy loss associated with a data point $(\mathbf{x}_n, y_n) \in \mathscr{D}$ is defined as

$$\ell_{\text{CE}}(\mathbf{x}_n, y_n) = -y_n \log(\hat{y}_n) - (1 - y_n)\log(1 - \hat{y}_n),$$

where $\hat{y}_n = (\text{softmax} \circ P \circ \Phi_T(\mathbf{x}_n))^{(1)}$ is the predicted probability that $y_n = 1$. This prediction is obtained by applying a linear transformation $P : \mathbb{R}^d \to \mathbb{R}^2$ and then performing component-wise normalization using the softmax function.

For any fixed P, our methods can be adapted to minimize this loss while maintaining the same complexity L. For example, if $d = 2$ and P is the identity matrix, then we can achieve $\ell_{\text{CE}}(\mathbf{x}_n, y_n) \to 0$ when $(-1)^{y_n} \Phi_T(\mathbf{x}_n) \cdot (-1, 1) \to +\infty$. To accomplish this, we can set $x^{(1)} = x^{(2)}$ as the decision boundary and increase the time T; however, this would not require additional switches.

- **Point separability**. Since our control methods fundamentally rely on identifying hyperplanes that separate two classes (mostly via combinatorics), an important question arises: does dynamic classification using neural ODEs essentially reduce to static separability using hyperplanes? The short answer is *no*; they are distinct frameworks.

  On one hand, a collection of separating hyperplanes does not necessarily translate into effective neural ODE dynamics. In other words, having fixed $\mathbf{a}(t)$ and $b(t)$, there may be no control $\mathbf{w}(t)$ that yields the desired classification via (2.2). Our constructions require those hyperplanes to follow a specific structure (e.g., pairwise parallel in Theorem 2.1), making it possible to determine a suitable $\mathbf{w}(t)$. However, for a random family of separating hyperplanes, finding such $\mathbf{w}(t)$ can be highly complex.

  It might then seem that dynamic classification with neural ODEs is more restrictive than linear classification methods. Yet this is also false: neural ODEs can classify without explicitly relying on separating hyperplanes, that is, there exist controls $\mathbf{a}(t)$ and $b(t)$ that lead to successful classification even if the hyperplanes given by $\mathbf{a}(t) \cdot \mathbf{x} + b(t) = 0$ do *not* fully separate both classes. In Figure 5.4 we illustrate a scenario where neural ODEs outperform linear methods.

  Our results address these questions by proving that, in the worst-case scenario (i.e., for any arbitrary dataset), neural ODEs can perform at least as well as linear classifiers. Furthermore, our numerical experiments seem to confirm that, under such worst-case conditions, the performance of neural ODEs trained via gradient descent matches our theoretical findings.

- **Discrete neural networks**. In the discrete setting, the capacity to control N points in $\mathbb{R}^d$ with a given level of complexity has been more broadly studied and often referred to as finite-sample expressivity. Early studies, focused on sigmoid activation functions, showed that networks with one hidden layer could perform this task using $O(N)$ neurons [43]. With the rise of deep models and the ReLU activation, researchers explored how adding depth could reduce the required complexity [44, 45], and existent results were extended to ReLU networks, see [46]. For deep feed-forward networks, [47] solved the problem using $O(\sqrt{N})$ neurons.

  In the past decade, the focus has shifted towards new architectures like convolutional networks [48], or ResNets [49], which achieve the goal using $O(N \log N)$ hidden nodes, assuming a minimum distance between points. This bound was later improved to $O(N/d)$ in [47], under the assumption of general position. Notably, for conditional networks, the complexity can be reduced to $O(\log N)$ neurons [50].

Future work

Several important directions still need to be explored. Below, we present some of them.

1. **Improvement or sharpness of the bound**. A first and natural question is whether the new bound we have introduced for the minimal number of switches L required to classify any dataset in general position is sharp or can be further improved.

   On one hand, it turns out that equality in (4.1) can be achieved in some simple cases, for example, when $d = N = 2$, as illustrated in Figure 4.1. However, our algorithm does not fully exploit the capacity of the parameter space because the piecewise constant controls $\mathbf{a}(t)$ and $\mathbf{w}(t)$ repeat certain values throughout the algorithm. Moreover, the algorithm is designed to iteratively classify clusters of d points but does not consider the possibility of classifying larger clusters, unlike Algorithm 2.

   If our bound for L is indeed sharp, a related problem is to identify the geometry of those configurations where this bound is attained, similar to Theorem 2.3.

2. **Decision boundary**. In this study, we have restricted the decision boundary to be any hyperplane defined by the equation $x^{(i)} = 1$ for $i \in \{1, \ldots, d\}$. Our control methods, detailed in Algorithms 1 and 2, can easily be adapted to every hyperplane in $\mathbb{R}^d$. However, the classification problem may also be addressed using more complex hypersurfaces than hyperplanes or by treating the decision boundary itself as an optimizable parameter. This leads to the open question of determining the optimal decision boundary that minimizes complexity while maximizing training accuracy.

3. **Generalization**. The ultimate goal of supervised learning models is to make accurate predictions on new, unseen data that were not part of the training dataset. While our

methods aim to determine the minimum complexity required for a model to achieve zero training error, this often leads to overfitting and poor performance on test data.

Proper generalization requires to model to capture the geometric patterns in the configuration of data points, rather than perfectly classifying every individual point regardless of their distribution. We must mention, however, that recent studies suggest overfitting does not necessarily imply poor generalization [51]. Understanding the relationship between both concepts remains an active area of research [52].

In practice, generalization is typically achieved by adding a regularization term to the loss function. In the context of simultaneous control, some outlier data points could be misclassified to reduce the control cost. Developing constructive control methods that intentionally sacrifice some accuracy on the training data to improve the model's ability to generalize remains an open question.

4. **Combination with self-attention**. Transformers are deep architectures that alternate between attention and feed-forward layers, adding residual connections. A highly simplified continuous-time version is the interacting particle system given by:

$$\dot{\mathbf{x}}_i(t) = \mathbf{w}(t)\sigma(\mathbf{a}(t)\cdot\mathbf{x}_i(t) + b(t)) + \sum_{j=1}^{N} \frac{e^{\mathbf{x}_i(t)\cdot B(t)\mathbf{x}_j(t)}}{\sum_{\ell=1}^{N} e^{\mathbf{x}_i(t)\cdot B(t)\mathbf{x}_\ell(t)}} V(t)\mathbf{x}_j(t).$$

Here, the vector $\mathbf{x}_i(t)$ represents the position of the i-th particle (or data point) at time t, whereas $V(t), B(t) \in \mathbb{R}^{d\times d}$ are additional controls. The attention term induces a clustering effect among the data points, causing them to concentrate towards certain limiting configurations, as studied in [53]. An open question is to quantify the reduction in complexity achieved by controlling this mechanism to cluster points by classes as a preprocessing step for classification. A constructive control of continuous-time transformers has been carried out in [54] for the mean-field formulation of this system restricted to the sphere.

Acknowledgments

This work has been supported by the Madrid Government (Comunidad de Madrid, Spain) under the multiannual Agreement with UAM in the line for the Excellence of the University Research Staff in the context of the V PRICIT (Regional Programme of Research and Technological Innovation) and by PID2023-146872OB-I00 of MICIU (Spain). AÁ has been funded by FPU21/05673 from the Spanish Ministry of Universities. RO has been funded by Severo Ochoa Programme for Centres of Excellence in R&D (CEX2023-001347-S) of MICIU (Spain). EZ has been funded by the Alexander von Humboldt-Professorship program, the ModConFlex Marie Curie Action, HORIZON-MSCA-2021-DN-01, the COST Action MAT-DYN-NET, the Transregio 154 Project "Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks" of the DFG and TED2021-131390B-I00 of MICINN (Spain).

References

[1] Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

[2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155, 2003.

[4] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(7):179–188, 1936.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[8] Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5:1–11, 02 2017.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[10] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 6572–6583. Curran Associates Inc., 2018.

[11] Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

[12] Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes. *Advances in Neural Information Processing Systems*, 33:3952–3963, 2020.

[13] Domènec Ruiz-Balet and Enrique Zuazua. Neural ODE control for classification, approximation, and transport. *SIAM Rev.*, 65(3):735–773, 2023.

[14] Antonio Álvarez-López, Arselane Hadj Slimane, and Enrique Zuazua. Interplay between depth and width for interpolation in neural odes. *Neural Networks*, page 106640, 2024.

[15] David Mumford, John Fogarty, and Frances Kirwan. *Geometric Invariant Theory*. Springer Berlin, 1994.

[16] Thomas M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, EC-14(3):326–334, 1965.

[17] Thomas M. Cover. The Number of Linearly Inducible Orderings of Points in d -Space. *SIAM Journal on Applied Mathematics*, 15(2):434–439, March 1967.

[18] Eduardo D. Sontag. Shattering All Sets of ‘k’ Points in “General Position” Requires (k — 1)/2 Parameters. *Neural Computation*, 9(2):337–348, February 1997.

[19] Paulo Tabuada and Bahman Gharesifard. Universal approximation power of deep residual neural networks through the lens of control. *IEEE Trans. Autom. Control.*, 68(5):2715–2728, 2023.

[20] Christa Cuchiero, Martin Larsson, and Josef Teichmann. Deep neural networks, generic universal interpolation, and controlled ODEs. *SIAM J. Math. Data Sci.*, 2(3):901–919, 2020.

[21] Karthik Elamvazhuthi, Xuechen Zhang, Samet Oymak, and Fabio Pasqualetti. Learning on manifolds: Universal approximations properties using geometric controllability conditions for neural odes. In *Learning for Dynamics and Control Conference*, Proceedings of Machine Learning Research, pages 1–11, 2023.

[22] Qianxiao Li, Ting Lin, and Zuowei Shen. Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, 25(5):1671–1709, 2022.

[23] Jingpu Cheng, Qianxiao Li, Ting Lin, and Zuowei Shen. Interpolation, approximation and controllability of deep neural networks, 2023.

[24] Isao Ishikawa, Takeshi Teshima, Koichi Tojo, Kenta Oono, Masahiro Ikeda, and Masashi Sugiyama. Universal Approximation Property of Invertible Neural Networks. *Journal of Machine Learning Research*, 24(287):1–68, 2023.

[25] Hongzhou Lin and Stefanie Jegelka. Resnet with one-neuron hidden layers is a universal approximator. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 6172–6181, 2018.

[26] Weinan E, Jiequn Han, and Qianxiao Li. A mean-field optimal control formulation of deep learning. *Research in the Mathematical Sciences*, 6(1):10, December 2018.

[27] Qianxiao Li, Long Chen, Cheng Tai, and Weinan E. Maximum principle based algorithms for deep learning. *Journal of Machine Learning Research*, 18(165):1–29, 2018.

[28] Benoît Bonnet, Cristina Cipriani, Massimo Fornasier, and Hui Huang. A measure theoretical approach to the mean-field maximum principle for training NeurODEs. *Nonlinear Analysis*, 227:113161, February 2023.

[29] Noboru Isobe and Mizuho Okumura. Variational formulations of ode-net as a mean-field optimal control problem and existence results. *Journal of Machine Learning*, 3(4):413–444, 2024.

[30] Borjan Geshkovski and Enrique Zuazua. Turnpike in optimal control of pdes, resnets, and beyond. *Acta Numerica*, 31:135–263, 2022.

[31] Carlos Esteve-Yagüe and Borjan Geshkovski. Sparsity in long-time control of neural ODEs. *Systems Control Lett.*, 172:Paper No. 105452, 14, 2023.

[32] Domènec Ruiz-Balet, Elisa Affili, and Enrique Zuazua. Interpolation and approximation via momentum resnets and neural odes. *Systems & Control Letters*, 162:105182, 2022.

[33] Domènec Ruiz-Balet and Enrique Zuazua. Control of neural transport for normalising flows. *Journal de Mathématiques Pures et Appliquées*, 181:58–90, 2024.

[34] Wlodzislaw Duch. K-separability. In *Artificial Neural Networks - ICANN 2006*, volume 4131 of *Lecture Notes in Computer Science*, pages 188–197, 2006.

[35] Ralph P. Boland and Jorge Urrutia. Separating collections of points in Euclidean spaces. *Inform. Process. Lett.*, 53(4):177–183, 1995.

[36] Robert Freimer, Joseph S. B. Mitchell, and Christine Piatko. On the Complexity of Shattering Using Arrangements. Technical Report, Cornell University, USA, March 1991.

[37] Esther M. Arkin, Ferran Hurtado, Joseph S. B. Mitchell, Carlos Seara, and Steven S. Skiena. Some lower bounds on geometric separability problems. *Internat. J. Comput. Geom. Appl.*, 16(1):1–26, 2006.

[38] Michael F. Houle. Algorithms for weak and wide separation of sets. *Discrete Applied Mathematics*, 45(2):139–159, 1993.

[39] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[40] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS 2017 Workshop on Autodiff*, 2017.

[41] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[42] David Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 01 2000.

[43] S.-C. Huang and Y.-F. Huang. Bounds on the number of hidden neurons in multilayer perceptrons. *IEEE Transactions on Neural Networks*, 2(1):47–55, 1991.

[44] Ronen Eldan and Ohad Shamir. The Power of Depth for Feedforward Neural Networks. *JMLR: Workshop and Conference Proceedings*, 49:1–34, 2015.

[45] Guang-Bin Huang. Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, 14(2):274–281, 2003.

[46] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *Communications of the ACM*, 64, 11 2016.

[47] Chulhee Yun, Suvrit Sra, and Ali Jadbabaie. Small relu networks are powerful memorizers: A tight analysis of memorization capacity. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, page 15558–15569. Curran Associates Inc., 2019.

[48] Quynh Nguyen and Matthias Hein. Optimization Landscape and Expressivity of Deep CNNs. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3730–3739. PMLR, July 2018.

[49] Moritz Hardt and Tengyu Ma. Identity matters in deep learning. In *International Conference on Learning Representations*, 2017.

[50] Erdem Koyuncu. Memorization capacity of neural networks with conditional computation. In *The Eleventh International Conference on Learning Representations*, 2023.

[51] Peter L. Bartlett, Philip M. Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[52] Hongkang Yang. A mathematical framework for learning probability distributions. *Journal of Machine Learning*, 1(4):373–431, 2022.

[53] Borjan Geshkovski, Cyril Letrouit, Yury Polyanskiy, and Philippe Rigollet. The emergence of clusters in self-attention dynamics. *Advances in Neural Information Processing Systems*, 36:57026–57037, December 2023.

[54] Borjan Geshkovski, Philippe Rigollet, and Domènec Ruiz-Balet. Measure-to-measure interpolation using Transformers, November 2024. arXiv:2411.04551 [cs, math, stat].