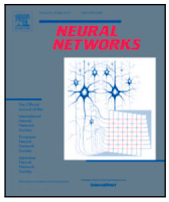
Neural Networks 180 (2024) 106640

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet



Full Length Article

Interplay between depth and width for interpolation in neural ODEs

Antonio Álvarez-López [a,c,*], Arselane Hadj Slimane [b], Enrique Zuazua [a,c,d]

[a] Universidad Autónoma de Madrid, Departamento de Matemáticas, C. Francisco Tomás y Valiente, 7, Madrid, 28049, Spain
[b] ENS Paris Saclay, Avenue des sciences, 4, Gif-sur-Yvette, 91190, France
[c] Friedrich-Alexander-Universität Erlangen-Nürnberg, Department of Mathematics, Chair for Dynamics, Control, Machine Learning, and Numerics (Alexander von Humboldt Professorship), Cauerstraße, 11, Erlangen, 91058, Germany
[d] Fundación Deusto, Av. de las Universidades, 24, Bilbao, 48007, Spain

ARTICLE INFO

Keywords:
Neural ODEs
Depth
Width
Simultaneous controllability
Transport control
Wasserstein distance

ABSTRACT

Neural ordinary differential equations have emerged as a natural tool for supervised learning from a control perspective, yet a complete understanding of the role played by their architecture remains elusive. In this work, we examine the interplay between the width p and the number of transitions between layers L (corresponding to a depth of L+1). Specifically, we construct explicit controls interpolating either a finite dataset $\mathcal{D}$, comprising N pairs of points in $\mathbb{R}^d$, or two probability measures within a Wasserstein error margin $\varepsilon > 0$. Our findings reveal a balancing trade-off between p and L, with L scaling as $1 + O(N/p)$ for data interpolation, and as $1 + O\left(p^{-1} + (1+p)^{-1}\varepsilon^{-d}\right)$ for measures.

In the high-dimensional and wide setting where $d, p > N$, our result can be refined to achieve $L = 0$. This naturally raises the problem of data interpolation in the autonomous regime, characterized by $L = 0$. We adopt two alternative approaches: either controlling in a probabilistic sense, or by relaxing the target condition. In the first case, when $p = N$ we develop an inductive control strategy based on a separability assumption whose probability increases with d. In the second one, we establish an explicit error decay rate with respect to p which results from applying a universal approximation theorem to a custom-built Lipschitz vector field interpolating $\mathcal{D}$.

1. Introduction

Residual neural networks (ResNets) can be defined, in their simplest form, as the family of discrete systems

$$\begin{cases} \mathbf{x}(k+1) &= \mathbf{x}(k) + W(k)\boldsymbol{\sigma}\left(A(k)\mathbf{x}(k) + \mathbf{b}(k)\right), \\ \mathbf{x}(0) &= \mathbf{x}_0 \in \mathbb{R}^d, \end{cases} \tag{1.1}$$

with $k = 0, \dots, L$, $(W(k), A(k), \mathbf{b}(k)) \in \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times d} \times \mathbb{R}^p$, for some $d \geq 1$, $L \geq 0$ and $p \geq 1$. Each time step k is a *layer* of the network, and the number of layers $L + 1$ is known as its *depth*. The parameter p, which identifies the number of neurons per layer, is known as the *width*. The activation function $\boldsymbol{\sigma} : \mathbb{R}^p \to \mathbb{R}^p$ is defined as the column vector $\boldsymbol{\sigma}(\mathbf{y}) = (\sigma(y^{(1)}), \dots, \sigma(y^{(p)}))$ from a fixed nonlinear function $\sigma : \mathbb{R} \to \mathbb{R}$. We will consider the ReLU activation, given by $\sigma(z) = \max\{z, 0\}$, $z \in \mathbb{R}$.

It has been noted (Chang, Meng, Haber, Tung, & Begert, 2018; Chen, Rubanova, Bettencourt, & Duvenaud, 2018; E, 2017; Haber & Ruthotto, 2017) that (1.1) corresponds to the forward Euler scheme for the family of continuous-time models known as neural ordinary differential equations (ODEs):

$$\begin{cases} \dot{\mathbf{x}}(t) &= W(t)\boldsymbol{\sigma}\left(A(t)\mathbf{x}(t) + \mathbf{b}(t)\right), \\ \mathbf{x}(0) &= \mathbf{x}_0 \in \mathbb{R}^d, \end{cases} \tag{1.2}$$

with $(W, A, \mathbf{b}) \in L^\infty\left((0,T), \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times d} \times \mathbb{R}^p\right)$ for some fixed $T > 0$. If σ is Lipschitz then existence and uniqueness of solutions hold for every initial datum $\mathbf{x}_0$.

In (1.2), the time $t \in (0, T)$ parameterizes the evolution of the state $\mathbf{x}$ through a continuous range of layers. It is common to take $(W, A, \mathbf{b})$ as piecewise constant over $(0, T)$ with a finite number of discontinuities L. This approach aligns the dynamics more closely with those of (1.1) and makes the parameter space finite-dimensional, which simplifies optimization (see Massaroli, Poli, Park, Yamashita, & Asama, 2020) and provides a practical way to quantify complexity of the model using L, as in Alvarez-Lopez, Orive-Illera, and Zuazua (2023), Ruiz-Balet and Zuazua (2023).

We thus define depth in neural ODEs as the number of distinct values of $(W, A, \mathbf{b})$, given by $L + 1$, which is a measure of complexity.

* Corresponding author.
E-mail addresses: antonio.alvarezl@uam.es (A. Álvarez-López), arselane.hadj_slimane@ens-paris-saclay.fr (A.H. Slimane), enrique.zuazua@fau.de (E. Zuazua).

https://doi.org/10.1016/j.neunet.2024.106640
Received 6 February 2024; Received in revised form 29 July 2024; Accepted 14 August 2024
Available online 19 August 2024

After discretization, (1.2) becomes a ResNet with $L+1$ layers, each repeating multiple times according to the discretization step size. Thus, the effective number of layers increases at the discrete level but $L+1$ remains a good measure of its macroscopic complexity.

Eq. (1.2) can be equivalently written as

$$\dot{\mathbf{x}} = \sum_{i=1}^{p} \mathbf{w}_i(t)\sigma(\mathbf{a}_i(t)\cdot\mathbf{x} + b_i(t)), \tag{1.3}$$

where, for $i = 1,\dots,p$, $\mathbf{w}_i(t), \mathbf{a}_i(t) \in \mathbb{R}^d$ are the columns of W and the rows of A, respectively, and $b_i(t)$ is the ith coordinate of $\mathbf{b}$. We will adopt formulation (1.3), although for simplicity, we denote the assembled parameters $(\mathbf{w}_i, \mathbf{a}_i, b_i)_{i=1}^p$ using the matrix form $(W, A, \mathbf{b})$.

We can extend neural ODEs in a natural way to handle probability distributions rather than just points in $\mathbb{R}^d$. By interpreting its right-hand side as the advection field that drives the evolution of a measure ρ, we obtain the so-called neural transport equation (Ruiz-Balet & Zuazua, 2024):

$$\partial_t \rho + \mathrm{div}_{\mathbf{x}}\left(\sum_{i=1}^{p} \mathbf{w}_i\, \sigma(\mathbf{a}_i\cdot\mathbf{x} + b_i)\rho\right) = 0. \tag{1.4}$$

Prior research indicates that control theory offers significant potential for examining the properties of neural ODEs, for instance, via optimal control (E, 2017; Esteve, Geshkovski, Pighin, & Zuazua, 2021; Esteve-Yagüe & Geshkovski, 2023) or geometric control techniques (Agrachev & Sarychev, 2021; Scagliotti, 2023; Tabuada & Gharesifard, 2023). A fundamental problem still open is to develop a comprehensive understanding of the roles played by depth and width with respect to the expressive power of the model, see Fan, Lai, and Wang (2020), Hardt and Ma (2017), Lu, Pu, Wang, Hu, and Wang (2017). This property is often evaluated by its capacity to interpolate either a finite set of point pairs or two given probability measures.

The first concept, commonly referred to as finite-sample expressivity (Yun, Sra, & Jadbabaie, 2019), relates to the approximation power of the model, as discussed in Li, Lin, and Shen (2022). It essentially amounts to a simultaneous control problem for the flow map at time T of (1.2), defined by

$$\Phi_T(\cdot; W, A, \mathbf{b}) : \mathbf{x}_0 \in \mathbb{R}^d \longmapsto \mathbf{x}(T; \mathbf{x}_0) \in \mathbb{R}^d,$$

where $\mathbf{x}(t;\mathbf{x}_0)$ is the solution of (1.2). The goal is to find a control function $(W, A, \mathbf{b})$ such that Φ_T maps N given input points to N corresponding target points in $\mathbb{R}^d$. Throughout this work, each pair input-target will be represented by a solid circle and an empty circle, respectively, of a particular color.

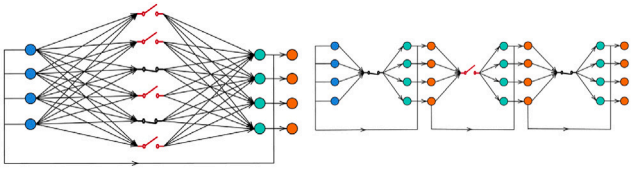
In the second scenario, we aim to control the transport dynamics described by (1.4) in order to evolve a given initial density ρ_0 into a target density ρ_T. This task is highly relevant in generative modeling of synthetic data via normalizing flows (Kobyzev, Prince, & Brubaker, 2021; Papamakarios, Nalisnick, Rezende, Mohamed, & Lakshminarayanan, 2021). We approach it as an approximate control problem in the Wasserstein-q metric space for $q \geq 1$, extending prior work focused on W_2, see Elamvazhuthi, Gharesifard, Bertozzi, and Osher (2022), or in W_1 with $p = 1$, see Ruiz-Balet and Zuazua (2023).

Our main objective is to develop a comprehensive theory of interpolation for the family of models described by (1.3), linking the error to the specific architecture (p, L). Both numerical and theoretical studies suggest that deeper networks usually achieve better performance (Eldan & Shamir, 2015; Huang, 2003; Mhaskar, Liao, & Poggio, 2017), a tendency particularly evident in training (Yun et al., 2019; Zhang, Bengio, Hardt, Recht, & Vinyals, 2016).

Understanding the balance between width and depth is thus vital for optimal design of neural ODEs. As we vary L and p, two limiting models emerge:

Shallow neural ODEs. $L = 0$ is fixed, while the width p can be as large as required:

$$\dot{\mathbf{x}} = \sum_{i=1}^{p} \mathbf{w}_i \sigma\left(\mathbf{a}_i\cdot\mathbf{x} + b_i\right), \tag{1.5}$$



(A) Shallow ResNet (B) Narrow ResNet

Fig. 1. Qualitative representation of models (1.5) and (1.6) as discrete systems. Blue, green and orange circles represent respectively the input $\mathbf{x}$, the residual term $W\sigma(A\mathbf{x}+\mathbf{b})$ output of each layer, respectively. Switches represent activation functions.

where $\{(\mathbf{w}_i, \mathbf{a}_i, b_i)\}_{i=1}^p \subset \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}$ are constant controls, making the equation autonomous. The field on the right-hand side of (1.5) corresponds to a one hidden layer neural network with d components. The approximation capacity of this class of functions has been extensively studied (see Cybenko, 1989; DeVore, Hanin, & Petrova, 2021; Pinkus, 1999). The discrete version of (1.5) can be identified with a one hidden layer ResNet, as seen in Fig. 1(a).

Narrow neural ODEs. $p = 1$ is fixed, while the depth $L+1$ can be as large as required:

$$\dot{\mathbf{x}} = \mathbf{w}(t)\sigma\left(\mathbf{a}(t)\cdot\mathbf{x} + b(t)\right), \tag{1.6}$$

where $(\mathbf{w}, \mathbf{a}, b) \in L^\infty\left((0,T), \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}\right)$. The ability of this model to interpolate data and approximate functions has been explored in Lin and Jegelka (2018), Ruiz-Balet and Zuazua (2023). It offers the advantage of easier construction of explicit controls compared to (1.5), owing to its simplified dynamics, albeit at the expense of increased depth, which scales with the cardinal N of the dataset. The discrete version of (1.6) corresponds to a deep ResNet with one neuron per hidden layer, so it alternates layers of dimension 1 and d, as seen in Fig. 1(b).

Developing a unified theory that bridges shallow and narrow neural ODEs would combine the vast work done for (1.5) with the intuitive dynamics of (1.6). Moreover, it would facilitate the optimal design of a neural ODE through the strategic choice of depth and width. This entails optimizing the complexity κ, defined as the total number of parameters in (1.3):

$$\kappa := (L+1)\times p \times (2d+1). \tag{1.7}$$

Indeed, on each of the $L+1$ hidden layers, p neurons of dimension $2d+1$ need to be determined.

1.1. Roadmap

In Section 2, we present the main results of our work in two parts. First, in Section 2.1 we study the problem of interpolating a finite dataset with (1.3), understood as the property of simultaneous control for neural ODEs. Second, in Section 2.2, we approach the approximate controllability of probability measures using the dynamics of (1.4). In Section 3, we discuss the main implications of our work and pose some open questions. In Section 4, we prove the main results and provide the necessary tools as lemmas.

1.2. Notation

- We use subscripts to identify the particular elements from a dataset and superscripts for the coordinates of a vector. In addition, (column) vectors are denoted with bold letters and matrices with capital letters.
- We denote by $\mathbf{x}\cdot\mathbf{y}$ the scalar product of $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.
- We denote by $\lceil z \rceil$ the lowest integer greater than or equal to $z \in \mathbb{R}$, and by $\lfloor z \rfloor$ the highest integer lower than or equal to z.
- We denote by $\mathbb{S}^{d-1}$ the $(d-1)$-dimensional sphere in $\mathbb{R}^d$.

- We denote by $\mathrm{Lip}\left(\mathbb{R}^d, \mathbb{R}^d\right)$ the space of Lipschitz-continuous vector fields in the usual norm, and by L_V the Lipschitz constant of each $\mathbf{V} \in \mathrm{Lip}\left(\mathbb{R}^d, \mathbb{R}^d\right)$.
- Given any Borel measure μ in $\mathbb{R}^d$ and continuous function $f : \mathbb{R}^d \to \mathbb{R}^d$, we denote by $f_{\#}\mu$ the push-forward measure, defined for each Borel $A \subset \mathbb{R}^d$ by

$$f_{\#}\mu(A) = \mu(f^{-1}(A)).$$

- We denote by $\mathcal{P}^c_{ac}(\mathbb{R}^d)$ the space of compactly supported, absolutely continuous probability measures on $\mathbb{R}^d$.

2. Main results

2.1. Simultaneous control

Let $N \geq 1$ and $d \geq 2$, and consider a dataset

$$D = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathbb{R}^d \times \mathbb{R}^d \tag{2.1}$$

with $\mathbf{x}_n \neq \mathbf{x}_m$ and $\mathbf{y}_n \neq \mathbf{y}_m$ for all $n \neq m$. First, we study the finite-sample expressivity of the general model (1.3), recast as a problem of simultaneous control.

Problem-Definition. For any fixed $T > 0$ and $p \geq 1$, find controls

$$\left\{(\mathbf{w}_i, \mathbf{a}_i, b_i)\right\}_{i=1}^p \subset L^\infty\left((0,T); \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}\right),$$

such that the flow map Φ_T generated by (1.3) *interpolates* the dataset D, i.e., it drives each data point from its initial position $\mathbf{x}_n$ to its target $\mathbf{y}_n$. This is fulfilled when

$$\Phi_T(\mathbf{x}_n; W, A, \mathbf{b}) = \mathbf{y}_n \qquad \text{for all } n = 1, \dots, N.$$

Note that neural ODEs allow for a time rescaling from T to any $T^* > 0$, i.e., for $s := tT^{-1}T^*$ it holds

$$\Phi_T\left(\cdot; (W, A, \mathbf{b})(t)\right) = \Phi_{T^*}\left(\cdot; \left(\frac{T}{T^*}W, A, \mathbf{b}\right)(s)\right). \tag{2.2}$$

If σ is positively homogeneous of degree 1, like ReLU, we can further write:

$$\Phi_T\left(\cdot; (W, A, \mathbf{b})(t)\right) = \Phi_{T^*}\left(\cdot; \left(\frac{\gamma T}{T^*}W, \gamma^{-1}A, \gamma^{-1}\mathbf{b}\right)(s)\right)$$

for all $\gamma > 0$. Loosely speaking, (2.2) shows that time and velocity (or control norm) can be interchanged when covering a fixed distance.

Our first result provides a relationship between L and p that ensures the interpolation of D at any time T:

Theorem 1. *Let $N \geq 1$, $d \geq 2$ and $T > 0$ be fixed. Consider the dataset D as defined in (2.1). For any $p \geq 1$, there exists a piecewise constant control*

$$(W, A, \mathbf{b}) \in L^\infty\left((0,T); \mathbb{R}^{p\times d} \times \mathbb{R}^{p\times d} \times \mathbb{R}^p\right)$$

such that the flow $\Phi_T(\cdot; W, A, \mathbf{b})$ generated by (1.3) interpolates D, i.e.,

$$\Phi_T(\mathbf{x}_n; W, A, \mathbf{b}) = \mathbf{y}_n, \qquad \text{for all } n = 1, \dots, N.$$

Furthermore, the number of discontinuities of $(W, A, \mathbf{b})$ is

$$L = 2\lceil N/p \rceil - 1. \tag{2.3}$$

Let us briefly describe the algorithm. First, we pivot around the $x^{(1)}$-coordinate and control the remaining $d-1$ coordinates. Consequently, the trajectory of each initial point $\mathbf{x}_n$ is confined within the hyperplane defined by $x^{(1)} = x_n^{(1)}$. Then, we pivot using the controlled coordinates to adjust $x^{(1)}$. This algorithm requires a depth of $2\lceil N/p \rceil$ layers, which is independent of the dimension d, since a constant control suffices to simultaneously steer $d-1$ coordinates in the first step, assuming $p > N$.

Remark 1. If $\mathbf{y}_n = \mathbf{y}_m$ for some $m \neq n$ in (2.1), then interpolation is not achievable due to the uniqueness of solutions in (1.3). In such cases, we can relax the statement to approximate controllability by applying Theorem 1 to an ϵ-perturbation of the targets, for any fixed $\epsilon > 0$.

Remark 2. The number of switches L in Theorem 1 is independent of the chosen time horizon T. Specifically, property (2.2) ensures that, for any $T^* > 0$, we can construct some controls that interpolate D using L discontinuities. In this way, the final time affects the estimates only through its trade-off with the control size $\|W\|$. This duality can be exploited to obtain more accurate estimates of the gap between optimal and explicit controls for exact or approximate controllability. For instance, in Fernández-Cara and Zuazua (2000) a similar analysis was conducted in the context of linear parabolic equations.

Remark 3. Theorem 1 is broadly applicable to any globally Lipschitz, nonlinear activation function, provided there exist $-\infty \leq a < b \leq \infty$ such that

$$\sigma(z) = 0, \qquad \text{for all } z \in (a, b). \tag{2.4}$$

Indeed, in Li et al. (2022), any σ satisfying (2.4) is referred to as a "well function", and is used as the key tool to approximate functions in L^p through simultaneous control.

For Lipschitz activations that do not satisfy (2.4), such as $\sigma(z) = \tanh(z)$, our method is not applicable. Moreover, to the best of our knowledge, a constructive control algorithm has not yet been developed for such cases. However, it has been established in Cheng, Li, Lin, and Shen (2023) that (1.2) possesses simultaneous controllability (or the universal interpolation property) as long as σ is globally Lipschitz and nonlinear.

Relaxation to locally Lipschitz activations satisfying (2.4), such as $\sigma^k(z) = \max\{z, 0\}^k$ (for $k \geq 1$), is more delicate since Φ_T might not be well-defined. In general, finite-time blow-ups can occur, meaning that each target should be sufficiently close to the corresponding initial datum to ensure feasibility (Cheng et al., 2023). Our algorithm, however, is based on the use of divergence-free fields, so it could be adapted for these cases.

In (2.3) we can see that, as p increases, the number of discontinuities L decreases with the same rate, meaning that width and depth play a similar role in the steering. Nevertheless, a result on the optimal design of the models for our particular algorithm can be derived:

Corollary 2. *For the family of controls given by Theorem 1 that ensure interpolation of D, the minimal complexity is*

$$\kappa_{min} = (4d + 2)(N + 1),$$

obtained when $p = 1$, i.e., when the neural ODE takes the form (1.6).

The complete transition from the narrow model (1.6) to the shallow model (1.5), characterized by $L = 0$ is not attained in (2.3). Due to the division into two steps in the proposed algorithm, whenever $p > N$ the selected control will exhibit a single switch ($L = 1$), reaching a two-layer architecture, rather than the autonomous ansatz (1.5). The restriction naturally raises the question of whether simultaneous control is possible in shallow neural ODEs (1.5). For this task, a reconsideration of the algorithm presented in Ruiz-Balet and Zuazua (2023) becomes necessary. In the high dimensional setting, and more precisely, when the dimension exceeds the number of data points ($d > N$), we can refine the statement of Theorem 1 to include the case $L = 0$:

Corollary 3. *Let $N \geq 1$, $d \geq 2$ with $d > N$, and $T > 0$ be fixed. Consider the dataset D as defined in (2.1). For any $p \geq 1$, there exists a piecewise constant control*

$$(W, A, \mathbf{b}) \in L^\infty\left((0,T); \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times d} \times \mathbb{R}^p\right)$$

such that the flow $\Phi_T(\cdot; W, A, \mathbf{b})$ generated by (1.3) interpolates D. Furthermore, the number of discontinuities of $(W, a, \mathbf{b})$ is

$$L = 2\left(\lceil N/p \rceil - 1\right).$$

The key idea is that, when $d > N$, the first step in the proof of Theorem 1 can be suppressed. This is done by transforming the $x^{(1)}$-axis so that each $\mathbf{x}_n$ shares the same first coordinate with $\mathbf{y}_n$, or, equivalently, by optimally reorienting the hyperplanes represented in Fig. 5(a). For more insights on the proof, see Fig. 6 in Section 4.

Remark 4. Corollary 3 suggests ideas similar to those in Esteve et al. (2021, Theorem 5.1). In that result, interpolation is established for $d \geq N$ in a simplified neural ODE, when $\sup_{n=1,\dots,N} |\mathbf{x}_n - \mathbf{y}_n| < \varepsilon$ for a sufficiently small $\varepsilon > 0$, under a geometric assumption on the images of the targets through $\boldsymbol{\sigma}$. Moreover, an estimation of the control cost is obtained, which is linear with respect to ε. The generation of new synthetic coordinates until $d \geq N$ is not typically a problem, as discussed in Dupont, Doucet, and Teh (2019), where the technique of embedding the dataset in $\mathbb{R}^d \times \{0, \dots, 0\}$ is proposed and its computational advantages are studied.

In practice, N tends to be larger than d. In that case, interpolation with constant controls can be obtained for $p = N$ under a certain separability hypothesis on D:

Assumption 1. Let $D = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathbb{R}^d \times \mathbb{R}^d$ as defined in (2.1). There exist a vector $\mathbf{a} \in \mathbb{S}^{d-1}$, a permutation τ of N elements and a sequence

$$-\infty < b_{N+1} < b_N < \cdots < b_1 < \infty$$

such that, for all $n = 1, \dots, N-1$:

$$-b_n < \mathbf{a} \cdot \mathbf{x}_{\tau(n)} < -b_{n+1} \quad \text{and} \quad -b_n < \mathbf{a} \cdot \mathbf{y}_{\tau(n)} < -b_{n+1}.$$

Assumption 1 claims that we can diagonally separate each pair $(\mathbf{x}_n, \mathbf{y}_n)$ from the rest. This is achieved by defining $N+1$ parallel hyperplanes $H_n = \{\mathbf{a} \cdot \mathbf{x} + b_n = 0\}$ such that the strip S_n bounded by H_n and H_{n+1} contains only the point $\mathbf{x}_n$ and its target $\mathbf{y}_n$, as shown in Fig. 2(a). While the hypothesis might seem overly restrictive, it is noteworthy that if the points are randomly sampled from a compact set, the probability that the condition is fulfilled converges to 1 when the dimension grows:

Proposition 4. *Let $\mu \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ such that the random variables $\pi_i X$ are independent and identically distributed for $i = 1, \dots, d$, where $X \sim \mu$ and π_i is the canonical projection on the ith coordinate. If every $\mathbf{x}_n$ and $\mathbf{y}_n$ in D is sampled from μ, and N is sufficiently large, then the probability P that Assumption 1 is satisfied is bounded as*

$$1 - \left[1 - \frac{1}{\sqrt{2}} \left(\frac{e}{2N}\right)^N\right]^d \leq P \leq 1.$$

The hypothesis of Proposition 4 are fulfilled by the uniform probability measure in any hypercube, or by any isotropic Gaussian distribution.

Under Assumption 1, we can build a constant control such that the flow of (1.5), taking a width $p = N$, interpolates D. Our result can be seen as the version of Zhang et al. (2016, Theorem 1) for neural ODEs, under Assumption 1. A representation of the trajectories can be seen in Fig. 2(b).

Corollary 5. *Consider a dataset $D \subset \mathbb{R}^d \times \mathbb{R}^d$ for $d \geq 2$, under Assumption 1. For any fixed $T > 0$, there exists a control $(W, A, \mathbf{b}) \in \mathbb{R}^{d\times N} \times \mathbb{R}^{N\times d} \times \mathbb{R}^N$ such that the flow Φ_T generated by (1.5) interpolates D.*

All in all, new strategies are required to study simultaneous control in the autonomous model (1.5) under general conditions. A natural starting point to assess the problem's feasibility is to relax it by admitting an error $\varepsilon > 0$, and use density tools provided by universal approximation theorems (UATs); see Cybenko (1989), Pinkus (1999). In deep learning, UATs establish the density of neural networks in certain function spaces, usually over compact domains. The decay rate

(A) Example of setup described in assumption 1, for $\mathbf{a} = \mathbf{e}_1$. (B) Trajectories for exact control, following transversal fields.

Fig. 2. Representation of the control algorithm followed in Corollary 5.

of the error in relation to the number of parameters of the network has been quantified for certain spaces (Bach, 2017; DeVore et al., 2021). Specifically, these studies bound the uniform error decay rate when the target function is Lipschitz continuous in a compact domain.

We will approach the UAT, often interpreted in a static manner, from our dynamic control perspective. In this regard, shallow neural ODEs provide a vector field that transitions initial data to final data, with its flow at time T approximating the target function. First, we establish the existence of a time-independent field whose integral curves guide each input point $\mathbf{x}_n$ in D to its corresponding target $\mathbf{y}_n$ within a fixed time T. This field is constructed based on purely geometric considerations (see Fig. 3) and can be chosen to be Lipschitz continuous.

Proposition 6. *Let $N \geq 1$, $d \geq 2$ and $T > 0$ be fixed. Consider the dataset $D \subset \mathbb{R}^d \times \mathbb{R}^d$ as defined in (2.1), and any compact subset $\Omega \subset \mathbb{R}^d$ such that $\mathrm{Int}(\Omega)$ is connected and $D \subset \mathrm{Int}(\Omega) \times \mathrm{Int}(\Omega)$. Then, there exists a vector field $\mathbf{V} \in \mathrm{Lip}\left(\mathbb{R}^d, \mathbb{R}^d\right)$ such that the flow $\Psi_{T,\mathbf{V}}$ of the equation*

$$\dot{\mathbf{x}} = \mathbf{V}(\mathbf{x})$$

interpolates D, and the N curves given by

$$C_n := \{\Psi_{t,\mathbf{V}}(\mathbf{x}_n) : t \in [0, T]\} \qquad (n = 1, \dots, N),$$

are contained in $\mathrm{Int}(\Omega)$.

The subset $\Omega \subset \mathbb{R}^d$, which will serve as our domain of approximation, can always be established as $\Omega = [-R, R]^d$ for a sufficiently large $R > 0$. Consequently, for any dataset $D \subset \mathbb{R}^d \times \mathbb{R}^d$, there exists a field $\mathbf{V} \in \mathrm{Lip}(\mathbb{R}^d, \mathbb{R}^d)$ whose integral curves C_n interpolate D, i.e., the space

$$\mathcal{V}_D := \left\{\mathbf{V} \in \mathrm{Lip}\left(\mathbb{R}^d, \mathbb{R}^d\right) : \Psi_{T,\mathbf{V}} \text{ interpolates } D\right\}$$

is non-empty. Moreover, we can define

$$L_0 := \inf_{\mathbf{V} \in \mathcal{V}_D} L_V,$$

so L_0 only depends on D and the chosen domain of approximation Ω. It suffices then to combine the UAT from DeVore et al. (2021) (see Lemma 12 in Section 4) with classical results on the stability of ODEs to obtain the following theorem:

Theorem 7. *Let $N \geq 1$, $d \geq 2$ and $T > 0$ be fixed. Consider the dataset D as defined in (2.1). For each $p \geq 1$, there exists a control $(W, A, \mathbf{b}) \in \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times d} \times \mathbb{R}^p$ such that the flow Φ_T generated by (1.5) satisfies*

$$\sup_{i=1,\dots,N} |\mathbf{y}_n - \Phi_T(\mathbf{x}_n; W, A, \mathbf{b})| \leq C_{d,\mathcal{L},T} \frac{\log_2(\kappa)}{\kappa^{1/d}}, \tag{2.5}$$

where $\kappa = (d+2)dp$ is the complexity of the NODE, and

$$C_{d,\mathcal{L},T} = C_{d,L_0} T \exp(\mathcal{L}T),$$

being $\mathcal{L} = \min\{L_0, \|W\|\|A\|\}$ where $\|\cdot\|$ is the spectral norm, and $C_{d,L_0} > 0$ a constant depending on d and L_0 but independent of κ.

Remark 5. The argument employed in this theorem extends beyond neural networks. Since we solely rely on a density result that provides

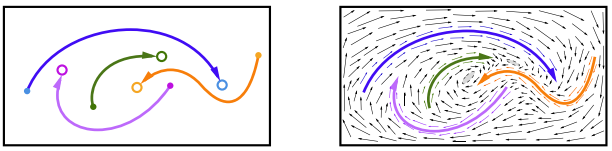


Fig. 3. Construction of the Lipschitz field $\mathbf{V}$ in Proposition 6 which interpolates $\mathcal{D}$ inside a compact set Ω that contains all the paths.

a convergence rate, other dense families of functions like polynomials, trigonometric, finite element methods or wavelets could also be considered, with their corresponding convergence rates.

Remark 6. Given a domain of approximation Ω, for the bound (2.5) to be optimal, it is natural to pose the problem of finding the interpolating field $\mathbf{V} \in \mathcal{V}_{\mathcal{D}}$ which has the smallest possible Lipschitz constant L_V within Ω.

Remark 7. When $d \geq 3$, the construction of a field $\mathbf{V} \in \mathcal{V}_{\mathcal{D}}$ is generally very simple. Since, in that case, two arbitrary curves are unlikely to intersect, we can generally consider the N segments that connect each pair $(\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{D}$ and build $\mathbf{V}$ as one of the piecewise constant fields having these segments as integral curves. Selecting the optimal field then becomes a combinatorial problem.

2.2. Transport control

So far, we have considered the system (1.3) with a finite set of points $\mathcal{D}$ as initial data. A natural extension of this setting, particularly pertinent when the data points are sampled from an underlying distribution, is to consider as input a probability measure μ_0 on $\mathbb{R}^d$. The scenario where this distribution is a finite combination of Dirac deltas aligns with the study previously conducted in Section 2.1.

Specifically, we consider the space $\mathcal{P}^c_{ac}(\mathbb{R}^d)$, introduced in Section 1.2. We aim to transform a given $\mu_0 \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ into a fixed target probability measure μ_* through the push-forward map generated by a neural ODE, that is,

$$\boldsymbol{\Phi}_T(\cdot; W, A, \mathbf{b})_{\#}\mu_0 = \mu_*.$$

This question can be reformulated as the control problem of a transport equation. For each $t \in [0,T]$, we consider the family of measures $\mu(t) = \boldsymbol{\Phi}_{t\#}\mu_0$, where $\boldsymbol{\Phi}_t$ represents the flow at time t generated by (1.3). Given that the field

$$\sum_{i=1}^{p} \mathbf{w}_i(t)\sigma(\mathbf{a}_i(t)\cdot\mathbf{x} + b_i(t))$$

is Lipschitz continuous with respect to $\mathbf{x}$, if $\mu_0 \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ then the curve of measures $\{\mu(t)\}_{t\in[0,T]}$ is contained in $\mathcal{P}^c_{ac}(\mathbb{R}^d)$. For each $t > 0$, $\mu(t)$ is determined by a density function $\rho(t)$ that satisfies the neural transport equation

$$\begin{cases} \partial_t \rho + \operatorname{div}_{\mathbf{x}}\left(\rho \sum_{i=1}^{p} \mathbf{w}_i\, \sigma(\mathbf{a}_i\cdot\mathbf{x} + b_i)\right) = 0 \\ \rho(0) = \rho_0. \end{cases} \tag{2.6}$$

Here, we have assumed that μ_0 has density ρ_0, and

$$\{(\mathbf{w}_i, \mathbf{a}_i, b_i)\}_{i=1}^{p} \subset L^\infty\left((0,T); \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}\right)$$

serve again as control functions. The projected characteristics of (2.6) solve the neural ODE (1.3) in $(0,T)\times\mathbb{R}^d$. If the controls are step functions, and since the ReLU function is Lipschitz, the continuity Eq. (2.6) is well-posed and the total mass is conserved. Therefore, we aim to find some controls such that the corresponding solution of (2.6) with initial condition ρ_0 satisfies

$$\rho(T) = \rho_*.$$

This task, however, can be very hard to achieve, so we consider a relaxation of the problem to approximate control of (2.6). For this purpose, first we must choose a function to quantify the difference between any two measures.

Definition 8. For any $q \geq 1$, the *Wasserstein-q distance* between $\mu, \nu \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ is defined as

$$W_q(\mu,\nu) := \left(\min_{\gamma\in\Pi(\mu,\nu)} \int_{\mathbb{R}^d\times\mathbb{R}^d} |\mathbf{x}-\mathbf{y}|^q d\gamma(x,y)\right)^{1/q},$$

where $\Pi(\mu,\nu)$ denotes the set of measures γ on $\mathbb{R}^d\times\mathbb{R}^d$ that couple μ and ν in the sense that $\gamma(\cdot\times\mathbb{R}^d) = \mu(\cdot)$ and $\gamma(\mathbb{R}^d\times\cdot) = \nu(\cdot)$. Note that $\mu \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ has finite q-th momentum for every $q \geq 1$, hence the Wasserstein$-q$ distance is well-defined in this space. Moreover, recalling Monge formulation of optimal transport, if μ and ν belong to $\mathcal{P}^c_{ac}(\mathbb{R}^d)$ then

$$W_q(\mu,\nu) = \left(\min_{T} \int_{\mathbb{R}^d} |\mathbf{x} - T(\mathbf{x})|^q d\mu \,:\, T_{\#}\mu = \nu\right)^{1/q}, \tag{2.7}$$

where $T : \mathbb{R}^d \to \mathbb{R}^d$ measurable, see Villani (2008).

Problem-Definition. Let μ_0 and μ_* be two compactly supported, absolutely continuous probability measures with respective densities ρ_0 and ρ_*. For any fixed time horizon $T > 0$ and $\varepsilon > 0$, find controls

$$\left\{(\mathbf{w}_i, \mathbf{a}_i, b_i)\right\}_{i=1}^{p} \subset L^\infty\left((0,T); \mathbb{R}^d\times\mathbb{R}^d\times\mathbb{R}\right),$$

for some $p \geq 1$, such that the solution of (2.6) in time T *approximately interpolates* the initial condition ρ_0 to the target density ρ_*. This is achieved when the W_q-error of the corresponding measures, for some $q \geq 1$, satisfies:

$$W_q(\mu(T), \mu_*) < \varepsilon.$$

The following theorem offers a partial solution to this problem. It assumes that $1 \leq q < \frac{d}{d-1}$ and targets the uniform measure in $[0,1]^d$. While this bears resemblance to achieving null controllability, the nonlinear nature of the problem prevents from directly extending this result to arbitrary targets.

Theorem 8. *Let $d \geq 1$, $\mu_0 \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ with density ρ_0, μ_* the uniform measure in $[0,1]^d$, and $T > 0$ be fixed. For any $\varepsilon > 0$, $q \in [1, \frac{d}{d-1})$ and $p \geq d$, there exists a piecewise constant control*

$$(W, A, \mathbf{b}) \in L^\infty\left((0,T); \mathbb{R}^{d\times p}\times\mathbb{R}^{p\times d}\times\mathbb{R}^p\right)$$

such that the measure $\mu(t) \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ whose density $\rho(t)$ solves (1.4) taking ρ_0 as initial condition, satisfies

$$W_q(\mu(T), \mu_*) < \varepsilon.$$

Furthermore, the number of discontinuities of $(W, A, \mathbf{b})$ is

$$L = \lceil 2d/p\rceil + \max\{\lceil n/p_1\rceil, \dots, \lceil n^d/p_d\rceil\} - 1,$$

for any $p_1, \dots, p_d \geq 1$ such that $p_1 + \cdots + p_d = p$, and

$$n := \left(\frac{3d^{1/2+1/q}}{\varepsilon}\right)^{\frac{1}{1+d/q-d}}.$$

For a given $\varepsilon > 0$, the behavior of L resembles that described in Theorem 1, as it decreases with an increase in p, reaching $L = 1$ when p is large enough. Our proof is similar to a strategy from Duprez, Morancey, and Rossi (2017), and based on the specific movements that the neural ODE (1.3) allows. We compress the support of μ_0 to $[0,1]^d$ and divide it into hyperrectangles, each with a mass of $O(\varepsilon^d)$. These subsets are then transformed to match a similar partition of $[0,1]^d$ corresponding to the uniform measure μ_*.

Remark 9. If $\varepsilon > 0$ is sufficiently small, and we choose $p_1 = \cdots = p_{d-1} = 1$, $p_d = p - d + 1$, it follows that

$$L = \lceil 2d/p \rceil + \left\lfloor \frac{1}{p-d+1} \left(\frac{3^{1+d/q}\sqrt{d}}{\varepsilon} \right)^{\frac{d}{1+d/q-d}} \right\rfloor - 1.$$

For $q = 1$, this expression simplifies to:

$$L = \lceil 2d/p \rceil + \left\lfloor \frac{1}{p-d+1} \left(\frac{3^{1+d}\sqrt{d}}{\varepsilon} \right)^{d} \right\rfloor - 1.$$

3. Discussion

3.1. Conclusions

We have established several results on the capacity of neural ODEs for interpolation and its relationship with the chosen architecture, particularly concerning depth $L + 1$ and width p. More precisely, we have provided explicit dependencies between these two parameters that are sufficient to (exactly or approximately) interpolate either two sets of N different points in $\mathbb{R}^d$ or any compactly supported, absolutely continuous probability measure in $\mathbb{R}^d$ with the uniform measure in $[0,1]^d$. Our work reveals that p and L can play similar roles in the algorithms, thereby exhibiting a trade-off in the network's structure.

Specifically, Theorem 1 proves that a neural ODE with p neurons can interpolate any dataset of N pairs of points using a piecewise constant control with $L = 2\lceil N/p \rceil - 1$ discontinuities. Although increasing p reduces the number of discontinuities, we find a limiting case of a 2-hidden layer neural ODE when $p \geq N$. Explicit controls for interpolation with a shallow neural ODE ($L = 0$) are obtained in Corollary 3 when $d > N$; or in Corollary 5, with $p = N$ under Assumption 1. More generally, Theorem 7 provides an error decay rate with respect to the number of parameters for shallow neural ODEs. Finally, Theorem 8 explores the Wasserstein-q approximate control of the neural transport equation to a uniform distribution on $[0,1]^d$ using piecewise constant controls. As in Theorem 1, the number of discontinuities diminishes as p increases.

3.2. Open questions

Some new objectives can be derived from our work:

1. Approaching the autonomous regime. As we have discussed, both $d > N$ and Assumption 1 are only special cases where we can find controls to interpolate in the autonomous regime of shallow neural ODE. The question of finding such a construction for any $d \geq 1$, or at least under a less restrictive hypothesis on the dataset than Assumption 1, is still open. A first step could involve assuming the relaxed condition that the projections of points onto a line with direction $\mathbf{a} \in \mathbb{S}^{d-1}$ are ordered as $\mathbf{a} \cdot \mathbf{x}_{\tau(1)} < \cdots < \mathbf{a} \cdot \mathbf{x}_{\tau(N)}$ and $\mathbf{a} \cdot \mathbf{y}_{\tau(1)} < \cdots < \mathbf{a} \cdot \mathbf{y}_{\tau(N)}$, for a certain permutation τ of N elements. The strategy to be adopted is clear. It entails a combination of the one-dimensional control delineated in Lemma 13 with the transversal control required to prove Corollary 5, followed by a case-by-case analysis.

2. Universal approximation. In Ruiz-Balet and Zuazua (2023), the authors demonstrate that a narrow neural ODE can approximate any simple function with compact support. The proof hinges on simultaneous control and on the compressive nature of neural ODEs. However, minimizing the number of time discontinuities required for this control, particularly by increasing p, is a non-trivial task that may require an entirely different approach.

3. Neural transport equation. The problem of controlling the autonomous neural transport Eq. (2.6) is still open. The general case, using arbitrary autonomous fields, was studied in Nitti and Fernández-Real (2024), but an approximation argument similar to Theorem 7 is not possible. One approach could be approximating both the initial and target measures with atomic measures of the form $\rho_N = \frac{1}{N}\sum_{n=1}^{N} \beta_n \delta_{\alpha_n}$, with $\beta_n > 0$ and $\alpha_n \in \mathbb{R}^d$, and then interpolating those Dirac deltas by controlling the characteristic curves. However, a potential issue arises as $N \to \infty$: the distance between $\rho(T)$ and $\rho^N(T)$ (the solutions to (2.6) with respective initial conditions ρ_0 and ρ_0^N) may diverge significantly. This error can be quantified using the Grönwall inequality, which suggests that the Lipschitz constant could increase unboundedly with N, especially as the number of controlled points grows.

4. Minimizing the number of time jumps. Another interesting question is how to frame the reduction of discontinuities as an optimal control problem. For instance, one could penalize the frequency of time jumps by targeting the total variation seminorm. However, this seminorm lacks regularity, and moreover the class of piecewise constant functions is not a closed set of admissible controls.

5. Switching dimensions. In our simplified ResNet (1.1), the dimension remains constant across layers. However, strategically varying the hidden dimension by defining $p = p(t)$ could offer advantages, either by reducing complexity through dimension shrinkage or by creating space through dimension increase. Exploring effective methods to implement these transitions, whether by employing projections or by applying nonlinear transformations to the data, constitutes a compelling area for research.

6. Optimal activation function. As discussed in Remark 3, explicit control algorithms for more general activation functions that do not satisfy (2.4) have yet to be developed. Despite the potential gain in smoothness, the control cost generally increases due to an added difficulty: the absence of regions fixed by the flow causes all points to be continuously dispersed. Developing new constructive methods similar to those in this work would allow for estimating the complexity of controls in different dynamics and addressing the question of which activation function offers the best performance.

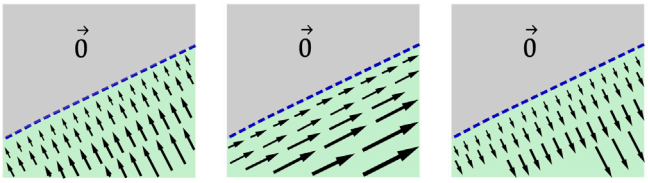


Fig. 4. Left to right: Compression, parallel motion, expansion. The blue dashed line represents the hyperplane $\mathbf{a} \cdot \mathbf{x} + b = 0$.

4. Proofs

4.1. Basic dynamics

We describe the simplest dynamics that we can generate via (1.6) by conveniently choosing $(\mathbf{w}, \mathbf{a}, b)$:

1. For each $t > 0$, the term $\mathbf{a}(t) \cdot \mathbf{x} + b(t)$ identifies a hyperplane in $\mathbb{R}^d$. For instance, taking $\mathbf{a} = \mathbf{e}_k$ and $b = -c$, we fix the hyperplane h of equation $x^{(k)} - c = 0$.

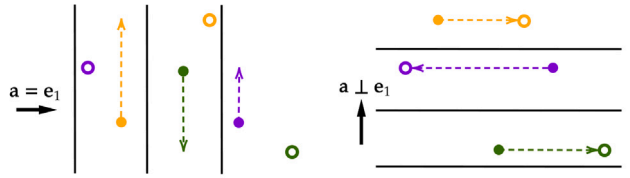
2. Application of σ and product with the vector $\mathbf{w}(t)$ yield the field $\mathbf{w}(t)\max\{x^{(k)} - c, 0\}$, which exhibits distinct dynamics in two complementary half-spaces:

$$H^+ \equiv \{x^{(k)} - c > 0\} \quad \text{and} \quad H^- \equiv \{x^{(k)} - c \leq 0\}$$

In H^+ the field equals $\mathbf{w}(t)(x^{(k)} - c)$, whereas in H^- the field is null, meaning H^- remains fixed under the flow.

3. The choice of $\mathbf{w}(t)$ specifies the orientation and magnitude of the field. For example, $\mathbf{w}(t) = \pm\mathbf{e}_k$ yields $\dot{x}^{(k)}(t) = \pm\max\{(x^{(k)} - c), 0\}$, so the points in H^+ can either be attracted to or repelled from h (i.e., along the kth coordinate). Conversely, $\mathbf{w}(t) = \mathbf{e}_i$ with $i \neq k$ results in $\dot{x}^{(i)}(t) = \max\{(x^{(k)} - c), 0\}$. In this case, we generate in H^+ a shear flow with direction $x^{(i)}$ that is parallel to h.

The three basic operations of compression, expansion and movement in parallel with the hyperplane (represented in Fig. 4) constitute our toolbox for many subsequent proofs.



(A) Step 1. Fix $x^{(1)}$ and simultaneously control $x^{(2)}, \dots, x^{(d)}$.

(B) Step 2. Fix $x^{(2)}, \dots, x^{(d)}$ and control $x^{(1)}$.

Fig. 5. Representation of the control algorithm followed in Theorem 1.

4.2. Proof of Theorem 1

We will employ the following lemma, whose proof we postpone to the end of this subsection:

Lemma 9. *Let $N \geq 1$, $d \geq 2$ and consider the dataset $\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N$ as defined in (2.1). There exists a change of coordinates in $\mathbb{R}^d$ such that*

$$x_n^{(1)} \neq x_m^{(1)} \qquad \text{and} \qquad y_n^{(2)} \neq y_m^{(2)}, \qquad \text{if } n \neq m. \tag{4.1}$$

Under the separability condition (4.1), we achieve the exact control by building on the methods developed in Ruiz-Balet and Zuazua (2023). Let $p \geq 1$ be fixed. We divide the proof in two steps, illustrated in Figs. 5(a) and 5(b).

Step 1: Control of $d-1$ coordinates. By (4.1), we can relabel the data $\{\mathbf{x}_n\}_{n=1}^N$ to impose the ordering

$$x_1^{(1)} < \cdots < x_N^{(1)}.$$

We define a partition of $\{\mathbf{x}_n\}_{n=1}^N$ in $\lceil N/p \rceil$ subsets by increasing order of the $x^{(1)}$-coordinate. The jth subset is

$$X_j := \{\mathbf{x}_{(j-1)\cdot p+1}, \dots, \mathbf{x}_{j\cdot p}\}, \quad \text{for } j = 1, \dots, \lceil N/p \rceil - 1,$$

and $X_{\lceil N/p \rceil}$ contains the remaining $N - p\lfloor N/p \rfloor$ points. We describe the control of the first subset X_1. We take controls $\mathbf{a}_i = \mathbf{e}_1$ and $b_i \in \mathbb{R}$, for $i = 1, \dots, p$, satisfying

$$-b_1 < x_1^{(1)} < -b_2 < \cdots < x_{p-1}^{(1)} < -b_p < x_p^{(1)}.$$

These controls define a family of parallel hyperplanes, given by $\mathbf{a}_i \cdot \mathbf{x} + b_i = x^{(1)} + b_i = 0$, which separate the points of X. In this way, $i-1$ terms of the sum in (1.3) cancel inside the strip $-b_i < x^{(1)} < -b_{i+1}$ for each $i = 1, \dots, p$, so (1.3) simplifies to

$$\dot{\mathbf{x}} = \sum_{l=1}^{i} \left\{ \mathbf{w}_l x^{(1)} + \mathbf{w}_l b_l \right\}. \tag{4.2}$$

We consider velocities of the form $\mathbf{w}_i = (0, w_i^{(2)}, \dots, w_i^{(d)})$, where the components $w_i^{(k)} \in \mathbb{R}$ have to be defined in order to achieve the exact control in time $T = 1$. The first point, $\mathbf{x}_1$, is subject only to one velocity, $\mathbf{w}_1$, so

$$x_1^{(k)}(t) = w_1^{(k)}(x_1^{(1)} + b_1)t + x_1^{(k)},$$

while $x_1^{(1)}$ remains fixed. Therefore, it is enough to take

$$w_1^{(k)} = \frac{y_1^{(k)} - x_1^{(k)}}{x_1^{(1)} + b_1}.$$

Similarly, for $i = 2, \dots, p$, having fixed $\mathbf{w}_1, \dots, \mathbf{w}_{i-1}$ it is enough to take

$$w_i^{(k)} = \frac{y_i^{(k)} - x_i^{(k)} - \sum_{l=1}^{i-1} w_l^{(k)}(x_i^{(1)} + b_l)}{x_i^{(1)} + b_i},$$

for $k = 1, \dots, d$. The described procedure can be simultaneously done for each X_j, with $j = 2, \dots \lceil N/p \rceil$, taking into account that the fields used to control $X_1, \dots, X_{j-1}$ (all of them orthogonal to $\mathbf{e}_1$) will be added as new terms in (4.2). In the end, we will have, for every $n = 1, \dots, N$:

$$\Phi_1(\mathbf{x}_n; W, A, \mathbf{b})^{(k)} = y_n^{(k)}, \qquad \text{for } k = 2, \dots, d.$$

The total number of iterations employed in this step is $\lceil N/p \rceil$, which corresponds to $\lceil N/p \rceil - 1$ switches.

Step 2: Control of the remaining coordinate. In a slight abuse of notation, we redefine $\mathbf{x}_n := \Phi_1(\mathbf{x}_n)$, where Φ_1 is the flow resulting from step 1. Once again, we can relabel the data, now assuming

$$x_1^{(2)} < \cdots < x_N^{(2)}.$$

Following the increasing order of the $x^{(2)}$-coordinate, we define $X_1, \dots, X_{\lceil N/p \rceil - 1}$, each being a subset of $\{\mathbf{x}_n\}_{n=1}^N$ with p points, and $X_{\lceil N/p \rceil}$, which contains the remaining $N - p\lfloor N/p \rfloor$ points.

We follow an analogous methodology to step 1. For each j, we define controls $\mathbf{a}_i = \mathbf{e}_2$ and b_i (for $i = 1, \dots, p$) that separate the points of X_j using p parallel hyperplanes, each described by the equation $x^{(2)} = b_i$. Now, we consider velocities of the form $\mathbf{w}_i = w_i \mathbf{e}_1$, where the values w_i are determined, as in step 1, to ensure

$$\Phi_1(\mathbf{x}_n; W, A, \mathbf{b})^{(1)} = y_n^{(1)}, \qquad \text{for } n = 1, \dots, N.$$

The number of switches employed in step 2 is $\lceil N/p \rceil - 1$. Then, by adding one more to transition between steps, the whole control requires $L = 2\lceil N/p \rceil - 1$ switches, hence proving Theorem 1.

Proof of Lemma 9. The finite unions of linear hyperplanes

$$\bigcup_{1 \leq n < m \leq N} (\text{span}(\mathbf{x}_n - \mathbf{x}_m))^\perp \subset \mathbb{R}^d$$

and

$$\bigcup_{1 \leq n < m \leq N} (\text{span}(\mathbf{y}_n - \mathbf{y}_m))^\perp \subset \mathbb{R}^d$$

have non-empty complements in $\mathbb{R}^d$. Therefore, one can always choose $\mathbf{u}_1, \mathbf{u}_2 \in \mathbb{S}^{d-1}$ such that for all $n \neq m$

$$\mathbf{u}_1 \cdot (\mathbf{x}_n - \mathbf{x}_m) \neq 0 \qquad \text{and} \qquad \mathbf{u}_2 \cdot (\mathbf{y}_n - \mathbf{y}_m) \neq 0.$$

If we complete to an orthonormal basis $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_d\}$ of $\mathbb{R}^d$, condition (4.1) is satisfied when expressing $\mathcal{D}$ in the new basis, as $x^{(k)} = \mathbf{x} \cdot \mathbf{u}_k$ for every $\mathbf{x} \in \mathbb{R}^d$. □

4.3. Proof of Corollary 3

We aim to eliminate the initial step in the algorithm defined in the proof of Theorem 1. To achieve this, we seek a new vector basis in $\mathbb{R}^d$ where the input-target pairs inherently share the first coordinate. When $d \leq N$, this condition usually cannot be met. However, when $d > N$, there exists an orthonormal vector basis $\mathcal{B} \subset \mathbb{R}^d$ such that for all $n = 1, \dots, N$, the first coordinates of the N pairs with respect to $\mathcal{B}$ satisfy $x_n^{(1)} = y_n^{(1)}$.

To construct such a vector system, without loss of generality we can assume that $d = N + 1$. Let $(\mathbf{x}, \mathbf{y}) \in \mathbb{R}^d \times \mathbb{R}^d$ with $\mathbf{x} \neq \mathbf{y}$. We seek a vector $\mathbf{u} \in \mathbb{S}^{d-1}$ such that $\mathbf{u} \cdot \mathbf{x} = \mathbf{u} \cdot \mathbf{y}$. This condition is equivalent to $\mathbf{u} \cdot (\mathbf{x} - \mathbf{y}) = 0$, which is satisfied by any unit vector $\mathbf{u}$ contained in the linear hyperplane orthogonal to $\mathbf{x} - \mathbf{y}$.

For $d-1$ input-target pairs of points, we consider the corresponding hyperplanes $\{H_n\}_{n=1}^{d-1}$. Note that some of these hyperplanes can be repeated. So, the intersection $\bigcap_{i=1}^{d-1} H_i$ yields a linear subspace of dimension at most $d-1$. We choose any unit vector $\mathbf{e}_1'$ contained in that subspace. Any completion to an orthonormal basis $\mathcal{B} = \{\mathbf{e}_1', \dots, \mathbf{e}_d\} \subset \mathbb{R}^d$ will satisfy the desired condition for $d-1$ points. The procedure is illustrated in Fig. 6.

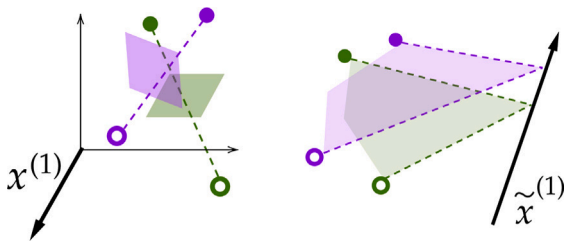


Fig. 6. Construction of a new axis $x^{(1)} \mapsto \tilde{x}^{(1)}$ in $\mathbb{R}^3$ for $N = 2$, ensuring that the first coordinates of each pair $(\mathbf{x}_1, \mathbf{y}_1)$ and $(\mathbf{x}_2, \mathbf{y}_2)$ are matched. Theorem 1 is then applied.

4.4. Proof of Corollary 5

Proof of Proposition 4. We carry out a similar analysis to the one in Alvarez-Lopez et al. (2023), where the probability of requiring k hyperplanes to separate two sets of N points for binary classification was estimated. We introduce the random variable $Z^p_{d,2N}(\mathcal{D})$, which assigns to each possible dataset

$$\mathcal{D} = \{(\mathbf{x}_n, \mathbf{y}_n)\}_{n=1}^N \subset \mathrm{supp}(\mu) \times \mathrm{supp}(\mu)$$

the minimum number of parallel hyperplanes needed to separate in $\mathbb{R}^d$ every pair $(\mathbf{x}_n, \mathbf{y}_n)$ from the others. Note that $\min_{\mathcal{D}} Z^p_{d,2N}(\mathcal{D}) = N - 1$. We estimate the probability $P_{d,N} := P(Z^p_{d,2N}(\mathcal{D}) = N-1)$ for any d and N.

First, we consider the one-dimensional case. Since all the points are sampled from the same distribution, every possible configuration of the $2N$ points in the real line will have the same probability, i.e., their distribution is uniform on the finite space of all possible orderings. Therefore, we can compute:

$$P_{1,N} = \frac{\text{favorable configurations}}{\text{total configurations}} = \frac{N!\,2^N}{(2N)!}.$$

Now, we apply Stirling's formula to approximate, for sufficiently large N:

$$P_{1,N} \approx \frac{N^N\sqrt{2\pi N}(2/e)^N}{(2N)^{2N}\sqrt{4\pi N}/e^{2N}} = \frac{1}{\sqrt{2}}\left(\frac{e}{2N}\right)^N.$$

Let $Z^{p,c}_{d,2N}$ be similarly defined to $Z^p_{d,2N}$ but restricting the hyperplanes to be orthogonal to one of the d canonical axes. Then, for any $d \geq 1$, we can bound:

$$P(Z^{p,c}_{d,2N} = N-1) \leq P(Z^p_{d,2N} = N-1). \tag{4.3}$$

By hypothesis, the d random variables defined as $Z^p_{1,2N}$ over the projection of $\mathcal{D}$ on each canonical axis are i.i.d. to $Z^p_{1,2N}$, so we can write:

$$P(Z^{p,c}_{d,2N} > N-1) = \left[1 - P(Z_{1,2N} = N-1)\right]^d = \frac{N!\,2^N}{(2N)!} \approx \left[1 - \frac{1}{\sqrt{2}}\left(\frac{e}{2N}\right)^N\right]^d,$$

if $N \gg 1$. By (4.3), the complementary provides the desired lower bound for $P_{d,N}$. □

Proof of Corollary 5. Let $\mathbf{a} \in \mathbb{S}^{d-1}$, $\{b_n\}_{n=1}^{N+1} \subset \mathbb{R}$ and τ be given by Assumption 1. With no loss of generality, we can assume that $\mathbf{a} = \mathbf{e}_1$ and τ is the identity permutation. The argument that we will use is similar to the one employed in the proof of Theorem 1, but now the motion must be longitudinal as well as transverse. It also hinges on the fact that, inside the nth strip

$$S_n := \{\mathbf{x} \in \mathbb{R}^d : -b_n < \mathbf{a}\cdot\mathbf{x} < -b_{n+1}\},$$

Eq. (1.5) simplifies to (4.2). The simultaneous control of the data points is achieved inductively, in increasing order of the first coordinates, by appropriately defining the field $\mathbf{w}_n$ associated with each hyperplane H_n. Both the base case and the inductive step are established by the following two lemmas, which will be proven later.

Lemma 10. *Consider two points $\mathbf{x}_1, \mathbf{y}_1 \in \mathbb{R}^d$ with $\mathbf{x}_1 \neq \mathbf{y}_1$. For any $T > 0$ and $b \in \mathbb{R}$ satisfying $x_1^{(1)} + b > 0$ and $y_1^{(1)} + b > 0$, there exists a unique $\mathbf{w} \in \mathbb{R}^d$ such that the solution of*

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{w}\sigma(x^{(1)} + b), \\ \mathbf{x}(0) &= \mathbf{x}_1 \in \mathbb{R}^d \end{cases} \tag{4.4}$$

reaches $\mathbf{x}(T) = \mathbf{y}_1$.

Having controlled $\mathbf{x}_1, \dots, \mathbf{x}_{n-1}$ to $\mathbf{y}_1, \dots, \mathbf{y}_{n-1}$ in a time horizon $T > 0$, and using parameters $\{\mathbf{w}_i\}_{i=1}^{n-1} \subset \mathbb{R}^d$ and $\{b_i\}_{i=1}^{n-1} \subset \mathbb{R}$ such that $b_{n-1} < \cdots < b_1$ and

$$x_i^{(1)} + b_i > 0 \quad \text{and} \quad y_i^{(1)} + b_i > 0 \quad \text{for } i = 1, \dots, N,$$

steering $\mathbf{x}_n$ involves overcoming an autonomous drift field

$$\mathbf{d}(\mathbf{x}) := \sum_{i=1}^{n-1} \mathbf{w}_i \sigma(\mathbf{a}\cdot\mathbf{x} + b_i) = \sum_{i=1}^{n-1} \mathbf{w}_i \sigma(x^{(1)} + b_i). \tag{4.5}$$

The drift field $\mathbf{d}$ becomes more intense as the first coordinate increases, owing to the characteristics of the ReLU function. However, the following lemma shows that the control is possible:

Lemma 11. *Consider $\mathbf{x}_n, \mathbf{y}_n \in \mathbb{R}^d$ with $\mathbf{x}_n \neq \mathbf{y}_n$. With the above notation, for any $T > 0$ there exists a unique $\mathbf{w}_n \in \mathbb{R}^d$ and some $b_n \in \mathbb{R}$ satisfying*

$$b_n < b_{n-1}, \quad x_n^{(1)} + b_n > 0 \quad \text{and} \quad y_n^{(1)} + b_n > 0$$

such that the solution of the Cauchy problem

$$\begin{cases} \dot{\mathbf{x}} &= \mathbf{d}(\mathbf{x}) + \mathbf{w}_n\sigma(x^{(1)} + b_n), \\ \mathbf{x}(0) &= \mathbf{x}_n, \end{cases} \tag{4.6}$$

where $\mathbf{d}$ is given by (4.5), reaches $\mathbf{x}(T) = \mathbf{y}_n$.

With Lemmas 10 and 11, the inductive argument is almost complete. It is left to show that the trajectory of each initial datum $\mathbf{x}_n$ will remain in its corresponding strip S_n for all $t \in (0, T)$.

On one hand, Lemma 11 guarantees that the trajectory $\mathbf{x}(t)$ originating from $\mathbf{x}_n$ will reach the endpoint $\mathbf{y}_n$. On the other hand, taking into account that the field is autonomous and invariant along each hyperplane $x^{(1)} = \text{const}$, then $\dot{x}^{(1)}(t)$ cannot change sign at any time, that is to say, the $x^{(1)}(t)$ does not change its direction. Consequently, the entire trajectory will be contained within the strip bounded by $x = x_n^{(1)}$ and $x = y_n^{(1)}$, which in turn is contained in S_n. □

Proof of Lemma 10. In the half-space $\{x^{(1)} + b > 0\}$, Eq. (4.4) is written as

$$\dot{\mathbf{x}} = \mathbf{w}\sigma(x^{(1)} + b) = \mathbf{w}x^{(1)} + \mathbf{w}\,b. \tag{4.7}$$

We can assume that $b = 0$, so the solution of (4.7) is

$$\mathbf{x}(t) = \frac{x_1^{(1)}}{w^{(1)}}\mathbf{w}\left(e^{w^{(1)}t} - 1\right) + \mathbf{x}_1,$$

which can be driven to $\mathbf{x}(T) = \mathbf{y}_1$ by taking

$$w^{(1)} = \frac{1}{T}\ln\left(\frac{y_1^{(1)}}{x_1^{(1)}}\right) \quad \text{and} \quad w^{(k)} = \frac{y_1^{(k)} - x_1^{(k)}}{y_1^{(1)} - x_1^{(1)}}w^{(1)},$$

or $w^{(k)} = y_1^{(k)} - x_1^{(k)}$ if $x_1^{(1)} = y_1^{(1)}$, for $k = 2, \dots, d$. Moreover, $\mathbf{x}(t)$ stays in the half-space $x^{(1)} > 0$ for $t \in [0, T]$ because $x^{(1)}(t)$ is monotone in that interval. □

Proof of Lemma 11. First, take any $b_n \in \mathbb{R}$ satisfying

$$b_n < b_{n-1}, \quad x_n^{(1)} + b_n > 0 \quad \text{and} \quad y_n^{(1)} + b_n > 0.$$

For simplicity, we rewrite (4.4) as

$$\dot{\mathbf{x}} = \left(\mathbf{s}_{n-1} + \mathbf{w}_n\right) x^{(1)} + \mathbf{w}_n b_n + \mathbf{c}_{n-1},$$

where $\mathbf{s}_{n-1} = \sum_{i=1}^{n-1} \mathbf{w}_i$, $\mathbf{c}_{n-1} = \sum_{i=1}^{n-1} \mathbf{w}_i b_i$. If we restrict to the first coordinate, we have:

$$\begin{cases} \dot{x}^{(1)} &= \left(s_{n-1}^{(1)} + w_n^{(1)}\right) x^{(1)} + w_n^{(1)} b_n + c_{n-1}^{(1)}, \\ x^{(1)}(0) &= x_n^{(1)}, \end{cases}$$

which has solution

$$x^{(1)}(t) = \frac{w_n^{(1)} b_n + c_{n-1}^{(1)}}{s_{n-1}^{(1)} + w_n^{(1)}} \left[e^{\left(s_{n-1}^{(1)} + w_n^{(1)}\right)t} - 1\right] + x_n^{(1)} e^{\left(s_{n-1}^{(1)} + w_n^{(1)}\right)t}.$$

First, we want to see if there exists $\hat{w}_n^{(1)} \in \mathbb{R}$ such that $x^{(1)}(T) = y_n^{(1)}$, or, equivalently, if the function

$$f(z) = \frac{z\, b_n + c_{n-1}^{(1)}}{s_{n-1}^{(1)} + z} \left[e^{\left(s_{n-1}^{(1)} + z\right)T} - 1\right] + x_n^{(1)} e^{\left(s_{n-1}^{(1)} + z\right)T} - y_n^{(1)} \tag{4.8}$$

has a real root. For that task, we compute:

$$\lim_{z\to\infty} f(z) = \operatorname{sign}\left(b_n + x_n^{(1)}\right) \cdot \infty = +\infty,$$
$$\lim_{z\to-\infty} f(z) = -b_n - y_n^{(1)} < 0.$$

On the other hand,

$$\lim_{z\to -s_{n-1}^{(1)}{}^+} f(z) = \left(c_{n-1}^{(1)} - b_n s_{n-1}^{(1)}\right) T + x_n^{(1)} - y_n^{(1)} = \lim_{z\to -s_{n-1}^{(1)}{}^-} f(z),$$

so f is continuous. Hence, we can assure that there exists $\hat{w}_n^{(1)} \in \mathbb{R}$ such that $f\left(\hat{w}_n^{(1)}\right) = 0$. Now, we denote

$$s_n^{(1)} = \hat{w}_n^{(1)} + s_{n-1}^{(1)} \quad \text{and} \quad c_n^{(1)} = \hat{w}_n^{(1)} b_n + c_{n-1}^{(1)}.$$

For each component $j \in \{2, \dots, d\}$,

$$\begin{cases} \dot{x}^{(j)} &= c_n^{(1)} \frac{s_{n-1}^{(j)} + w_n^{(j)}}{s_n^{(1)}} \left[e^{s_n^{(1)} t} - 1\right] \\ &\quad + x_n^{(1)} \left(s_{n-1}^{(j)} + w_n^{(j)}\right) e^{s_n^{(1)} t} + w_n^{(j)} b_n + c_{n-1}^{(j)}, \\ x^{(j)}(0) &= x_n^{(j)}, \end{cases}$$

which has solution

$$x^{(j)}(t) = x_n^{(j)} + \left[w_n^{(j)} b_n + c_{n-1}^{(j)} - c_n^{(1)} \frac{s_{n-1}^{(j)} + w_n^{(j)}}{s_n^{(1)}}\right] t + \left[c_n^{(1)} \frac{s_{n-1}^{(j)} + w_n^{(j)}}{s_n^{(1)^2}} + x_n^{(1)} \frac{s_{n-1}^{(j)} + w_n^{(j)}}{s_n^{(1)}}\right] e^{s_n^{(1)} t}.$$

Now, we want to find a solution of $g(z) = 0$ for

$$g(z) = x_n^{(j)} + \left[z b_n + c_{n-1}^{(j)} - c_n^{(1)} \frac{s_{n-1}^{(j)} + z}{s_n^{(1)}}\right] T + \left[c_n^{(1)} \frac{s_{n-1}^{(j)} + z}{s_n^{(1)^2}} + x_n^{(1)} \frac{s_{n-1}^{(j)} + z}{s_n^{(1)}}\right] e^{s_n^{(1)} T} - y_n^{(j)}.$$

This is an affine function in z, so $g(z) = 0$ has a unique solution $\hat{w}_n^{(j)} \in \mathbb{R}$ if and only if the slope is non-zero. Suppose that the b_n we have chosen yields a zero slope, i.e.,

$$\left[\frac{c_n^{(1)}}{s_n^{(1)}} + x_n^{(1)}\right] e^{s_n^{(1)} T} + (b_n s_{n-1}^{(1)} - c_{n-1}^{(1)}) T = 0.$$

Recalling that $f(\hat{w}_n^{(1)}) = 0$, we can write this equation as

$$y_n^{(1)} + \frac{c_n^{(1)}}{s_n^{(1)}} + (b_n s_{n-1}^{(1)} - c_{n-1}^{(1)}) T = 0,$$

so

$$\frac{c_n^{(1)}}{s_n^{(1)}} = (c_{n-1}^{(1)} - b_n s_{n-1}^{(1)}) T - y_n^{(1)} \tag{4.9}$$

and we can solve for $\hat{w}_n$ in this equation as

$$\hat{w}_n^{(1)} = \frac{\left[(c_{n-1}^{(1)} - b_n s_{n-1}^{(1)}) T - y_n^{(1)}\right] s_{n-1}^{(1)} - c_{n-1}^{(1)}}{b_n - \left(c_{n-1}^{(1)} - b_n s_{n-1}^{(1)}\right) T - y_n^{(1)}}.$$

Meanwhile, substituting (4.9) in $f(\hat{w}_n^{(1)}) = 0$, we get:

$$\left[x_n^{(1)} - y_n^{(1)}\right] e^{s_n^{(1)} T} - T(c_{n-1}^{(1)} + b_n s_{n-1}^{(1)})[e^{s_n^{(1)} T} - 1] = 0,$$

which can also be solved for $\hat{w}_n^{(1)}$ as

$$\hat{w}_n^{(1)} = \frac{1}{T} \ln\left(\frac{(c_{n-1}^{(1)} - b_n s_{n-1}^{(1)}) T}{(c_{n-1}^{(1)} - b_n s_{n-1}^{(1)}) T - y_n^{(1)} + x_n^{(1)}}\right) - s_{n-1}^{(1)}.$$

Equalizing both expressions of $\hat{w}_n^{(1)}$, we obtain an equation with different analytic functions of b_n and not involving $\hat{w}_n^{(1)}$. Therefore, changing slightly the chosen value of b_n, these expressions become different, independently of the corresponding value of $\hat{w}_n^{(1)}$, so there is a perturbation of b_n that satisfies the statement of the lemma while $g(z) = 0$ has a unique solution. □

4.5. Proof of Theorem 7

First, we prove Proposition 6 with an inductive argument of topological nature.

Proof of Proposition 6. First, we aim to build a family of N disjoint C^∞ curves contained in $\operatorname{Int}(\Omega)$, each connecting the two points of a corresponding pair $(\mathbf{x}_n, \mathbf{y}_n) \in D$. When $d \geq 2$, any connected open set in $\mathbb{R}^d$ is path-connected. Therefore, we can take a continuous path C that connects any two given points $(\mathbf{x}, \mathbf{y})$ inside $U := \operatorname{Int}(\Omega) \setminus K$, where K represents any finite union of disjoint curves contained in $\operatorname{Int}(\Omega)$. Moreover, by a well-known approximation argument this path can be chosen to be C^∞.

Now, we have N disjoint C^∞ curves $\{C_n\}_{n=1}^N$ contained in $\operatorname{Int}(\Omega)$, each connecting a corresponding pair of points $(\mathbf{x}_n, \mathbf{y}_n)$. The tangent velocity field of each curve is also C^∞, so the assembled field $\mathbf{V}'$, defined in $\bigcup_{n=1}^N C_n$, is smooth too. Since its domain is compact, it is also Lipschitz-continuous. The required vector field $\mathbf{V}$ is explicitly provided by Kirzsbraun's Theorem (see Valentine, 1945), which ensures the existence of a Lipschitz map $\mathbf{V} : \mathbb{R}^d \to \mathbb{R}^d$ that extends $\mathbf{V}'$ sharing the same Lipschitz constant. □

The proof of Theorem 7 employs the following lemma from DeVore et al. (2021, Section 7.2.2):

Lemma 12 (*Approximation Rate for Lipschitz Functions*). *Let K be the unit ball in $\operatorname{Lip}(\Omega, \mathbb{R})$, where $\Omega = [-R, R]^d$. We have*

$$\frac{1}{[\kappa \log_2 \kappa]^{1/d}} C_{d,R} \leq E(K, \Sigma_\kappa)_{C(\Omega)} \leq C_{d,R} \frac{\log_2 \kappa}{\kappa^{1/d}},$$

where:

- $\Sigma_\kappa := \left\{S_\kappa : \mathbb{R}^d \to \mathbb{R}\right\}$ *is the space of shallow neural networks with $\kappa = (d+2)p$ parameters, p being the number of neurons in the hidden layer;*
- $E(K, \Sigma_\kappa)_{C(\Omega)} = \sup_{f\in K} \inf_{S\in\Sigma_\kappa} \|f - S\|_{C(\Omega)}$ *measures the capacity of Σ_κ to approximate any function in K.*

This result was in turn derived from Bach (2017, Proposition 6), where the upper bound is obtained for K_L, the ball of radius L in $\mathrm{Lip}(\Omega, \mathbb{R})$:

$$E(K_L, \Sigma_\kappa)_{C(\Omega)} \leq C_{d,R}\, L\, \frac{\log_2 \kappa}{\kappa^{1/d}}. \tag{4.10}$$

Proof of Theorem 7. Proposition 6 ensures that we can find a Lipschitz-continuous field $\mathbf{V} : \mathbb{R}^d \to \mathbb{R}^d$ such that the flow Ψ_T of the ODE

$$\dot{\mathbf{x}} = \mathbf{V}(\mathbf{x}) \tag{4.11}$$

interpolates the dataset. The classical UAT in Cybenko (1989) guarantees that we can uniformly approximate in Ω with precision $\varepsilon/\sqrt{d}$ each of its components $V^{(i)}$ using a corresponding shallow neural network $S^i_{\kappa_i} : \Omega \to \mathbb{R}$, with $\kappa_i \geq 1$ for $i = 1, \dots, d$. Moreover, Lemma 12 quantifies the dependence of the error with respect to the number of parameters of each $S^i_{\kappa_i}$, ensuring that

$$\kappa_1 = \kappa_2 = \cdots = \kappa_d = (d+2)p$$

parameters suffice to ensure (4.10) on each component. The assembled field

$$\mathbf{V}_{NN} = \left(S^1_{\kappa_1}, \dots, S^d_{\kappa_d}\right) : \mathbb{R}^d \to \mathbb{R}^d$$

is of the form $\mathbf{V}_{NN}(\mathbf{x}) = W\sigma(A\mathbf{x}+\mathbf{b})$ with complexity $\kappa = (d+2)pd$, and satisfies $\sup_{\mathbf{x}\in\Omega} |\mathbf{V}(\mathbf{x}) - \mathbf{V}_{NN}(\mathbf{x})| < \varepsilon$. Consider the neural ODE given by

$$\dot{\mathbf{x}} = \mathbf{V}_{NN}(\mathbf{x}). \tag{4.12}$$

Let $\mathbf{X}_V(t;\mathbf{x}_0)$ and $\mathbf{X}_{NN}(t;\mathbf{x}_0)$ be the respective trajectories in time $t > 0$ that a point $\mathbf{x}_0 \in \Omega$ will follow under the dynamics provided by (4.11) and (4.12). The deviation

$$\mathbf{z}(t) = |\mathbf{X}_V(t;\mathbf{x}_0) - \mathbf{X}_{NN}(t;\mathbf{x}_0)|$$

is bounded as

$$\begin{aligned}
\mathbf{z}(t) &\leq \int_0^t \left|\mathbf{V}(\mathbf{X}_V(s;\mathbf{x}_0)) - \mathbf{V}_{NN}(\mathbf{X}_{NN}(s;\mathbf{x}_0))\right| \mathrm{ds} \\
&\leq \int_0^t \Big\{\left|\mathbf{V}(\mathbf{X}_V(s;\mathbf{x}_0)) - \mathbf{V}(\mathbf{X}_{NN}(s;\mathbf{x}_0))\right| \\
&\quad + \left|\mathbf{V}(\mathbf{X}_{NN}(s;\mathbf{x}_0)) - \mathbf{V}_{NN}(\mathbf{X}_{NN}(s;\mathbf{x}_0))\right|\Big\} \mathrm{ds} \\
&\leq L_V \int_0^t \left|\mathbf{X}_V(s;\mathbf{x}_0) - \mathbf{X}_{NN}(s;\mathbf{x}_0)\right| \mathrm{ds} + \varepsilon\, t \\
&= L_V \int_0^t \mathbf{z}(s)\mathrm{ds} + \varepsilon\, t.
\end{aligned}$$

By Grönwall's inequality, it follows that

$$\mathbf{z}(t) \leq \varepsilon\ t\ \exp\left(L_V\ t\right).$$

On the other hand, in the second line, we could alternatively add and subtract $\mathbf{V}_{NN}(\mathbf{X}_V(s;\mathbf{x}_0))$. Then, if we denote by L_{NN} the Lipschitz constant of $\mathbf{V}_{NN}$, we have:

$$\begin{aligned}
\mathbf{z}(t) &\leq \int_0^t \Big\{\left|\mathbf{V}(\mathbf{X}_V(s;\mathbf{x}_0)) - \mathbf{V}_{NN}(\mathbf{X}_V(s;\mathbf{x}_0))\right| \\
&\quad + \left|\mathbf{V}_{NN}(\mathbf{X}_V(s;\mathbf{x}_0)) - \mathbf{V}_{NN}(\mathbf{X}_{NN}(s;\mathbf{x}_0))\right|\Big\} \mathrm{ds} \\
&\leq \varepsilon\, t + L_{NN} \int_0^t \left|\mathbf{X}_{NN}(s;\mathbf{x}_0) - \mathbf{X}_V(s;\mathbf{x}_0)\right| \mathrm{ds} \\
&= \varepsilon\, t + L_{NN} \int_0^t \mathbf{z}(s)\mathrm{ds}.
\end{aligned}$$

By Grönwall's inequality, it follows that

$$\mathbf{z}(t) \leq \varepsilon\ t\ \exp\left(L_{NN}\ t\right).$$

Taking $\mathbf{x}_0 = \mathbf{x}_n$ and $t = T$, the two bounds for $\mathbf{z}(t)$ give:

$$|\mathbf{y}_n - \boldsymbol{\Phi}_T(\mathbf{x}_n)| \leq \varepsilon\ T\ \exp\left(\min\{L_V, L_{NN}\}\, T\right).$$

Note that $L_{NN} \leq \|W\|\|A\|$ because σ is 1-Lipschitz, so the approximation rate (2.5) is obtained by direct application of (4.10). □

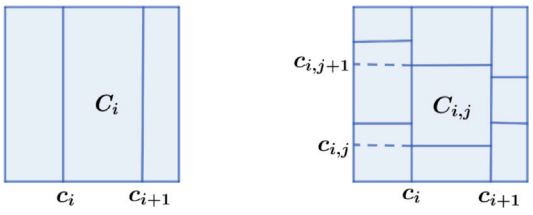


Fig. 7. Division of $[0,1]^2$ into rectangles, each containing a mass of $1/n^2$ following the distribution given by ρ_0.

4.6. Proof of Theorem 8

We seek to find $(W, A, \mathbf{b})$ such that the generated vector field moves, compresses and stretches the mass distributed following ρ_0, to drive it approximately to the target density, given by ρ_*. This is achieved in four steps, illustrated in Figs. 7 to 10 for the case $d = 2$.

1. Preparation. We compress $\mathrm{supp}(\mu_0)$ into $[0,1]^d$. To do this, we aim to find a control

$$(W, A, \mathbf{b}) \in L^\infty\left((0,T); \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times d} \times \mathbb{R}^p\right)$$

such that, in a time $T_1 > 0$, the flow $\boldsymbol{\Phi}_{T_1}$ of (1.3) satisfies

$$\boldsymbol{\Phi}_{T_1}(\mathrm{supp}(\mu_0)) \subset [0,1]^d$$

For $k = 1, \dots, d$, we fix the hyperplane $x^{(k)} = 0$ and a compressive velocity field by taking $(\mathbf{w}, \mathbf{a}, b) = (-\mathbf{e}_k, \mathbf{e}_k, 0)$. We choose $T_{1,k} > 0$ sufficiently large to ensure

$$\boldsymbol{\Phi}_{T_{1,k}}\left(\mathrm{supp}(\mu_0) \cap \{x^{(k)} \geq 0\}\right) \subset \{0 \leq x^{(k)} \leq 1\}.$$

We repeat the operation with the hyperplanes $x^{(k)} = 1$ for $k = 1, \dots, d$, taking $(\mathbf{w}, \mathbf{a}, b) = (\mathbf{e}_k, -\mathbf{e}_k, 1)$ and $T'_{1,k} > 0$ such that

$$\boldsymbol{\Phi}_{T'_{1,k}} \circ \boldsymbol{\Phi}_{T_{1,k}}\left(\mathrm{supp}(\mu_0) \cap \{x^{(k)} \leq 0\}\right) \subset \{0 \leq x^{(k)} \leq 1\}.$$

Both operations are possible in a finite time because μ_0 has compact support. In the end, we will have built piecewise constant controls $(\mathbf{w}, \mathbf{a}, b)$ that take $2d$ values, such that

$$\boldsymbol{\Phi}_{T_1}(\mathrm{supp}(\mu_0)) \subset [0,1]^d, \quad \text{for } T_1 := \sum_{k=1}^d \left(T_{1,k} + T'_{1,k}\right).$$

Using p neurons, we can simultaneously apply p controls, because when $(\mathbf{w}, \mathbf{a}, b) = (\pm\mathbf{e}_k, \pm\mathbf{e}_k, 1)$, then the characteristic curves of (2.6) fulfill $x^{(l)} = \mathrm{const}$ for every $l \neq k$. So, the total number of values taken by $(W, A, \mathbf{b})$ is $\lceil 2d/p \rceil$.

2. Partition. We aim to divide $[0,1]^d$ into a collection of n^d hyperrectangles, each containing a mass of $1/n^d$, as distributed by μ_0. The process can be visualized in Fig. 7 for $d = 2$. For simplicity, we redefine $\mu_0 := \boldsymbol{\Phi}_{T_1\#}\mu_0$ with density ρ_0, now satisfying $\mathrm{supp}(\mu_0) \subset [0,1]^d$. Let $n \geq 1$ and consider the function

$$t \longmapsto \int_{[0,t)\times[0,1]^{d-1}} d\mu_0 = \int_{[0,t)\times[0,1]^{d-1}} \rho_0.$$

This function is continuous, strictly increasing (by absolute continuity), equal to 0 at $t = 0$ and equal to 1 at $t = 1$. Therefore, we can choose $n+1$ numbers

$$c_0 = 0 < c_1 < \cdots < 1 = c_n$$

such that, for $i_1 = 0, \dots, n-1$,

$$\int_{[c_{i_1}, c_{i_1+1}]\times[0,1]^{d-1}} \rho_0 = \frac{1}{n}.$$

Similarly, for each $i_1 = 0, \dots, n-1$ we choose $n+1$ numbers

$$c_{i_1,0} = 0 < c_{i_1,1} < \cdots < 1 = c_{i_1,n}$$

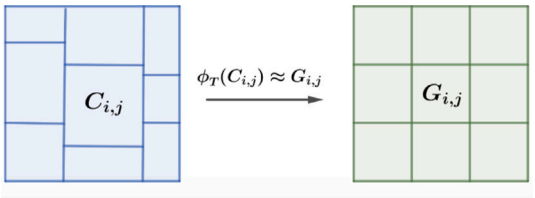


Fig. 8. Transformation of the rectangles C_{I_d} into the corresponding ones G_{I_d}.

such that, for $i_2 = 0, \dots, n-1$,

$$\int_{[c_{i_1}, c_{i_1+1}]\times[c_{i_1,i_2}, c_{i_1,i_2+1}]\times[0,1]^{d-2}} \rho_0 = \frac{1}{n^2}.$$

Repeating this operation recursively for each coordinate, we end up with n^d hyperrectangles

$$C^0_{i_1,\dots,i_d} := \left[c_{i_1}, c_{i_1+1}\right] \times \cdots \times \left[c_{i_1,\dots,i_d}, c_{i_1,\dots,i_d+1}\right] \subset \mathbb{R}^d,$$

with $i_k \in \{0, \dots, n-1\}$ for every $k = 1, \dots, d$, such that

$$\int_{C^0_{i_1,\dots,i_d}} \rho_0 = \frac{1}{n^d}.$$

The analogous partition for the uniform measure μ_* is

$$G_{i_1,\dots,i_d} := \left[\frac{i_1}{n}, \frac{i_1+1}{n}\right] \times \dots \times \left[\frac{i_d}{n}, \frac{i_d+1}{n}\right].$$

For the sake of readability, we will denote each multi-index by $I_k \equiv (i_1, \dots, i_k) \in \{0, \dots, n-1\}^k$, for $k = 1, \dots, d$, so for instance we write $C^0_{I_d} \equiv C^0_{i_1,\dots,i_d}$ and $G_{I_d} \equiv G_{i_1,\dots,i_d}$.

3. Control. We aim to define the controls that expand and compress the mass until the hyperrectangles $C^0_{I_d}$ approximate a corresponding collection of n^d hypercubes, each of them containing the same mass $1/n^d$, as distributed by μ_*. Ideally, we would build $(W, A, \mathbf{b})$ such that the flow of the ODE (1.3) satisfied

$$\Phi_T(C^0_{I_d}; W, A, \mathbf{b}) = G_{I_d} \tag{4.13}$$

for each $I_d \in \{0, \dots, n-1\}^d$. This would be done by transforming each hyperplane $\{x^{(k)} = c_{I_k}\}$ into a target hyperplane $\{x^{(k)} = i_k/n\}$.

However, this task is not possible in general, since any $\{x^{(k)} = i_k/n\}$ can be a target for multiple distinct hyperplanes $\{x^{(k)} = c_{I_k}\}$. We therefore relax the problem to δ-approximate control by considering δ-displacements of the target hyperplanes $\{x^{(k)} = i_k/n\}$. Now we aim to control each $C^0_{I_d}$ to a corresponding target $G^\delta_{I_d}$ that is δ-close to G_{I_d}, for a sufficiently small $\delta > 0$.

Let us build the new hyperrectangles $G^\delta_{I_d}$. The process is shown in Fig. 9. For each $k \in \{2, \dots, d\}$, $i_k \in \{0, \dots, n\}$ and $I_{k-1} \in \{0, \dots, n-1\}^{k-1}$, we define

$$g^\delta_{I_k} := i_k/n + \delta(c_{I_k} - \tilde{c}_{i_k}),$$

where

$$\tilde{c}_{i_k} := \min\{c_{I'_{k-1},i_k} : I'_{k-1} \in \{0, \dots, n-1\}^{k-1}\}.$$

Note that $g^\delta_{I_k} = 1$ whenever $i_k = n$. By construction,

$$g^\delta_{I_{k-1},i_k} < g^\delta_{I'_{k-1},i_k} \iff c_{I_{k-1},i_k} < c_{I'_{k-1},i_k}$$

and

$$g^\delta_{I_{k-1},i_k} = g^\delta_{I'_{k-1},i_k} \iff c_{I_{k-1},i_k} = c_{I'_{k-1},i_k}.$$

By recursion, we define a new partition of $[0,1]^d$ into a collection of rectangles $G^\delta_{I_d}$ with $I_d \in \{0, \dots, n-1\}^d$, where

$$G^\delta_{I_d} \subset G_{I_d} + \{0\} \times [-\delta, \delta]^{d-1}.$$

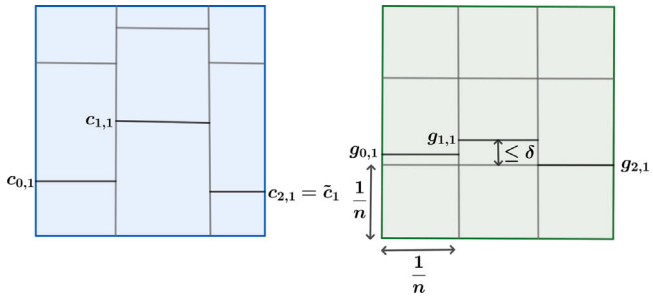


Fig. 9. Construction of the partition in rectangles $G^\delta_{I_d}$.

Moreover, this partition mimics the structure of the partition defined for μ_0, in the sense that there is the same number of distinct target hyperplanes as initial ones to be controlled. To sum up, if we take $\delta < 1/n$, we end up with:

$$\left\{C^0_{I_d} : I_d \in \{0, \dots, n-1\}^d\right\} \text{ s.t. } \int_{C^0_{I_d}} d\mu_0 = \frac{1}{n^d}, \tag{4.14}$$

$$\left\{G^\delta_{I_d} : I_d \in \{0, \dots, n-1\}^d\right\} \text{ s.t. } \int_{G^\delta_{I_d}} d\mu_* \leq \frac{3^d}{n^d},$$

$$\text{and } \operatorname{diam}(G^\delta_{I_d}) \leq \frac{3\sqrt{d}}{n}.$$

It is left to map $C^0_{I_d}$ to $G^\delta_{I_d}$ for each I_d. This is based on the following lemma, whose proof we postpone for readability.

Lemma 13. *Let $d \geq 2$, $\mu_0 \in \mathcal{P}^c_{ac}(\mathbb{R}^d)$ with density ρ_0, ρ_* the uniform density in $[0,1]^d$, and $T > 0$ be fixed. Let $n \geq 1$ and consider a family of hyperrectangles such as (4.14). For any $p_1, \dots, p_d \geq 1$, there exists a piecewise constant control*

$$(W, A, \mathbf{b}) \in L^\infty((0,T); \mathbb{R}^{p\times d} \times \mathbb{R}^{d\times p} \times \mathbb{R}^p)$$

with $p := p_1 + \cdots + p_d$ such that the flow Φ_T generated by (1.3) satisfies, for each $I_d \in \{0, \dots, n-1\}^d$:

$$\Phi_T(C^0_{I_d}; W, A, \mathbf{b}) = G^\delta_{I_d}.$$

Furthermore, the number of discontinuities of $(W, A, \mathbf{b})$ is

$$L = \max\{\lceil n/p_1 \rceil, \dots, \lceil n^d/p_d \rceil\} - 1.$$

4. Estimates. We compute the W_q-distance between both measures to verify the approximate control. Let Φ_T be the flow given by Lemma 13, satisfying

$$\Phi_T(C_{I_d}) = G^\delta_{I_d}, \quad \text{for } I_d \in \{0, \dots, n-1\}^d.$$

Let us quantify, in the Wasserstein-q distance, the proximity of $\Phi_{T\#}\mu_0$ to μ_*. We have

$$W_q(\mu(T), \mu_*) = W_q(\Phi_{T\#}\mu_0, \mu_*)$$
$$\leq \sum_{I_d \in \{0,\dots,n-1\}^d} W_q(\Phi_{T\#}\mu_0|_{G^\delta_{I_d}}, \mu_*|_{G^\delta_{I_d}}), \tag{4.15}$$

see Villani (2008) for the inequality. For each $I \in \{0, \dots, n-1\}^d$, let $\gamma_{I_d} : \mathbb{R}^d \to \mathbb{R}^d$ be the measurable function that satisfies

$$\gamma_{I_d\#}(\Phi_{T\#}\mu_0|_{G^\delta_{I_d}}) = \mu_*|_{G^\delta_{I_d}},$$

attaining the minimum in (2.7) for W_q. In particular, γ_{I_d} only redistributes the mass inside $G^\delta_{I_d}$, so

$$\int_{\mathbb{R}^d} |x - \gamma_{I_d}(x)|^q d\mu_*|_{G^\delta_{I_d}} = \int_{G^\delta_{I_d}} |x - \gamma_{I_d}(x)|^q d\mu_*$$
$$\leq \operatorname{diam}(G^\delta_{I_d})^q \int_{G^\delta_{I_d}} d\mu_* \leq \frac{3^{q+d} d^{q/2}}{n^{q+d}}.$$

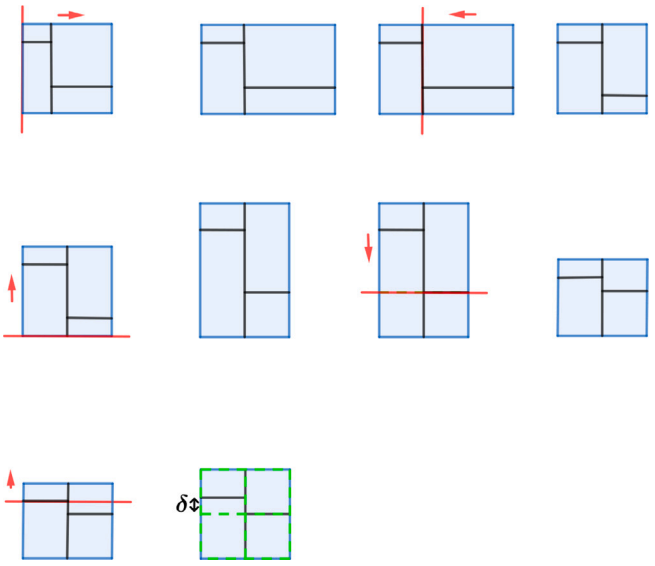


Fig. 10. From left to right, by rows: Transformation, step by step, of each rectangle C_{I_d} to the corresponding $G^{\delta}_{I_d}$ (in green).

Plugging this bound into (4.15), we conclude that

$$W_q(\mu(T),\mu_*) \le 3^{1+d/q}\sqrt{d}\, n^d \left(\frac{1}{n^{q+d}}\right)^{1/q} = 3^{1+d/q}\sqrt{d}\, n^{-(1+d/q-d)}.$$

It follows that

$$W_q(\mu(T),\mu_*) \xrightarrow{n\to\infty} 0 \iff q < \frac{d}{d-1},$$

and, in that case, $W_q(\mu(T),\mu_*) < \varepsilon$ is obtained for

$$n > \left(\frac{3^{1+d/q}\sqrt{d}}{\varepsilon}\right)^{\frac{1}{1+d/q-d}},$$

hence proving Theorem 8.

Proof of Lemma 13. We control the rectangles by mapping each hyperplane to its corresponding target hyperplane. The strategy is illustrated in Fig. 10.

The proof has three steps. First, we simultaneously control any p_k hyperplanes orthogonal to a fixed direction k. Second, for N hyperplanes, with $N \ge p_k$, we iteratively control p_k-subsets. Third, we show that this process can be done in all d canonical directions simultaneously.

Step 1. Let $k \in \{1,\dots,d\}$, $p_k \ge 1$ and

$$-\infty < c_1 < \cdots < c_{p_k} < \infty,$$
$$-\infty < g_1 < \cdots < g_{p_k} < \infty.$$

We aim to find controls $\{(\mathbf{w}_i,\mathbf{a}_i,b_i)\}_{i=1}^{p_k} \subset \mathbb{R}^d\times\mathbb{R}^d\times\mathbb{R}$ such that the flow generated by (1.3) satisfies

$$\boldsymbol{\Phi}_T(\{x^{(k)} = c_i\}) = \{x^{(k)} = g_i\}, \quad i = 1,\dots,p_k. \tag{4.16}$$

In particular, we will take $\mathbf{a}_i = \mathbf{e}_k$ for all i, and b_i such that $-b_i < \min\{c_i, g_i\}$. Since the field

$$\sum_{i=1}^{p_k} \mathbf{w}_i\sigma(\mathbf{a}_i\cdot\mathbf{x}+b_i) = \sum_{i=1}^{p_k}\mathbf{w}_i\sigma(x^{(k)}+b_i)$$

only depends on the $x^{(k)}$-coordinate, it is projectable onto the $x^{(k)}$-axis, i.e., the forward evolution of a hyperplane orthogonal to any coordinate-axis is a hyperplane orthogonal to the same coordinate-axis, at every time. We can thus identify each $\{x^{(k)} = c_i\}$ and $\{x^{(k)} = g_i\}$ with the point $c_i \in \mathbb{R}$ or $g_i \in \mathbb{R}$ and study its evolution in $\mathbb{R}$, so the problem becomes one-dimensional. Thus, we identify $x^{(k)} \equiv x$ and fix $\mathbf{w}_i = w_i\mathbf{e}_k$ with $w_i \in \mathbb{R}$, so we aim to find $\{w_i\}_{i=1}^{p_k} \subset \mathbb{R}$ and $\{b_i\}_{i=1}^{p_k} \subset \mathbb{R}$ such that

$$\boldsymbol{\Phi}_T(c_i) = g_i, \qquad \text{for } i = 1,\dots,p_k. \tag{4.17}$$

We proceed by induction on p_k.

First, we consider $p_k = 1$ and let $c_1, g_1 \in \mathbb{R}$. Take any $-b < \min\{c_1,g_1\}$ to ensure $\{c_1,g_1\} \subset \{x+b>0\}$, and

$$w_1 = \frac{1}{T}\ln\left(\frac{g_1+b}{c_1+b}\right),$$

so $\boldsymbol{\Phi}_T(c_1) = g_1$. Note that $\{x+b\le 0\}$ is fixed by $\boldsymbol{\Phi}_t$.

In the inductive step, we assume that the statement is true for some $p_k \ge 1$, and consider

$$-\infty < c_1 < \cdots < c_{p_k+1} < \infty,$$
$$-\infty < g_1 < \cdots < g_{p_k+1} < \infty.$$

Let $\{(w_i,b_i)\}_{i=1}^{p_k} \subset \mathbb{R}\times\mathbb{R}$ with $-b_i < \min\{c_i,g_i\}$ for all i, be such that (4.17) is satisfied. We want to add a new pair $(w_{p_k+1}, b_{p_k+1}) \in \mathbb{R}\times\mathbb{R}$ to further obtain:

$$\boldsymbol{\Phi}_T(c_i) = g_i, \quad i = 1,\dots,p_k+1.$$

Case 1: $c_{p_k} < g_{p_k}$. Take $b_{p_k+1} = -g_{p_k}$, so (1.3) becomes

$$\dot{x} = \sum_{i=1}^{p_k} w_i(x+b_i)\mathbb{1}_{c_i<x}(x) + w_{p_k+1}(x-g_{p_k})\mathbb{1}_{g_{p_k}<x}(x).$$

The added velocity $w_{p_k+1}(x-g_{p_k})$ only acts on the half-space $\{x \ge g_{p_k}\}$, so the points $\{c_1,\dots,c_{p_k}\}$ are only subject to the drift field

$$d(x) := \sum_{i=1}^{p_k} w_i(x+b_i)\mathbb{1}_{c_i<x}(x).$$

Therefore, when adding (w_{p_k+1}, b_{p_k+1}), we still have

$$\boldsymbol{\Phi}_T(c_i) = g_i, \quad i = 1,\dots,p_k.$$

If $c_{p_k+1} \le g_{p_k}$, there exists $\tau \in [0,T]$ such that

$$\boldsymbol{\Phi}_\tau(c_{p_k+1}) = g_{p_k}.$$

Otherwise, if $c_{p_k+1} > g_{p_k}$, consider $\tau = 0$. Note that τ only depends on the p_k first neurons, on c_{p_k+1} and on g_{p_k}, and it is thus independent of (w_{p_k+1}, b_{p_k+1}). So, c_{p_k+1} is only subject to $d(x)$ for $t \in (0,\tau)$, and to $d(x) + w_{p+1}(x-g_{p_k})$ for $t \in (\tau,T)$. More precisely:

$$\frac{d}{dt}\boldsymbol{\Phi}_t(c_{p_k+1}) = d(\boldsymbol{\Phi}_t(c_{p_k+1})) + w_{p_k+1}(\boldsymbol{\Phi}_t(c_{p_k+1}) - g_{p_k})\mathbb{1}_{\tau\le t<T}(t).$$

Then, with a similar computation to that of Lemma 11:

$$\begin{aligned}\boldsymbol{\Phi}_T(c_{p_k+1}) &= \boldsymbol{\Phi}_{T-\tau}\circ\boldsymbol{\Phi}_\tau(c_{p_k+1}) \\ &= \left(g_{p_k} + \frac{\sum_{i=1}^{p_k} w_ib_i - w_{p_k+1}g_{p_k}}{s_{p_k+1}}\right)e^{(T-\tau)s_{p_k+1}} \\ &\quad - \frac{\sum_{i=1}^{p_k} w_ib_i - w_{p_k+1}g_{p_k}}{s_{p_k+1}},\end{aligned}$$

where we have reused notation $\mathbf{s}_n = \sum_{i=1}^n \mathbf{w}_i$ for every n. It follows that

$$\lim_{w_{p_k+1}\to-\infty}\boldsymbol{\Phi}_T(c_{p_k+1}) = g_{p_k}, \qquad \lim_{w_{p_k+1}\to\infty}\boldsymbol{\Phi}_T(c_{p_k+1}) = \infty.$$

Therefore, by continuity, and since $g_{p_k} < g_{p_k+1}$, there exists $w_{p_k+1} \in \mathbb{R}$ such that $\boldsymbol{\Phi}_T(c_{p_k+1}) = g_{p_k+1}$.

Case 2: $g_{p_k} \le c_{p_k} < g_{p_k+1}$. Take $b_{p_k+1} = -c_{p_k}$, so (1.3) becomes

$$\dot{x} = \sum_{i=1}^{p_k} w_i(x+b_i)\mathbb{1}_{c_i<x}(x) + w_{p_k+1}(x-c_{p_k})\mathbb{1}_{c_{p_k}<x}(x),$$

Again, the points $\{c_i\}_{i=1}^{p_k}$ are only subject to $d(x)$ because the added velocity only acts on $\{x \geq c_{p_k}\}$, so

$$\Phi_T(c_i) = g_i, \quad i = 1, \dots, p_k.$$

The point c_{p_k+1} is subject to the total velocity at $t = 0$. A similar computation to Case 1 leads to

$$\Phi_T(c_{p_k+1}) = \left(c_p + \frac{\sum_{i=1}^{p_k} w_i b_i - w_{p_k+1} c_{p_k}}{s_{p_k+1}} \right) e^{T s_{p_k+1}} - \frac{\sum_{i=1}^{p_k} w_i b_i - w_{p_k+1} c_{p_k}}{s_{p_k+1}},$$

which satisfies

$$\lim_{w_{p_k+1} \to -\infty} \Phi_T(c_{p_k+1}) = c_{p_k}, \quad \lim_{w_{p_k+1} \to \infty} \Phi_T(c_{p_k+1}) = \infty.$$

By continuity, and since $c_{p_k} < g_{p_k+1}$, there exists $w_{p_k+1} \in \mathbb{R}$ such that $\Phi_T(c_{p_k+1}) = g_{p_k+1}$.

Case 3: $g_{p_k+1} \leq c_{p_k}$. Take $b_{p_k+1} = -c_{p_k}$, so (1.3) becomes

$$\dot{x} = \sum_{i=1}^{p_k} w_i (x + b_i) \mathbb{1}_{c_i < x}(x) + w_{p_k+1}(x - c_{p_k}) \mathbb{1}_{c_{p_k} < x},$$

and

$$\Phi_T(c_i) = g_i, \quad i = 1, \dots, p_k.$$

Let $0 < \tau \leq T$ be the first time such that $\Phi_\tau(c_{p_k+1}) = c_{p_k}$. An analogous computation to the previous cases gives

$$\Phi_\tau(c_{p_k+1}) = \left(c_{p_k+1} + \frac{\sum_{i=1}^{p_k} w_i b_i - w_{p_k+1} c_{p_k}}{s_{p_k+1}} \right) e^{\tau s_{p_k+1}} - \frac{\sum_{i=1}^{p_k} w_i b_i - w_{p_k+1} c_{p_k}}{s_{p_k+1}},$$

so

$$s = \frac{1}{s_{p_k+1}} \log \left(\frac{\sum_{i=1}^{p_k} w_i (c_{p_k} - b_i) + 2 w_{p_k+1} c_{p_k}}{\sum_{i=1}^{p_k} w_i (c_{p_k+1} + b_i) + w_{p_k+1}(c_{p_k+1} - c_p)} \right).$$

By varying w_{p_k+1} in $(-\sum_{i=1}^{p_k} w_i, +\infty)$, we can ensure that τ can take any value in $(0, T)$. Then,

$$\Phi_T(c_{p_k+1}) = \Phi_{T-\tau} \circ \Phi_\tau(c_{p_k+1}) = \left(c_{p_k} + \frac{\sum_{i=1}^{p_k} w_i b_i}{s_{p_k}} \right) e^{(T-\tau) s_{p_k}} - \frac{\sum_{i=1}^{p_k} w_i b_i}{s_{p_k}}.$$

For $\tau = T$, the expression is equal to c_{p_k}, and for $\tau = 0$ it equals $\Phi_T(c_{p_k}) = g_p$. By continuity, and also because $g_{p_k} < g_{p_k+1} \leq c_{p_k}$, an argument like in case 1 ensures the existence of $w_{p_k+1} \in \mathbb{R}$ such that $\Phi_T(c_{p_k+1}) = g_{p_k+1}$.

Step 2. Let $k \in \{1, \dots, d\}$, $p_k \geq 1$, and $N \geq p_k$ and

$$-\infty < c_1 < \cdots < c_N < \infty, \quad -\infty < g_1 < \cdots < g_N < \infty.$$

We will show that there exist piecewise constant controls $(w_i, a_i, b_i)_{i=1}^{p_k}$ such that the flow of (1.3) satisfies

$$\Phi_T(c_i) = g_i, \quad i = 1, \dots, N,$$

and the number of discontinuities is $\lceil N/p_k \rceil - 1$. We use a similar argument to the one in the proof of Theorem 1. We divide $\{c_i\}_{i=1}^N$ and $\{g_i\}_{i=1}^N$ into subsets of p points

$$C_j := \{c_{(j-1)\cdot p_k+1}, \dots, c_{j \cdot p_k}\},$$
$$G_j := \{g_{(j-1)\cdot p_k+1}, \dots, g_{j \cdot p_k}\},$$

for $j = 1, \dots, \lceil N/p_k \rceil - 1$, and $C_{\lceil N/p_k \rceil}$, $G_{\lceil N/p_k \rceil}$ with the remaining $N - p_k \lfloor N/p_k \rfloor$ points.

The piecewise constant controls are obtained by induction on j. In each iteration, we apply step 1 to define p_k constant controls $(w_i^j, a_i^j, b_i^j)_{i=1}^{p_k}$ that map the p_k points of C_j to the corresponding ones in G_j in time $\frac{T}{\lceil N/p_k \rceil}$. Note that the initialization of induction in step 1 ensures that the previously controlled subsets $C_1, \dots, C_{j-1}$ can remain fixed during the subsequent iterations, which trivializes the induction. Finally, we have the piecewise constant controls

$$(w_i, a_i, b_i)_{i=1}^{p_k} = \sum_{j=1}^{\lceil N/p_k \rceil} (w_i^j, a_i^j, b_i^j)_{i=1}^{p_k} \mathbb{1}_{\left(\frac{(j-1)T}{\lceil N/p_k \rceil}, \frac{jT}{\lceil N/p_k \rceil} \right)}(t),$$

which achieve the desired objective.

Step 3. For every $k = 1, \dots, d$, let $p_k, N_k \geq 1$, and

$$-\infty < c_1^k < \cdots < c_{N_k}^k < \infty, \ -\infty < g_1^k < \cdots < g_{N_k}^k < \infty.$$

For each fixed direction $k \in \{1, \dots, d\}$, step 2 is used to build piecewise constant controls

$$(\mathbf{w}_{j,k}, \mathbf{a}_{j,k}, b_{j,k})_{j=1}^{p_k} = (w_{j,k}\, \mathbf{e}_k, \mathbf{e}_k, b_{j,k})_{j=1}^{p_k}$$

with $\lceil N_k/p_k \rceil - 1$ discontinuities, such that

$$\Phi_T(\{x^{(k)} = c_i^k\}) = \{x^{(k)} = g_i^k\}, \quad i = 1, \dots, N.$$

Moreover, only the kth coordinate is varying on each flow, as argued in steps 1 and 2 when we simplified the problem to one dimension. We define each tern of the assembled control $(\mathbf{w}_j, \mathbf{a}_j, b_j)_{j=1}^p$, with $p := \sum_{k=1}^d p_k$, by

$$(\mathbf{w}_j, \mathbf{a}_j, b_j) = \left(\mathbf{w}_{j - \sum_{i=1}^{k-1} p_i, p_k}, \mathbf{a}_{j - \sum_{i=1}^{k-1} p_i, p_k}, b_{j - \sum_{i=1}^{k-1} p_i, p_k} \right),$$

for $\sum_{i=1}^{k-1} p_i + 1 \leq j \leq \sum_{i=1}^{k} p_i$. Therefore, the resulting neural ODE (1.3) on each coordinate writes

$$\dot{x}^{(k)} = w_{1,k}^{(k)}(x^{(k)} + b_{1,k}) + \cdots + w_{p_k,k}^{(k)}(x^{(k)} + b_{p_k,k})$$

All the equations of the system are independent, so each movement does not interfere with the other $d - 1$ movements. Therefore, the corresponding flow of (1.3) satisfies

$$\Phi_T(\{x^{(k)} = c_i^k\}) = \{x^{(k)} = g_i^k\}$$

for $k = 1, \dots, d$ and $i = 1, \dots, N_k$, and moreover, the number of discontinuities in the controls is

$$L = \max\{\lceil N_1/p_1 \rceil - 1, \dots, \lceil N_d/p_d \rceil - 1\} = \max_{k=1,\dots,d} \lceil N_k/p_k \rceil - 1.$$

Recalling that $N_k = n^k$ for $k = 1, \dots, d$ (by construction of the rectangles $C_{I_d}^0$ and $G_{I_d}^\delta$), it follows the desired result. □

CRediT authorship contribution statement

Antonio Álvarez-López: Writing – review & editing, Writing – original draft, Visualization, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Arselane Hadj Slimane:** Writing – original draft, Visualization, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Enrique Zuazua:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

This paper was 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). A. Álvarez-López has been funded by a contract FPU21/05673 from the Spanish Ministry of Universities. A. Hadj Slimane has been funded by École Normale Supérieur Paris-Saclay and Université Paris-Saclay. E. Zuazua has been funded by the Alexander von Humboldt-Professorship program, ModConFlex Marie Curie Action, HORIZON-MSCA-2021-DN-01, COST Action MAT-DYN-NET, Transregio 154 Project "Mathematical Modelling, Simulation and Optimization Using the Example of Gas Networks" of the DFG, grants PID2020-112617GB-C22 and TED2021-131390B-I00 of MICINN (Spain).

References

Agrachev, A. A., & Sarychev, A. V. (2021). Control on the manifolds of mappings with a view to the deep learning. *Journal of Dynamical and Control Systems*, *28*, 989–1008.

Alvarez-Lopez, A., Orive-Illera, R., & Zuazua, E. (2023). Optimized classification with neural ODEs via separability. arXiv:2312.13807.

Bach, F. (2017). Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, *18*(19), 1–53.

Chang, B., Meng, L., Haber, E., Tung, F., & Begert, D. (2018). Multi-level residual networks from dynamical systems view. arXiv:1710.10348.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., & Duvenaud, D. (2018). Neural ordinary differential equations. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 6572–6583). Curran Associates Inc.

Cheng, J., Li, Q., Lin, T., & Shen, Z. (2023). Interpolation, approximation and controllability of deep neural networks. arXiv:2309.06015.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314.

DeVore, R., Hanin, B., & Petrova, G. (2021). Neural network approximation. *Acta Numerica*, *30*, 327–444.

Dupont, E., Doucet, A., & Teh, Y. W. (2019). Augmented neural ODEs. In *Proceedings of the 33rd international conference on neural information processing systems* (pp. 3140–3150). Curran Associates Inc.

Duprez, M., Morancey, M., & Rossi, F. (2017). Controllability and optimal control of the transport equation with a localized vector field. In *2017 25th mediterranean conference on control and automation* (pp. 74–79).

E, W. (2017). A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, *5*, 1–11.

Elamvazhuthi, K., Gharesifard, B., Bertozzi, A. L., & Osher, S. (2022). Neural ODE control for trajectory approximation of continuity equation. *IEEE Control Systems Letters*, *6*, 3152–3157.

Eldan, R., & Shamir, O. (2015). The Power of Depth for Feedforward Neural Networks. *JMLR: Workshop and Conference Proceedings*, *49*, 1–34.

Esteve, C., Geshkovski, B., Pighin, D., & Zuazua, E. (2021). Large-time asymptotics in deep learning. arXiv:2008.02491.

Esteve-Yagüe, C., & Geshkovski, B. (2023). Sparsity in long-time control of neural ODEs. *Systems & Control Letters*, *172*, Paper No. 105452, 14.

Fan, F., Lai, R., & Wang, G. (2020). Quasi-equivalence of width and depth of neural networks.

Fernández-Cara, E., & Zuazua, E. (2000). The cost of approximate controllability for heat equations: the linear case. *Advances in Differential Equations*, *5*(4–6), 465–514.

Haber, E., & Ruthotto, L. (2017). Stable architectures for deep neural networks. *Inverse Problems*, *34*(1), Article 014004.

Hardt, M., & Ma, T. (2017). Identity matters in deep learning. In *International conference on learning representations*.

Huang, G.-B. (2003). Learning capability and storage capacity of two-hidden-layer feedforward networks. *IEEE Transactions on Neural Networks*, *14*(2), 274–281.

Kobyzev, I., Prince, S. J., & Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *43*(11), 3964–3979.

Li, Q., Lin, T., & Shen, Z. (2022). Deep learning via dynamical systems: An approximation perspective. *Journal of the European Mathematical Society*, *25*(5), 1671–1709.

Lin, H., & Jegelka, S. (2018). ResNet with one-neuron hidden layers is a universal approximator. In *Proceedings of the 32nd international conference on neural information processing systems* (pp. 6172–6181).

Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The expressive power of neural networks: A view from the width. In *Proceedings of the 31st international conference on neural information processing systems* (pp. 6232–6240). Red Hook, NY, USA: Curran Associates Inc.

Massaroli, S., Poli, M., Park, J., Yamashita, A., & Asama, H. (2020). Dissecting neural odes. *Advances in Neural Information Processing Systems*, *33*, 3952–3963.

Mhaskar, H., Liao, Q., & Poggio, T. (2017). When and why are deep networks better than shallow ones? *31*, In *Proceedings of the AAAI conference on artificial intelligence*. (1).

Nitti, N. D., & Fernández-Real, X. (2024). Optimal transport of measures via autonomous vector fields. arXiv:2405.06503.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., & Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, *22*(1).

Pinkus, A. (1999). Approximation theory of the MLP model in neural networks. *Acta Numerica*, *8*, 143–195.

Ruiz-Balet, D., & Zuazua, E. (2023). Neural ODE control for classification, approximation, and transport. *SIAM Review*, *65*(3), 735–773.

Ruiz-Balet, D., & Zuazua, E. (2024). Control of neural transport for normalising flows. *Journal de Mathématiques Pures et Appliquées*, *181*, 58–90.

Scagliotti, A. (2023). Deep learning approximation of diffeomorphisms via linear-control systems. *Mathematical Control and Related Fields*, *13*(3), 1226–1257.

Tabuada, P., & Gharesifard, B. (2023). Universal approximation power of deep residual neural networks through the lens of control. *IEEE Transactions on Automatic Control*, *68*(5), 2715–2728.

Valentine, F. (1945). A Lipschitz condition preserving extension for a vector function. *American Journal of Mathematics*, *67*(1), 83–93.

Villani, C. (2008). *Optimal transport – Old and new*: *Vol. 338*, (p. xxii+973). Springer Berlin, Heidelberg.

Yun, C., Sra, S., & Jadbabaie, A. (2019). Small ReLU networks are powerful memorizers: A tight analysis of memorization capacity. In *Proceedings of the 33rd international conference on neural information processing systems* (pp. 15558–15569). Curran Associates Inc.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *Communications of the ACM*, *64*.