SIAM J. MATH. DATA SCI.
Vol. 2, No. 3, pp. 901–919

Deep Neural Networks, Generic Universal Interpolation, and Controlled ODEs*

Christa Cuchiero†, Martin Larsson‡, and Josef Teichmann§

Abstract. A recent paradigm views deep neural networks as discretizations of certain controlled ordinary differential equations, sometimes called neural ordinary differential equations. We make use of this perspective to link expressiveness of deep networks to the notion of controllability of dynamical systems. Using this connection, we study an expressiveness property that we call universal interpolation and show that it is generic in a certain sense. The universal interpolation property is slightly weaker than universal approximation and disentangles supervised learning on finite training sets from generalization properties. We also show that universal interpolation holds for certain deep neural networks even if large numbers of parameters are left untrained and are instead chosen randomly. This lends theoretical support to the observation that training with random initialization can be successful even when most parameters are largely unchanged through the training. Our results also explore what a minimal amount of trainable parameters in neural ordinary differential equations could be without giving up on expressiveness.

Key words. deep neural networks, universal interpolation, controlled dynamical systems, Lie brackets, Hörmander condition

AMS subject classifications. 93B05, 93B15, 34F05

DOI. 10.1137/19M1284117

1. Deep neural networks as controlled ODEs.

Several recent studies of deep neural networks revolve around the idea of viewing such networks as discretizations of ordinary differential equations (ODEs). This led to the terminology *neural ODEs*, a perspective which has successfully been applied to a number problems; see, e.g., E (2017); Chang et al. (2017); Chen et al. (2018); Grathwohl et al. (2018); and Dupont, Doucet, and Teh (2019) among many others. See also E, Han, and Li (2018) and Liu and Markowich (2019) for mathematically rigorous analyses. In this paper we make progress towards a theoretical understanding of this success. Using ideas from dynamical systems and control theory, we show why it can be beneficial to view deep neural networks as discretized controlled ODEs. Our analysis suggests that randomization of the vector fields can be used to substantially reduce the number of trainable parameters. This sheds new light on random initialization of deep neural networks with fully trainable parameters.

*Received by the editors August 28, 2019; accepted for publication (in revised form) July 16, 2020; published electronically September 28, 2020.

https://doi.org/10.1137/19M1284117

Funding: The first author gratefully acknowledges financial support by the Vienna Science and Technology Fund (WWTF) under grant MA16-021. The third author gratefully acknowledges financial support by the Swiss National Science Foundation (SNF) under grant 179114.

†Department of Statistics and Operations Research, Data Science, University of Vienna, Vienna, Austria (christa.cuchiero@univie.ac.at).

‡Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213 USA (martinl@andrew.cmu.edu).

§Department of Mathematics, ETH Zurich, Zurich, 8092, Switzerland (jteichma@math.ethz.ch).

The approach in E (2017), Chen et al. (2018), and Liu and Markowich (2019) rests on the observation that the input X_k to any given layer k is mapped to an output X_{k+1} that can be expressed as a residual network style transition (He et al. (2015)) of the form $X_{k+1} = X_k + V(X_k, \theta_k)$. The right-hand side depends both on the input X_k and on a parameter vector θ_k, both of which vary from layer to layer.

The representation of X_{k+1} as a perturbation of X_k suggests that for sufficiently deep networks, the cumulative effect of repeated transitions mimics the behavior of an ODE. This ODE can then be studied instead of the original network. The discrete parameter $k = 0, 1, 2, \dots$ that counts the layers is replaced by a continuous parameter $t \in [0, 1]$, and one lets the "state" X_t at "layer" t evolve according to a law of motion of the form

$$\frac{d}{dt} X_t = V(X_t, \theta_t). \tag{1.1}$$

In other words, one views depth as the running time of a dynamical system. The solution X_t of (1.1) forms a curve through its state space, which we here take to be $\mathbb{R}^m$ for some fixed dimension m, and θ_t represents a curve through the space of possible parameters. Given an initial condition $x \in \mathbb{R}^m$, we let X_t^x denote the corresponding solution of (1.1), subject to

$$X_0^x = x.$$

For all choices of $V(x, \theta)$ and θ_t considered in this paper, the solution of (1.1) exists and is unique. The following example connects (1.1) with standard neural network architectures.

Example 1.1. In a standard (residual) neural network layer, the components of $V(x, \theta)$ are of the form $V^j(x, \theta) = b^j + \sum_{k=1}^m a_k^j \sigma(x^k)$ for $j = 1, \dots, m$, where the parameters a_k^j, b^j make up the vector θ, and $\sigma(\,\cdot\,)$ is a fixed nonlinearity acting on the components of $x = (x^1, \dots, x^m)$. In this example, somewhat oddly, we let the nonlinearity act *before* the affine map. However, the ordering is inessential when multiple layers are composed, because the nonlinearity takes the affine map from the previous layer as input. The choice made here will be convenient in later examples.

For an input $x \in \mathbb{R}^m$, the "continuous-depth" network (1.1) outputs X_1^x. This is, however, still a vector in $\mathbb{R}^m$ and will usually be mapped to a much lower dimensional output, say, $R(X_1^x)$ for some readout map $R\colon \mathbb{R}^m \to \mathbb{R}^{m'}$ with $m' \ll m$. Supervised learning in this framework amounts to the following: for a given training set of input/ouput pairs, $(x_i, y_i) \in \mathbb{R}^m \times \mathbb{R}^{m'}$ for $i = 1, \dots, N$, identify parameters θ_t, $t \in [0, 1]$, and a readout map R such that $R(X_1^{x_i}) \approx y_i$ for all i, perhaps while imposing a regularization penalty on θ_t. Our results are formulated for $m' = m$ with either the identity readout, leading to $x \mapsto X_1^x$, or the readout structure $x \mapsto \lambda(X_1^x - x)$ that depends directly on the input data and a trained scalar parameter $\lambda > 0$.

In the present paper we recognize (1.1) as a *controlled ordinary differential equation* (CODE) and the training task as a problem of optimal control. One of our key motivations is to show that it is actually not necessary to train all parameters. Only a minority needs to be trained. We capture this idea by decomposing $V(x, \theta)$ in a way where the dependence on the trainable parameters enters linearly, which corresponds to the most natural

and simplest parameterization. Indeed, our results will be proved in the following setting. Suppose the function $V(x,\theta)$ determining the right-hand side of (1.1) is of the form

$$V(x,\theta) = u^1 V_1(x) + \cdots + u^d V_d(x), \tag{1.2}$$

where $u^1,\dots,u^d$ are scalar parameters and $V_1,\dots,V_d$ are smooth vector fields on $\mathbb{R}^m$.[1] We think of $u^1,\dots,u^d$ as trainable parameters (thus part of θ) that will be t-dependent. The vector fields $V_1,\dots,V_d$ are specified by the remaining parameters in θ, which will be nontrainable and constant in t. The following example illustrates that this decomposition is in line with the standard neural network architecture of Example 1.1.

Example 1.2. Recall the standard architecture of Example 1.1, where each layer depends on $m+m^2$ parameters. If each vector field $V_i(x)$ is of this form, then so is $V(x,\theta)$ in (1.2). To see this, suppose $V_i^j(x) = b_i^j + \sum_{k=1}^m a_{ik}^j \sigma(x^k)$ for some parameters b_i^j, a_{ik}^j. Then $V^j(x,\theta) = b^j + \sum_{k=1}^m a_k^j \sigma(x^k)$ with $b^j = \sum_{i=1}^d u^i b_i^j$ and $a_k^j = \sum_{i=1}^d u^i a_{ik}^j$, which again has the standard form in Example 1.1. This construction should be viewed as one way of decomposing the full parameter set into trainable and nontrainable parameters. In fact, in this example, the number of trainable parameters per layer is d, which should be thought of as being much smaller than the number of nontrainable parameters $m+m^2$. A key message of our results is that similar reductions in the number of nontrainable parameters are possible in the CODE setting without compromising expressive power.

With the specification (1.2), the CODE (1.1) takes the form

$$\frac{d}{dt}X_t = u_t^1 V_1(X_t) + \cdots + u_t^d V_d(X_t), \tag{1.3}$$

where $u_t^1,\dots,u_t^d$ are the controls (the trainable parameters). As before, if the initial condition is x, the solution is denoted by X_t^x. The output is X_1^x, or if composed with a readout, $R(X_1^x)$. If the controls are square-integrable functions of t and the vector fields are smooth and bounded (i.e., $\sup_{x\in\mathbb{R}^m}\|V_i(x)\| < \infty$ for all i), one has existence and uniqueness of solutions of (1.3) for every initial condition.

The system (1.3) turns out to be remarkably expressive if the vector fields are chosen appropriately. Our goal in this paper is to make this statement rigorous. In section 2 we establish Theorem 2.2, which states that one can match any training set of finite size using just $d=5$ suitably chosen vector fields $V_1,\dots,V_5$. That is, for any finite set of input/ouput pairs $(x_i,y_i)\in\mathbb{R}^m\times\mathbb{R}^m$, there exist controls such that $X_1^{x_i} = y_i$ for all i. We refer to this property as the *universal interpolation property*. This differs from the well-studied notion of universal approximation (e.g., Cybenko (1989); Hornik (1991)), and makes no statement about generalization properties. Let us stress that we do not claim that perfect interpolation is necessarily a desirable training goal. Still, we believe it serves as a useful measure of expressiveness. Moreover, recent work on the so-called *double-descent phenomenon* has shown that even when machine learning models interpolate the training set, they can still generalize well on unseen data; see, e.g., Ma, Bassily, and Belkin (2018); Belkin, Hsu, and Mitra (2018); and Liang and Rakhlin (2018). For classical results on interpolation via neural networks,

[1] That is, the V_i are smooth maps from $\mathbb{R}^m$ to $\mathbb{R}^m$.

e.g., multilayer feedforward perceptrons, we refer to Pinkus (1999, Theorem 5.1). In contrast to this classical theorem, our result does not depend on the number of training samples that one aims to match. Recently, universal approximation of neural ODEs has been considered in Zhang et al. (2019), where the authors prove that certain homeomorphism on $\mathbb{R}^p$ can be embedded into flows of CODEs on $\mathbb{R}^{2p}$. One essential difference to our results is the question of minimal controllability of the flows, which is not addressed in Zhang et al. (2019). No-go results have been shown in Dupont, Doucet, and Teh (2019). These results do not contradict our findings as we only work with finite training data sets.

The proofs of our results rely on mathematical machinery from control theory, involving classical notions like Lie brackets and controllability. This is reviewed in section 3. In addition to laying the groundwork for the proofs, we aim to convey the intuition for why control theory can help explain expressiveness in deep learning. The formal proof of Theorem 2.2 is then given in section 4, with some lengthier computations postponed to the appendix.

In section 5 we go further by showing that not only are five vector fields enough, they can be chosen randomly in the class of real analytic vector fields. We make this precise in Theorem 5.1. As a consequence, common structures such as the one in Example 1.1 (with real analytic nonlinearities such as the standard functions $\arctan(x)$ or $\tanh(x)$) can be shown to retain this strong form of expressiveness. This is done in Corollary 5.4.

We do not make any statement about optimality of these generic expressive networks for specific learning tasks. However, our analysis produces the remarkable conclusion that deep neural networks, expressed as discretizations of (1.3) with only five random vector fields, can interpolate any functional relation with a precision that depends only on depth and the amount of training data. Our approach supports the "folklore" statement that randomness is of great importance for training. Indeed, the role of randomness, which is ubiquitous in training procedures (*stochastic* gradient descent, *random* initialization of weights, etc.), receives a theoretical basis through Theorem 5.1. In section 5, we comment on these algorithmic aspects, although we do not perform any empirical analysis in this paper. The proof of Theorem 5.1 is given in section 6.

A full-fledged geometric and quantitative analysis in a very general analytic setting is performed in the companion paper Cuchiero, Larsson, and Teichmann (2019). There $\mathbb{R}^m$ is replaced by a so-called convenient vector space, covering various infinite-dimensional situations of interest. We give a new proof of the Chow–Rashevskii theorem and present quantitative results on training CODEs. This lets us analyze controlled transport equations or PDEs, as well as the effect of convolutional layers.

2. Universal interpolation.

Interpreting (1.1) and (1.3) as CODEs establishes an interface to control theory. This opens the door to powerful mathematical techniques that we will deploy to establish an expressiveness property that we call *universal interpolation*. When satisfied, this property guarantees that any supervised learning task has a solution. It is formalized in the following definition, which uses the identity readout $R(x) = x$.

Definition 2.1. *The control system* (1.3)*, specified by the vector fields* $V_1, \dots, V_d$*, is called a* universal N-point interpolator *on a subset* $\Omega \subseteq \mathbb{R}^m$ *if, for any training set* $\{(x_i, y_i) \in \Omega \times \Omega \colon i = 1, \dots, N\}$ *of size* N*, there exist controls* $u_t^1, \dots, u_t^d$ *that achieve the exact matching*

$X_1^{x_i} = y_i$ for all $i = 1, \dots, N$. Here it is required that the training inputs $x_1, \dots, x_N$, as well as the targets $y_1, \dots, y_N$, are pairwise distinct.[2]

Universal N-point interpolation may look like a rather strong requirement, especially if the size N of the training set and/or the ambient dimension m is large. Clearly, this property is primarily of interest if the number d of vector fields can be chosen small compared to N and m. Our first main result states that, in a striking manner, this is always possible.

Theorem 2.2. *Fix $m \geq 2$ and a bounded open connected subset $\Omega \subset \mathbb{R}^m$. There exist $d = 5$ smooth bounded vector fields $V_1, \dots, V_5$ on $\mathbb{R}^m$ such that (1.3) is a universal N-point interpolator in Ω for every N.*

The formal proof of Theorem 2.2 is presented in section 4, building on classical ideas from control theory reviewed in section 3. Before discussing the proof, let us comment on the content of the theorem.

First, observe that $V_1, \dots, V_5$ do not depend on N. Thus the *same* five vector fields can be used to interpolate any arbitrary (but finite) training set. Of course, the controls $u_t^1, \dots, u_t^d$ that achieve interpolation do depend on the training set. If the training set changes, for example, if it is augmented with additional training pairs, the controls will generally change as well.

Next, the vector fields themselves depend on the ambient dimension m, by the very definition of a vector field on $\mathbb{R}^m$. However, we stress that no matter how large m is, $d = 5$ vector fields always suffice to achieve universal interpolation for arbitrarily large training sets.

Further, the case $m = 1$ is not covered. This reflects the fact that N points $x_1, \dots, x_N$ on the real line cannot be continuously transported to targets $y_1, \dots, y_N$ without intersecting, if the inputs and targets are ordered differently. Such a training task cannot be achieved by (1.3), since trajectories $\{X_t \colon t \in [0, 1]\}$ corresponding to different initial conditions always remain disjoint.

Finally, Theorem 2.2 is an existence result with no quantitative estimates on, for example, the size of the controls $u_t^1, \dots, u_t^d$ needed to achieve interpolation. Similarly, nothing is asserted regarding the behavior of the map $x \mapsto X_1^x$ away from the training inputs x_i. In practice, one does not insist on exact interpolation but trades off accuracy for more regular controls. A rigorous analysis of these issues would be of great interest, though it is not the subject of this paper. Here we only provide the following proposition which states the form of the first derivative of $x \mapsto X_1^x$, along with a bound on its size in terms of the size of the vector fields and controls. The derivative of $x \mapsto X_t^x$ at a point x (called *first variation*) is an $m \times m$ matrix that we denote by J_t^x for Jacobian.

Proposition 2.3. *Consider the CODE (1.3) under the assumptions of existence and uniqueness. Then J_t^x solves the linear differential equation*

$$\frac{d}{dt} J_t^x = \sum_{i=1}^{d} u_t^i DV_i(X_t^x) J_t^x, \quad t \in [0, 1], \tag{2.1}$$

[2] A system like (1.3) can never map different inputs to the same output. Moreover, it is not meaningful to pair one single input with two different outputs in the training set.

with initial value $J_0^x = I$ (the $m \times m$ identity matrix), where DV_i denotes the Jacobian of the vector field V_i. The operator norm of J_t^x is bounded by

$$\|J_t^x\|_{op} \leq \exp\left(\int_0^t \left\| \sum_{i=1}^d u_s^i \, DV_i(X_s^x) \right\|_{op} ds \right).$$

Proof. We obtain (2.1) by differentiating the equation $X_t^x = x + \sum_{i=1}^d \int_0^t u_s^i V_i(X_s^x)\, ds$ and applying the chain rule. To deduce the bound on $\|J_t^x\|_{\mathrm{op}}$, pick an arbitrary unit vector $z \in \mathbb{R}^d$, and use (2.1) along with the triangle inequality and the definition of the operator norm to get

$$\|J_t^x z\| \leq 1 + \int_0^t \left\| \sum_{i=1}^d u_s^i \, DV_i(X_s^x) \right\|_{\mathrm{op}} \|J_s^x z\| ds.$$

Gronwall's inequality yields $\|J_t^x z\| \leq \exp(\int_0^t \sum_{i=1}^d |u_s^i| \, \|DV_i(X_s^x)\|_{\mathrm{op}} ds)$. This implies the claimed bound on $\|J_t^x\|_{\mathrm{op}}$ since z was an arbitrary unit vector. ∎

Some related quantitative questions are discussed in the companion paper Cuchiero, Larsson, and Teichmann (2019).

3. Lie brackets and controllability.

In preparation for the proof of Theorem 2.2, and to aid intuition as to why such a small number of vector fields can result in a highly expressive system, we review some ideas from control theory. The developments take place in a generic Euclidean space $\mathbb{R}^n$; later we will take $n = mN$, where N is the size of the training set. As we do not assume the reader is familiar with this theory, we will give examples in an attempt to convey the underlying intuition.

Definition 3.1. *Let U, V, $U_1, \ldots, U_d$ be smooth vector fields on $\mathbb{R}^n$.*

- *The* Lie bracket *$[U, V]$ is the smooth vector field on $\mathbb{R}^n$ given by*

$$[U, V](x) = DV(x)\, U(x) - DU(x)\, V(x),$$

  where DU is the Jacobian matrix of partial derivatives; thus its (i, j) entry is $\partial U^i / \partial x^j$, and similarly for $DV(x)$.

- *The* Lie algebra generated by $U_1, \ldots, U_d$, *denoted by* $\mathrm{Lie}(U_1, \ldots, U_d)$, *is the smallest linear space of vector fields that contains $U_1, \ldots, U_d$ and is stable under Lie brackets. Equivalently, we have*

$$\mathrm{Lie}(U_1, \ldots, U_d) = \mathrm{span}\{U_1, \ldots, U_d \text{ and all iterated Lie brackets}\}.$$

  For any $x \in \mathbb{R}^n$, we also consider the subspace of $\mathbb{R}^n$ obtained by evaluating all the vector fields in the Lie algebra at x, namely,

$$\mathrm{Lie}(U_1, \ldots, U_d)(x) = \{W(x) \colon W \in \mathrm{Lie}(U_1, \ldots, U_d)\} \subseteq \mathbb{R}^n.$$

Let us look at the case of linear vector fields, where the Lie brackets have simple expressions.

Example 3.2. Consider linear vector fields $U(x) = Ax$ and $V(x) = Bx$, where A and B are $n \times n$ matrices. A direct calculation shows that $[U, V](x) = (AB - BA)x$. Therefore, $\mathrm{Lie}(U, V)$ consists of all linear vector fields of the form Cx, where C is obtained from A and B by taking matrix commutators and linear combinations finitely many times.

The main tool in the proof of Theorem 2.2 is the Chow–Rashevskii theorem, which can be stated as follows. For details, see Montgomery (2002, Chapter 2).

Theorem 3.3 (Chow–Rashevskii). *Let $\Omega \subseteq \mathbb{R}^n$ be an open connected subset, and assume the smooth bounded vector fields $U_1, \dots, U_d$ satisfy the* Hörmander condition*,*

$$\mathrm{Lie}(U_1, \dots, U_d)(x) = \mathbb{R}^n,$$

at every point $x \in \Omega$. Then controllability *holds: for every input/output pair $(x, y) \in \Omega \times \Omega$, there exist smooth scalar controls $u_t^1, \dots, u_t^d$ that achieve $X_1 = y$, where X_t is the solution of*

$$\frac{d}{dt} X_t = u_t^1 U_1(X_t) + \cdots + u_t^d U_d(X_t), \quad X_0 = x.$$

Example 3.4. To see why Lie brackets are relevant for controllability, it is useful to consider the case of linear vector fields $U(x) = Ax$ and $V(x) = Bx$. A particle starting at x and flowing along the vector field U for an amount of time t ends up at $e^{At}x$, where the standard matrix exponential is used. This is because $e^{At}x$ is the solution of $\frac{d}{dt}X_t = AX_t$, $X_0 = x$. Alternating between V, U, $-V$, and $-U$, therefore moves the particle from x to $e^{-At}e^{-Bt}e^{At}e^{Bt}x$. A Taylor expansion in t shows that

$$e^{-At}e^{-Bt}e^{At}e^{Bt}x = x + t^2(AB - BA)x + O(t^3)\,.$$

Therefore if t is small, the alternating behavior produces motion in the direction $(AB-BA)x = [U, V](x)$. For general vector fields, an analogous computation gives the same result. The Chow–Rashevskii theorem is now quite intuitive: controllability holds if at each point one can produce motion in all directions. However, moving in the Lie bracket direction requires more "energy" (larger and more oscillatory controls), reflected by the short-time asymptotic t^2.

Example 3.5. To see that a small number of vector fields can generate very large Lie algebras, consider the two vector fields $U(x) = x^2$ and $V(x) = x^k$ on $\mathbb{R}$, where $k \in \mathbb{N}$. Note that vector fields on $\mathbb{R}$ are just scalar functions. Then $[U, V](x) = V'(x)U(x) - U'(x)V(x) = (k-2)x^{k+1}$. As a result, the Lie algebra generated by x^2 and x^3 contains all x^k, $k \geq 2$.

In the context of deep learning, one can view the Lie bracket operation as a way to generate *features*. This requires a large number of layers when brackets are iterated. Indeed, each layer is associated with an Euler step of the discretized CODE. Example 3.4 then shows that four layers are needed to move along the length-2 bracket $[U, V]$. The number of layers required to move along a general length-n bracket is exponential in n.

On the other hand, the dimensionality of the feature space generated in this way can also grow extremely quickly due to noncommutativity of Lie brackets. Let us illustrate this using the *free Lie algebra* on d generators $Y_1, \dots, Y_d$. This is an abstract Lie algebra whose elements are formal linear combinations of *Lie words* in the generators. A Lie word is a

formal expression involving the generators and the bracket $[\cdot,\cdot]$, for example, $[Y_1,[Y_2,Y_1]]$ and $[Y_2,[Y_1,Y_1]]$. Two Lie words are considered equal if they can be transformed into one another using the axioms satisfied by the bracket, namely, bilinearity, anticommutativity, and the Jacobi identity. For example, $[Y_2,[Y_1,Y_1]]=[Y_2,0]=0$. The dimension of the subspace $\mathcal{L}_n$ spanned by all Lie words of length n is given by *Witt's dimension formula*,

$$\dim \mathcal{L}_n = \frac{1}{n}\sum_{k|n}\mu(k)d^{n/k}$$

see Magnus, Karrass, and Solitar (1976, Theorem 5.11). Here the sum ranges over all k that divide n, and $\mu(\cdot)$ is the *Möbius function* which takes values in $\{-1,0,1\}$. The asymptotic behavior for large n is exponential,

$$\dim \mathcal{L}_n \sim d^n.$$

This is related to the fact that the Lie bracket is noncommutative. For comparison, the space of polynomials of degree at most n in d commuting variables has dimension $\binom{n+d}{d}\sim n^d$, which only grows polynomially in n.

If $U_1,\dots,U_d$ are smooth vector fields on $\mathbb{R}^n$ that are sufficiently unstructured or "generic," we expect the Lie algebra that they generate to behave similarly to the free Lie algebra on d generators. In particular, we expect the dimensionality of the feature space to grow very quickly. Notice, however, that the price to pay is exponentially growing depth to generate all brackets.

4. Universal N-point interpolators exist. In this section we apply the Chow-Rashevskii theorem and algebraic results on polynomial vector fields to prove Theorem 2.2. The proof is constructive and relies on five specific linear and quadratic vector fields $V_1,\dots,V_5$ for which (1.3) is a universal N-point interpolator.

We select an arbitrary N and work on the set $\overline{\Omega}\subset(\mathbb{R}^m)^N$ of pairwise distinct N-tuples $(x_1,\dots,x_N)$ of points in Ω. Here $m\geq 2$ is the ambient dimension, and N represents the number of training pairs as in section 2. In other words, we consider the bounded open connected subset

$$\overline{\Omega}=\Omega^N\setminus\Delta$$

of $(\mathbb{R}^m)^N$, where

$$\Delta=\{(x_1,\dots,x_N)\in\Omega^N\colon x_i=x_j \text{ for some } i\neq j\}.$$

($\overline{\Omega}$ is connected because $m\geq 2$.) Then, given d smooth bounded vector fields $V_1,\dots,V_d$ on $\mathbb{R}^m$, (1.3) is a universal N-point interpolator in Ω if and only if the "stacked" system

$$\frac{d}{dt}\begin{pmatrix}X_t^{x_1}\\ \vdots\\ X_t^{x_N}\end{pmatrix}=u_t^1\begin{pmatrix}V_1(X_t^{x_1})\\ \vdots\\ V_1(X_t^{x_N})\end{pmatrix}+\cdots+u_t^d\begin{pmatrix}V_d(X_t^{x_1})\\ \vdots\\ V_d(X_t^{x_N})\end{pmatrix}$$

can bring any initial point $\bar{x} = (x_1, \dots, x_N) \in \overline{\Omega}$ to any target $\bar{y} = (y_1, \dots, y_N) \in \overline{\Omega}$ by means of a suitable choice of controls $u_t^1, \dots, u_t^d$. By the Chow–Rashevskii theorem, this holds if and only if the stacked vector fields

$$V_i^{\oplus N}(\bar{x}) := \begin{pmatrix} V_i(x_1) \\ \vdots \\ V_i(x_N) \end{pmatrix}, \quad i = 1, \dots, d,$$

satisfy the Hörmander condition at every $\bar{x} = (x_1, \dots, x_N) \in \overline{\Omega}$. The following definition and subsequent lemma strongly hint at how we plan to verify the Hörmander condition.

Definition 4.1. *A collection $\mathcal{V}$ of vector fields on $\mathbb{R}^m$ is said to* interpolate at a tuple $(x_1, \dots, x_N) \in \overline{\Omega}$ *if for every $(v_1, \dots, v_N) \in (\mathbb{R}^m)^N$ there exists a vector field $\widehat{V} \in \mathcal{V}$ such that $\widehat{V}(x_i) = v_i$ for all $i = 1, \dots, N$.*

Lemma 4.2. *Let $V_1, \dots, V_d$ be smooth vector fields on $\mathbb{R}^m$ such that* $\mathrm{Lie}(V_1, \dots, V_d)$ *interpolates at the tuple $\bar{x} = (x_1, \dots, x_N) \in \overline{\Omega}$. Then*

$$\mathrm{Lie}\left(V_1^{\oplus N}, \dots, V_d^{\oplus N}\right)(\bar{x}) = (\mathbb{R}^m)^N;$$

that is, the vector fields $V_1^{\oplus N}, \dots, V_d^{\oplus N}$ satisfy the Hörmander condition at $\bar{x}$.

Proof. Pick any $\bar{v} = (v_1, \dots, v_N) \in (\mathbb{R}^m)^N$. Since $\mathrm{Lie}(V_1, \dots, V_d)$ interpolates at $\bar{x}$, it contains a vector field $\widehat{V}$ such that

$$\widehat{V}^{\oplus N}(\bar{x}) = \begin{pmatrix} \widehat{V}(x_1) \\ \vdots \\ \widehat{V}(x_N) \end{pmatrix} = \begin{pmatrix} v_1 \\ \vdots \\ v_N \end{pmatrix}.$$

Moreover, due to the identity $[V^{\oplus N}, W^{\oplus N}] = [V, W]^{\oplus N}$, which is valid for any smooth vector fields V, W on $\mathbb{R}^m$, it follows that $\mathrm{Lie}(V_1^{\oplus N}, \dots, V_d^{\oplus N})$ contains $\widehat{V}^{\oplus N}$. Therefore $\bar{v} \in \mathrm{Lie}(V_1^{\oplus N}, \dots, V_d^{\oplus N})(\bar{x})$, which completes the proof. ∎

We now confirm that the collection of all polynomial vector fields interpolates any number of pairwise distinct points.

Lemma 4.3. *The set of all polynomial vector fields on $\mathbb{R}^m$ interpolates at every tuple $(x_1, \dots, x_N) \in \overline{\Omega}$.*

Proof. The result follows by standard multivariate polynomial interpolation. Specifically, consider arbitrary $(x_1, \dots, x_N) \in \overline{\Omega}$ and $(v_1, \dots, v_N) \in (\mathbb{R}^m)^N$. Since the x_i are pairwise distinct, it is possible to find, for each $j = 1, \dots, m$, a polynomial $p^j(x)$ on $\mathbb{R}^m$ such that $p^j(x_i) = v_i^j$ for $i = 1, \dots, N$. The vector field

$$\widehat{V}(x) = \begin{pmatrix} p^1(x) \\ \vdots \\ p^m(x) \end{pmatrix}$$

is then polynomial and satisfies $\widehat{V}(x_i) = v_i$ for all $i = 1, \dots, N$. ∎

Thanks to the Chow–Rashevskii theorem as stated in Theorem 3.3, as well as Lemma 4.2 and 4.3, in order to prove Theorem 2.2 it only remains to exhibit five smooth vector fields that do not depend on N and whose Lie algebra contains all polynomial vector fields. This is accomplished by the following result, which therefore completes the proof of the theorem. (Note that we actually want *bounded* vector fields. This is easily achieved by multiplying the vector fields below by a smooth compactly supported function $\varphi(x)$ that equals one on Ω.)

Proposition 4.4. *There exist $d = 5$ smooth vector fields $V_1, \dots, V_5$ on $\mathbb{R}^m$ with the property that* $\mathrm{Lie}(V_1, \dots, V_5)$ *contains all polynomial vector fields. Specifically, one can take*

$$V_1(x) = Ax, \quad V_2(x) = Bx,$$

$$V_3(x) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad V_4(x) = \begin{pmatrix} (x^m)^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad V_5(x) = \begin{pmatrix} x^1 x^m \\ x^2 x^m \\ \vdots \\ (x^m)^2 \end{pmatrix},$$

where A and B are suitable traceless $m \times m$ matrices, and $x^1, \dots, x^m$ denote the components of x.[3]

Proof. We divide the proof into three separate statements that together imply the claimed result. We use $e_1, \dots, e_m$ to denote the canonical basis vectors in $\mathbb{R}^m$.

Claim 1. There is a choice of traceless $m \times m$ matrices A and B such that $\mathrm{Lie}(V_1, V_2) = \{Cx \colon C \text{ is traceless}\}$.

Indeed, Example 3.2 shows that $\{Cx \colon C \text{ is traceless}\}$ is a Lie algebra of vector fields that can be identified with the Lie algebra of all traceless $m \times m$ matrices. The latter is the *special linear Lie algebra* $\mathfrak{sl}_m(\mathbb{R})$, which is known to admit two generators A and B; see, for instance, Kuranishi (1951), where it is shown that in fact any semisimple Lie algebra admits two generators.

Claim 2. With A and B as above, $\mathrm{Lie}(V_1, V_2, V_3, V_4)$ contains all linear vector fields.

Indeed, we know it contains all vector fields Cx with C traceless. Moreover, it contains the Lie bracket $[V_3, V_4](x) = 2x^m e_1 = 2e_1 e_m^\top x$. Expressing the identity matrix $I = (I - me_1 e_m^\top) + me_1 e_m^\top$ as a sum of a traceless matrix and a multiple of $2e_1 e_m^\top$, it follows that the identity vector field $W(x) = x$ is in $\mathrm{Lie}(V_1, V_2, V_3, V_4)$. This proves the claim, since any matrix can be expressed as a traceless matrix plus a multiple of the identity.

Claim 3. V_3, V_4, and V_5 together with all linear vector fields generate all polynomial vector fields.

This is asserted without proof by Leites and Poletaeva (1997) and can be verified by direct computation. We do this in full detail in the appendix.

Combining Claim 2 and Claim 3 proves the proposition. ∎

Remark 4.5. The use of polynomials in the above proof is due to their relatively tractable structure. We believe the conclusion remains true for other classes of vector fields, also on curved spaces. For example, on the torus a natural choice would be to consider Fourier basis functions.

[3] A traceless matrix is one whose trace is equal to zero.

5. Generic expressiveness. Theorem 2.2 shows that universal interpolators can be constructed using just five vector fields, but not how common or rare such vector fields are. Our next goal is to prove that parsimonious yet expressive systems exist in great abundance. To do so, rather than using (1.3) to interpolate the outputs y_i directly, we will use it to interpolate the transformed outputs $x_i + \lambda^{-1} y_i$, where $\lambda > 0$ is a (trained) constant. Thus the input x and output y are related by

$$y = \lambda(X_1^x - x), \tag{5.1}$$

where the right-hand side can be interpreted as a particular readout map. Our next result shows that with five or more appropriately *randomly chosen* nonlinear vector fields, the system (1.3) and (5.1) is sufficiently expressive to interpolate almost every training set.

The setup of the theorem is as follows. Fix a dimension $m \geq 2$ and a bounded open connected subset $\Omega \subset \mathbb{R}^m$. Consider $d \geq 5$ vector fields $V_1, \ldots, V_d$ that depend on a parameter $z \in \mathbb{R}^l$ for some $l \in \mathbb{N}$, in addition to their dependence on the point $x \in \mathbb{R}^m$. More precisely, we assume that the components of the V_i are of the form

$$V_i^j(x) = V_i^j(x, z), \quad i = 1, \ldots, d, \quad j = 1, \ldots, m, \quad x \in \Omega, \quad z \in \mathbb{R}^l,$$

where each map $(x, z) \mapsto V_i^j(x, z)$ is real analytic in a neighborhood of $\mathrm{cl}(\Omega) \times \mathbb{R}^l$ with $\mathrm{cl}(\Omega)$ denoting the closure of Ω.[4] The vector fields are now chosen randomly by replacing the parameter z by a random vector Z in $\mathbb{R}^l$. We thus consider the randomly chosen vector fields $V_i = V_i(\,\cdot\,, Z)$, $i = 1, \ldots, d$. We can now state our main theorem.

Theorem 5.1. *Assume that*

(i) *the law of Z admits a probability density on $\mathbb{R}^l$,*

(ii) *for some $\widehat{z} \in \mathbb{R}^l$, the Lie algebra generated by the d vector fields $\widehat{V}_i = V_i(\,\cdot\,, \widehat{z})$ corresponding to $\widehat{z}$ contains all polynomial vector fields.*

Then with probability one, (1.3) and (5.1) form a universal interpolator for generic training data in the following sense. Consider a training set $\{(x_i, y_i) \in \Omega \times \Omega \colon i = 1, \ldots, N\}$ of arbitrary size, where $(x_1, \ldots, x_N)$ is drawn from an arbitrary density on $(\mathbb{R}^m)^N$ and the y_i are pairwise distinct but otherwise arbitrary. Then, with probability one, there exist controls $u_t^1, \ldots, u_t^d$ and a constant $\lambda > 0$ such that $y_i = \lambda(X_1^{x_i} - x_i)$ for all i.

Example 5.2. Fix $k \geq 2$ and $d \geq 5$. In order to specify d polynomial vector fields of degree at most k, one needs $l = dm\binom{m+k}{m}$ real coefficients. Let $\mathbb{R}^l$ be the space of all such sets of coefficients, and let $V_i(\,\cdot\,, z)$, $i = 1, \ldots, d$, be the polynomial vector fields specified by $z \in \mathbb{R}^l$. Then $(x, z) \mapsto V_i^j(x, z)$ is a polynomial, and in particular real analytic. By letting $\widehat{z} \in \mathbb{R}^l$ be the coefficients of the vector fields in Proposition 4.4, we see that condition (ii) of the theorem holds. Condition (i) holds whenever Z is drawn from an arbitrary density on $\mathbb{R}^l$.

The proof of Theorem 5.1 is presented in section 6. Ultimately it is based on the fact that any real analytic function is either identically zero, or nonzero on a set of full Lebesgue

[4] To ensure that the vector fields are globally bounded on $\mathbb{R}^m$ for each fixed z, we multiply the given real analytic functions by a compactly supported function $\varphi(x)$ that equals one on Ω. This ensures global existence and uniqueness of solutions to (1.3). The form of the vector fields outside Ω does not matter for the theorem.

measure. Condition (ii) is used to exclude the former possibility, while condition (i) is used to avoid zeros which can exist but only constitute a nullset.

The central message of Theorem 5.1 is this. The seemingly strong property of universal interpolation is not only achieved in a dimension-free manner as shown in Theorem 2.2. It is actually a *generic* property in the class of real analytic vector fields. Specifically, by drawing the vector fields randomly in the described manner, one is guaranteed with probability one that the resulting vector fields produce a universal interpolator (at least for generic training data and allowing for the additional trained readout parameter λ). Possible sampling schemes include nondegenerate normal distributions and uniform distributions on bounded open regions of the parameter space $\mathbb{R}^l$. The theorem is, however, more general than that, and we make use of this in Corollary 5.4 below.

The λ-scaling in (5.1) is reminiscent of batch normalization, especially if we were to use different parameters λ for different coordinates. Our mathematical results do not require this, however. Moreover, thanks to the normalization it is not a restriction to work with a bounded set Ω.

In practice, the CODE (1.3) is replaced by a discretization, say, with M steps. This yields a network of depth M. After randomly choosing d vector fields, the number of trainable parameters (including λ in (5.1)) becomes $Md+1$. This tends to be much smaller than the total number of parameters needed to specify the vector fields and can potentially simplify the training task significantly. The required depth M depends on the desired training error. The fact that most parameters are chosen randomly reinforces the view that randomness is a crucial ingredient for training. Investigating different sampling schemes and training algorithms in this setting is an important research question that will be treated elsewhere.

The fact that the sampling density for the vector field coefficients can be completely arbitrary leads to the following simple proof that the universal interpolator property is in a certain sense generic in the class of all smooth vector fields.

Corollary 5.3. *Fix $m \geq 2$ and a bounded open connected subset $\Omega \subset \mathbb{R}^m$. Consider $d \geq 5$ smooth vector fields $\widehat{V}_1, \dots, \widehat{V}_d$ and a tolerance $\varepsilon > 0$. Then there exist smooth vector fields $V_1, \dots, V_d$ that are uniformly ε-close to the given vector fields on Ω, in the sense that*

$$\sup_{x \in \Omega} \|V_i(x) - \widehat{V}_i(x)\| < \varepsilon, \quad i = 1, \dots, d,$$

and such that (1.3) and (5.1) form a universal interpolator for generic training data in the sense of Theorem 5.1.

Proof. By polynomial approximation, there exist polynomial vector fields $W_1, \dots, W_d$ with $\sup_{x \in \Omega} \|W_i(x) - \widehat{V}_i(x)\| < \varepsilon/2$ for all i. Let k be the largest degree among the W_i but no smaller than 2. Parameterize all polynomial vector fields of degree at most k by a coefficient vector $z \in \mathbb{R}^l$ as in Example 5.2. Let $\Theta \subset \mathbb{R}^l$ be the set of all coefficients corresponding to polynomial vector fields $V_1, \dots, V_d$ with $\sup_{x \in \Omega} \|W_i(x) - V_i(x)\| < \varepsilon/2$ for all i. Then Θ is an open set, so we can find a probability density concentrated on Θ. Thanks to Theorem 5.1 and Example 5.2, by drawing a coefficient vector Z from this density we get, with probability one, vector fields $V_1, \dots, V_d$ with the required properties. ∎

Our second corollary establishes a randomly chosen set of neural network type vector fields that satisfy the universal interpolator property.

Corollary 5.4. *Fix $m \geq 2$ and a bounded open connected subset $\Omega \subset \mathbb{R}^m$. Consider $d = 7$ vector fields of the form*

$$V_i(x) = \sigma_i(C_i x + b_i), \quad i = 1, \dots, 7,$$

where each C_i is a random matrix in $\mathbb{R}^{m\times m}$, b_i a random vector in $\mathbb{R}^m$, and $\sigma_i(\cdot)$ a real analytic nonlinearity acting componentwise, parameterized by some random vector Z_0 in a real analytic manner. Assume that for some value $\widehat{z}_0$ of Z_0, we have $\sigma_i(r) = r$ for $i = 1, 2, 3$ and $\sigma_i(r) = r^2$ for $i = 4, 5, 6, 7$. Assume also that the random elements $Z_0, C_1, \dots, C_7, b_1, \dots, b_7$ admit a joint density. Then with probability one, (1.3) and (5.1) form a universal interpolator for generic training data in the sense of Theorem 5.1.

Proof. To apply Theorem 5.1, first observe that the vector fields $V_1, \dots, V_7$ are jointly real analytic in x and in the random vector Z consisting of $Z_0, C_1, \dots, C_7, b_1, \dots, b_7$. This admits a density by assumption, so condition (i) of the theorem is satisfied. It only remains to verify condition (ii). Define the vector fields $\widehat{V}_1(x) = Ax$ and $\widehat{V}_2(x) = Bx$, where A and B are the traceless $m \times m$ matrices from Proposition 4.4. Define also the vector fields

$$\widehat{V}_3(x) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}, \quad \widehat{V}_4(x) = \begin{pmatrix} (x^m)^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \widehat{V}_5(x) = \begin{pmatrix} (x^1 + x^m)^2 \\ (x^2 + x^m)^2 \\ \vdots \\ (x^m + x^m)^2 \end{pmatrix},$$

$$\widehat{V}_6(x) = \begin{pmatrix} (x^1)^2 \\ (x^2)^2 \\ \vdots \\ (x^m)^2 \end{pmatrix}, \quad \widehat{V}_7(x) = \begin{pmatrix} (x^m)^2 \\ (x^m)^2 \\ \vdots \\ (x^m)^2 \end{pmatrix}.$$

Then the five vector fields $\widehat{V}_1, \dots, \widehat{V}_4$ and $\frac{1}{2}(\widehat{V}_5 - \widehat{V}_6 - \widehat{V}_7)$ are exactly the ones from Proposition 4.4. The Lie algebra they generate, and therefore also the Lie algebra generated by $\widehat{V}_1, \dots, \widehat{V}_7$, contains all polynomial vector fields. Let now $\widehat{z}$ be the value of Z for which $Z_0 = \widehat{z}_0$, $b_1 = b_2 = b_4 = b_5 = b_6 = b_7 = 0$, $b_3 = e_m$ (the mth canonical basis vector), $C_1 = A$, $C_2 = B$, $C_3 = 0$,

$$C_4 = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 0 & 0 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 0 \end{pmatrix}, \quad C_7 = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1 \end{pmatrix},$$

$C_6 = I$ (the $m \times m$ identity matrix), and $C_5 = C_6 + C_7$. For this value $\widehat{z}$ of Z, the vector fields $V_1, \dots, V_7$ coincide with $\widehat{V}_1, \dots, \widehat{V}_7$. Condition (ii) of Theorem 5.1 is therefore satisfied, and the proof is complete. ∎

Example 5.5. We illustrate Corollary 5.4 with one concrete example. Let all the entries of the matrices C_i and vectors b_i be standard normal. Choose a fixed real analytic nonlinearity $\sigma(\cdot)$, for example, $\sigma(r) = \arctan(r)$ or $\sigma(r) = \tanh(r)$. Define

$$\sigma_i(r) = Z_0^1 r + (1 - Z_0^1)\sigma(r)$$

for $i = 1, 2, 3$ and

$$\sigma_i(r) = Z_0^2 r^2 + (1 - Z_0^2)\sigma(r)$$

for $i = 4, 5, 6, 7$, with the two components of $Z_0 = (Z_0^1, Z_0^2)$ standard normal. All random variables are taken mutually independent. The hypotheses of the corollary are then satisfied with $\widehat{z}_0 = (1, 1)$.

6. Proof of Theorem 5.1.

We focus on the case $d = 5$. The result for larger values of d then follows by restricting to controls in the CODE (1.3) with $u_t^6 = \ldots = u_t^d = 0$. (Of course, more than five vector fields could still be important to achieve better results in practice.)

Consider therefore vector fields $V_1(\cdot, z), \ldots, V_5(\cdot, z)$ on $\mathbb{R}^m$, parameterized by a parameter $z \in \mathbb{R}^l$, such that the map $(x, z) \mapsto V_i^j(x, z)$ is real analytic in a neighborhood of $\mathrm{cl}(\Omega) \times \mathbb{R}^l$ for all $i = 1, \ldots, d$ and $j = 1, \ldots, m$. Recall from condition (ii) of the theorem that $\widehat{V}_i = V_i(\cdot, \widehat{z})$ denote the vector fields obtained by taking $z = \widehat{z}$ which, by assumption, has the property that $\mathrm{Lie}(\widehat{V}_1, \ldots, \widehat{V}_5)$ contains all polynomial vector fields.

The following lemma is the technical core of the proof of Theorem 5.1. It uses the notion of *interpolating at a tuple*, introduced in Definition 4.1.

Lemma 6.1. *Fix any $N \in \mathbb{N}$. There exists a Lebesgue nullset $\mathcal{M}_N \subset \mathbb{R}^l$ with the following property: for every $z \in \mathbb{R}^l \setminus \mathcal{M}_N$, there exists a Lebesgue nullset $\mathcal{N}_N \subset \Omega^N$ (depending on z) such that* $\mathrm{Lie}(V_1, \ldots, V_5)$ *interpolates at every tuple $\bar{x} = (x_1, \ldots, x_N) \in \Omega^N \setminus \mathcal{N}_N$.*

Proof. For each $n \in \mathbb{N}$, let $D_n = m\binom{m+n}{m}$ denote the dimension of the space of polynomial vector fields on $\mathbb{R}^m$ of degree at most n.[5] Since $\mathrm{Lie}(\widehat{V}_1, \ldots, \widehat{V}_5)$ contains all polynomial vector fields, it contains in particular a sequence of vector fields $E_1, E_2, \ldots$ such that for each n, $\{E_1, \ldots, E_{D_n}\}$ forms a basis for the space of polynomial vector fields of degree at most n. By definition of the Lie algebra, each E_j is of the form

$$E_j(x) = L_j\Big(\widehat{V}_1, \ldots, \widehat{V}_5\Big)(x)$$

for some Lie polynomial L_j on five symbols (i.e., a linear combination of Lie words built from iterated brackets).

Consider now an arbitrary $z \in \mathbb{R}^l$ and the corresponding vector fields $V_i = V_i(\cdot, z)$, $i = 1, \ldots, 5$. For each $n \in \mathbb{N}$, define the collection of vector fields

$$\mathcal{V}_n = \text{linear span of } L_1(V_1, \ldots, V_5), \ldots, L_{D_n}(V_1, \ldots, V_5).$$

[5] D_n is m times $\binom{m+n}{m}$, the dimension of the space of polynomials of degree at most n in m variables.

The collection $\mathcal{V}_n$ interpolates at a tuple $(x_1,\dots,x_N)\in\Omega^N$, in the sense of Definition 4.1, if and only if the $mN\times D_n$ matrix

$$\begin{pmatrix} L_1(V_1,\dots,V_5)(x_1) & \cdots & L_{D_n}(V_1,\dots,V_5)(x_1)\\ \vdots & & \vdots\\ L_1(V_1,\dots,V_5)(x_N) & \cdots & L_{D_n}(V_1,\dots,V_5)(x_N)\end{pmatrix}$$

has columns that span $(\mathbb{R}^m)^N$. This holds if and only if at least one $mN\times mN$ submatrix has nonzero determinant, which in turn holds if and only if the nonnegative quantity

$$\Gamma_n=\sum_{\substack{J\subseteq\{1,\dots,D_n\}\\ |J|=mN}}\det\left[\begin{pmatrix} L_j(V_1,\dots,V_5)(x_1)\\ \vdots\\ L_j(V_1,\dots,V_5)(x_N)\end{pmatrix},\ j\in J\right]^2$$

is strictly positive.

Since products, sums, and derivatives of real analytic functions remain real analytic, we have that $\Gamma_n=\Gamma_n(x_1,\dots,x_N,z)$ is jointly real analytic in $(x_1,\dots,x_N,z)$ in a neighborhood of $(\mathrm{cl}(\Omega))^N\times\mathbb{R}^l$. Furthermore, by construction, the vector fields $L_j(\widehat{V}_1,\dots,\widehat{V}_5)$, $j=1,\dots,D_n$, span all polynomial vector fields of degree at most n. Therefore, in view of Lemma 4.3, for n large enough depending on N, we have

$$\Gamma_n(x_1,\dots,x_N,\widehat{z})>0$$

for all pairwise distinct $(x_1,\dots,x_N)\in(\mathbb{R}^m)^N$. In particular,

$$(x_1,\dots,x_N,z)\mapsto\Gamma_n(x_1,\dots,x_N,z)$$

is not identically zero and thus, being a nonnegative real analytic function, is strictly positive almost everywhere. Therefore, there is a Lebesgue nullset $\mathcal{M}_N\subset\mathbb{R}^l$ such that whenever $z\in\mathbb{R}^l\setminus\mathcal{M}_N$, the real analytic function

$$(x_1,\dots,x_N)\mapsto\Gamma_n(x_1,\dots,x_N,z)$$

is not identically zero. Its zero set,

$$\mathcal{N}_N=\{(x_1,\dots,x_N)\in\Omega^N\colon\Gamma_n(x_1,\dots,x_N,z)=0\},$$

is then a Lebesgue nullset. (Note that $\mathcal{N}_N$ depends on the choice of z.) Since $\mathrm{Lie}(V_1,\dots,V_5)$ contains $\{L_j(V_1,\dots,V_5)\colon j=1,\dots,D_n\}$ and hence contains $\mathcal{V}_n$ as well, it interpolates at every tuple $\bar{x}=(x_1,\dots,x_N)\in\Omega^N\setminus\mathcal{N}_N$. The lemma is proved. ∎

We can now prove Theorem 5.1. Let $\mathcal{M}_N\subset\mathbb{R}^l$ for $N\in\mathbb{N}$ be the nullsets given in Lemma 6.1. Define

$$\mathcal{M}=\bigcup_{N=1}^{\infty}\mathcal{M}_N,$$

which is still a nullset. Assume now that $V_i = V_i(\cdot, Z)$, $i = 1, \dots, d$, are chosen randomly as described in the theorem. Then, since the law of Z has a density, $Z \in \mathbb{R}^l \setminus \mathcal{M}$ with probability one. Fix any $N \in \mathbb{N}$, and let $\mathcal{N}_N \subset \Omega^N$ be the nullset whose existence is guaranteed by Lemma 6.1. Choose $\{(x_i, y_i) \in \Omega \times \Omega \colon i = 1, \dots, N\}$ as described in the theorem. Then $\bar{x} = (x_1, \dots, x_N)$ lies in $\Omega^N \setminus \mathcal{N}_N$ with probability one, so that $\mathrm{Lie}(V_1, \dots, V_5)$ interpolates at $\bar{x}$. Lemma 4.2 now implies that the Hörmander condition holds at $\bar{x}$:

$$\mathrm{Lie}\left(V_1^{\oplus N}, \dots, V_d^{\oplus N}\right)(\bar{x}) = (\mathbb{R}^m)^N.$$

By continuity, there is an open connected neighborhood $\mathcal{U} \subset \Omega^N$ of $\bar{x}$ such that the Hörmander condition holds everywhere in $\mathcal{U}$. Moreover, since $\mathcal{U}$ is open, it is possible to choose $\lambda > 0$ large enough that $\bar{x} + \lambda^{-1}\bar{y} \in \mathcal{U}$, where $\bar{y} = (y_1, \dots, y_N)$. We can then apply the Chow–Rashevskii theorem in $\mathcal{U}$ to get controls $u_t^1, \dots, u_t^5$ that achieve $x_i + \lambda^{-1} y_i = X_1^{x_i}$ for $i = 1, \dots, N$. This completes the proof of the theorem.

Remark 6.2. We conjecture that $d = 2$ vector fields would actually be sufficient for the conclusion of Theorem 5.1. Notice also how the rescaling trick of introducing an additional parameter λ localizes the problem. This circumvents potentially very difficult questions about the global structure of the zero sets $\mathcal{N}_N$ that may prevent us from applying the Chow–Rashevskii theorem globally.

Appendix A. Generators for the polynomial vector fields. In this appendix we verify Claim 3 in the proof of Proposition 4.4. To avoid confusion with powers, we here use subscripts to denote the components of the vector $x = (x_1, \dots, x_m)$. Moreover, to make computations more transparent we canonically identify any vector field $V(x) = f_1(x)e_1 + \cdots + f_m(x)e_m$ on $\mathbb{R}^m$ with the differential operator $f_1(x)\partial_1 + \cdots + f_m(x)\partial_m$, which we again denote by V. Here $\partial_i = \frac{\partial}{\partial x_i}$ denotes partial derivative with respect to x_i. The action of V on a smooth scalar function g is $Vg = f_1\partial_1 g + \cdots + f_m\partial_m g$. The Lie bracket of two vector fields $f\partial_i$ and $g\partial_j$ is $[f\partial_i, g\partial_j] = f\partial_i g - g\partial_j f$.[6]

We now proceed with the proof. Let $\mathcal{L}$ be the Lie algebra generated by the vector fields

$$\partial_m, \quad x_m^2\partial_1, \quad x_m \sum_{i=1}^m x_i\partial_i, \quad x_i\partial_j \quad (i, j = 1, \dots, m).$$

We must show that $\mathcal{L}$ contains all polynomial vector fields.

Let us first show that $\mathcal{L}$ contains all polynomial vector fields of degree at most two. All linear vector fields lie in $\mathcal{L}$ by assumption. Furthermore, all constant vector fields lie in $\mathcal{L}$ because $\partial_m \in \mathcal{L}$ by assumption, and $\partial_i = [\partial_m, x_m\partial_i] \in \mathcal{L}$ for $i = 1, \dots, m-1$.

We now turn to the quadratic vector fields and start by considering the following identities. For $i \in \{1, \dots, m-1\}$, we compute

[6] In particular, this gives the formula $[U, V]g = U(Vg) - V(Ug)$ for every smooth function g, showing that the Lie bracket of vector fields coincides with the linear commutator of the associated differential operators.

$$
\begin{aligned}
2x_{m-1}x_m\partial_i &= [x_{m-1}\partial_m, x_m^2\partial_i],\\
x_{m-2}x_m\partial_i &= [x_{m-2}\partial_{m-1}, x_{m-1}x_m\partial_i],\\
&\vdots\\
x_{i+1}x_m\partial_i &= [x_{i+1}\partial_{i+2}, x_{i+2}x_m\partial_i],
\end{aligned}
\tag{A.1}
$$

where the last line is only included if $i \le m-2$. For $i \in \{2,\dots,m\}$ we compute

$$
\begin{aligned}
2x_1x_m\partial_i &= [x_1\partial_m, x_m^2\partial_i],\\
x_2x_m\partial_i &= [x_2\partial_1, x_1x_m\partial_i],\\
&\vdots\\
x_{i-1}x_m\partial_i &= [x_{i-1}\partial_{i-2}, x_{i-2}x_m\partial_i].
\end{aligned}
$$

Moreover, we have $x_m^2\partial_i = [x_m^2\partial_1, x_1\partial_i]$ for $i = 1,\dots,m-1$. From these computations we deduce that $\mathcal{L}$ contains all vector fields of the form $f(x)\partial_i$, where $i \in \{1,\dots,m-1\}$ and $f(x)$ ranges across the monomials listed in the following matrix:

$$
\begin{pmatrix}
x_1^2 & x_1x_2 & \cdots & x_1x_{i-1} & 0 & x_1x_{i+1} & \cdots & x_1x_m\\
 & x_2^2 & \cdots & x_2x_{i-1} & 0 & x_2x_{i+1} & \cdots & x_2x_m\\
 & & \ddots & \vdots & & \vdots & & \vdots\\
 & & & x_{i-1}^2 & 0 & x_{i-1}x_{i+1} & \cdots & x_{i-1}x_m\\
 & & & & 0 & 0 & \cdots & 0\\
 & & & & & x_{i+1}^2 & \cdots & x_{i+1}x_m\\
 & & & & & & \ddots & \vdots\\
 & & & & & & & x_m^2
\end{pmatrix}.
\tag{A.2}
$$

We now extend this to $i = m$. A calculation shows that

$$
-x_m^2\partial_m = [x_m^2\partial_1, x_1\partial_m] + 2\sum_{i=1}^{m-1}[x_i\partial_{i+1}, x_{i+1}x_m\partial_i],
$$

which therefore lies in $\mathcal{L}$. Repeating (A.1), this time with $i = m$, gives

$$
\begin{aligned}
2x_{m-1}x_m\partial_m &= [x_{m-1}\partial_m, x_m^2\partial_m],\\
x_{m-2}x_m\partial_m &= [x_{m-2}\partial_{m-1}, x_{m-1}x_m\partial_m],\\
&\vdots\\
x_1x_m\partial_m &= [x_1\partial_2, x_2x_m\partial_m].
\end{aligned}
$$

Moreover, we have $x_jx_k\partial_m = [x_j\partial_m, x_kx_m\partial_m]$ for $j,k < m$. From this we deduce that $\mathcal{L}$ additionally contains all vector fields of the form $f(x)\partial_m$, where $f(x)$ ranges across the monomials listed in (A.2) with $i = m$.

There are still monomials missing in (A.2). Consider first the case $i = m$. We have $2x_jx_m\partial_m = [x_j\partial_m, x_m^2\partial_m]$ for $j < m$, and $x_m^2\partial_m \in \mathcal{L}$ by assumption. This confirms that $f(x)\partial_m \in \mathcal{L}$ whenever $f(x)$ is a monomial of degree two. Consider instead the case $i < m$. We compute

$$x_i \sum_{j=1}^{m} x_j\partial_j = \left[x_i\partial_m, x_m \sum_{j=1}^{m} x_j\partial_j\right] + x_ix_m\partial_m,$$

$$-x_ix_m^2\partial_m = \left[x_m^2\partial_m, x_i \sum_{j=1}^{m} x_j\partial_j\right],$$

$$x_ix_m\partial_i = x_m^2\partial_m + \frac{1}{2}[[\partial_m, x_ix_m^2\partial_m], x_m\partial_i].$$

This implies that $x_ix_m\partial_i \in \mathcal{L}$ for all $i < m$. Furthermore, for all $i \neq j$ we have

$$x_i^2\partial_i = [x_i\partial_m, x_ix_m\partial_i] + x_ix_m\partial_m,$$
$$2x_ix_j\partial_i = [x_j\partial_i, x_i^2\partial_i].$$

This confirms that $f(x)\partial_i \in \mathcal{L}$ whenever $f(x)$ is a monomial of degree two and $i \in \{1, \dots, m-1\}$. In summary, we have shown that $\mathcal{L}$ contains all polynomial vector fields of degree at most two.

It remains to prove that $\mathcal{L}$ contains all higher-degree polynomial vector fields as well. This follows by induction from the following claim; note that we have already established the base case $k = 2$.

Claim. Let $k \geq 2$, and assume $\mathcal{L}$ contains all $x^{\boldsymbol{\alpha}}\partial_i$ with $|\boldsymbol{\alpha}| \leq k$. Then $\mathcal{L}$ also contains all $x^{\boldsymbol{\alpha}}\partial_i$ with $|\boldsymbol{\alpha}| = k + 1$.

To prove the claim, pick $\boldsymbol{\alpha}$ with $|\boldsymbol{\alpha}| = k + 1$. We prove that $\mathcal{L}$ contains $x^{\boldsymbol{\alpha}}\partial_1$; the vector fields $x^{\boldsymbol{\alpha}}\partial_i$ with $i = 2, \dots, m$ are treated in the same way. There are three cases. First, if $\alpha_1 = 0$, then $\alpha_i \geq 1$ for some $i \geq 2$. Thus $2x^{\boldsymbol{\alpha}}\partial_1 = [x^{\alpha-e_1}\partial_i, x_i^2\partial_1] \in \mathcal{L}$. Second, if $\alpha_1 \geq 1$ and $\alpha_1 \neq 3$, then $(3-\alpha_1)x^{\boldsymbol{\alpha}}\partial_1 = [x^{\alpha-e_1}\partial_1, x_1^2\partial_1]$, so that $x^{\boldsymbol{\alpha}}\partial_1 \in \mathcal{L}$. Third, if $\alpha_1 = 3$, we have $x^{\boldsymbol{\alpha}} = x_1^3x^{\boldsymbol{\beta}}$ with $\boldsymbol{\beta} = (0, \alpha_2, \dots, \alpha_m)$. Then $2x^{\boldsymbol{\alpha}}\partial_1 = [x_1x^{\boldsymbol{\beta}}\partial_1, [x_1^2\partial_2, x_1x_2\partial_1] + 2[x_1^2\partial_1, x_1x_2\partial_2]] \in \mathcal{L}$. This completes the proof of the claim and shows that $\mathcal{L}$ contains all polynomial vector fields.

Acknowledgments. The authors thank the Associate Editor and two anonymous referees for their valuable comments.

REFERENCES

M. Belkin, D. J. Hsu, and P. Mitra, *Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 2306–2317.

B. Chang, L. Meng, E. Haber, F. Tung, and D. Begert, *Multi-level Residual Networks from Dynamical Systems View*, e-print, arXiv:1710.10348, 2017.

T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, *Neural ordinary differential equations*, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 6571–6583.

C. Cuchiero, M. Larsson, and J. Teichmann, *Controlled Differential Equations on Convenient Spaces*, working paper, 2019.
G. Cybenko, *Approximation by superpositions of a sigmoidal function*, Math. Control Signals Systems, 2 (1989), pp. 303–314.
E. Dupont, A. Doucet, and Y. W. Teh, *Augmented Neural ODEs*, e-print, https://arxiv.org/abs/1904.01681, 2019.
W. E., *A proposal on machine learning via dynamical systems*, Commun. Math. Stat., 5 (2017), pp. 1–11.
W. E, J. Han, and Q. Li, *A mean-field optimal control formulation of deep learning*, Res. Math. Stat., 6 (2018).
W. Grathwohl, T. Q. Chen, J. Bettencourt, I. Sutekever, and D. K. Duvenaud, *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models*, e-print, arXiv:1810.01367, 2018.
K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, in Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 770–778.
K. Hornik, *Approximation capabilities of multilayer feedforward networks*, Neural Networks, 4 (1991), pp. 251–257.
M. Kuranishi, *On everywhere dense imbedding of free groups in Lie groups*, Nagoya Math. J., 2 (1951), pp. 63–71.
D. Leites and E. Poletaeva, *Defining relations for classical Lie algebras of polynomial vector fields*, Math. Scand., 81 (1998), pp. 5–19.
T. Liang and A. Rakhlin, *Just interpolate: Kernel "ridgeless" regression can generalize*, Math. Statist. 3(2020), pp. 1329–1347.
H. Liu and P. Markowich, *Selection Dynamics for Deep Neural Networks*, e-print, arXiv:1905.09076, 2019.
S. Ma, R. Bassily, and M. Belkin, *The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning*, in Proceedings of the 35th International Conference on Machine Learning, 2018.
W. Magnus, A. Karrass, and D. Solitar, *Combinatorial Group Theory*, revised edition, Dover Publications, New York, 1976.
R. Montgomery, *A Tour of Subriemannian Geometries, Their Geodesics and Applications*, Math. Surveys Monogr. 91, American Mathematical Society, Providence, RI, 2002.
A. Pinkus, *Approximation theory of the MLP model in neural networks*, Acta Numer., 8 (1999), pp. 143–195.
H. Zhang, X. Gao, J. Unterman, and T. Arodz, *Approximation Capabilities of Neural Ordinary Differential Equations*, e-print, https://arxiv.org/abs/1907.12998, 2019.