European Journal of Applied Mathematics (2024), 1–43
doi:10.1017/S0956792524000032

CAMBRIDGE UNIVERSITY PRESS

PAPER

From NeurODEs to AutoencODEs: A mean-field control framework for width-varying neural networks

Cristina Cipriani[1,2,3], Massimo Fornasier[1,2,3] and Alessandro Scagliotti[1,3]

[1]School of Computation, Information and Technology, Technical University Munich, Munich, Germany, [2]Munich Data Science Institute (MDSI), Munich, Germany and [3]Munich Center for Machine Learning (MCML), Munich, Germany
Corresponding author: Alessandro Scagliotti; Email: scag@ma.tum.de

Received: 01 September 2023; **Revised:** 05 January 2024; **Accepted:** 12 January 2024

Keywords: Machine learning; optimal control; gradient flow; minimising movement scheme; autoencoders

2020 Mathematics Subject Classification: 49M05, 49M25, 35Q93 (Primary); 68T07, 49J20 (Secondary)

Abstract

The connection between Residual Neural Networks (ResNets) and continuous-time control systems (known as NeurODEs) has led to a mathematical analysis of neural networks, which has provided interesting results of both theoretical and practical significance. However, by construction, NeurODEs have been limited to describing constant-width layers, making them unsuitable for modelling deep learning architectures with layers of variable width. In this paper, we propose a continuous-time Autoencoder, which we call AutoencODE, based on a modification of the controlled field that drives the dynamics. This adaptation enables the extension of the mean-field control framework originally devised for conventional NeurODEs. In this setting, we tackle the case of low Tikhonov regularisation, resulting in potentially non-convex cost landscapes. While the global results obtained for high Tikhonov regularisation may not hold globally, we show that many of them can be recovered in regions where the loss function is locally convex. Inspired by our theoretical findings, we develop a training method tailored to this specific type of Autoencoders with residual connections, and we validate our approach through numerical experiments conducted on various examples.

1. Introduction

In recent years, the field of artificial intelligence has witnessed remarkable progress across diverse domains, including computer vision and natural language processing. In particular, neural networks have emerged as a prominent tool, revolutionising numerous machine learning tasks. Consequently, there is an urgent demand for a robust mathematical framework to analyse their intricate characteristics. A deep neural network can be seen as map $\phi : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$, obtained as the composition of $L \gg 1$ applications $\phi = \phi_L \circ \ldots \circ \phi_1$, where, for every $n = 1, \ldots, L$, the function $\phi_n : \mathbb{R}^{d_n} \to \mathbb{R}^{d_{n+1}}$ (also referred as *the n-th layer* of the network) depends on a *trainable* parameter $\theta_n \in \mathbb{R}^{m_n}$. The crucial process of choosing the values of the parameters $\theta_1, \ldots, \theta_L$ is known as the *training of the network*. For a complete survey on the topic, we recommend the textbook [24].

Recent advancements have explored the link between dynamical systems, optimal control and deep learning, proposing a compelling perspective. In the groundbreaking work [29], it was highlighted how the problem of training very deep networks can be alleviated by the introduction of a new type of layer called 'Residual Block'. This consists in using the identity map as skip connection and after-addition activations. In other words, every layer has the following form:

$$X_{n+1} = \phi_n(X_n) = X_n + \mathscr{F}(X_n, \theta_n), \tag{1.1}$$

where X_{n+1} and X_n are, respectively, the output and the input of the n-th layer. This kind of architecture is called *Residual Neural Network* (or ResNet). It is important to observe that, in order to give sense to the sum in (1.1), in each layer, the dimension of the input should coincide with the dimension of the output. In the practice of Deep Learning, this novel kind of layer has turned out to be highly beneficial, since it is effective in avoiding the 'vanishing of the gradients' during the training [5], or the saturation of the network's accuracy [28]. Indeed, before [29], these two phenomena had limited for long time the large-scale application of deep architectures.

Despite the original arguments in support of residual blocks being based on empirical considerations, their introduction revealed nevertheless a more mathematical and rigorous bridge between residual deep networks and controlled dynamical systems. Indeed, what makes ResNets particularly intriguing is that they can be viewed as discretized versions of continuous-time dynamical systems. This dynamical approach was proposed independently in [18] and [26], and it was greatly popularised in the machine learning community under the name of NeurODEs by [13]. This connection with dynamical systems relies on reinterpreting the iteration (1.1) as a step of the forward-Euler approximation of the following dynamical system:

$$\dot{X}(t) = \mathscr{F}(X(t), \theta(t)), \tag{1.2}$$

where $t \mapsto \theta(t)$ is the map that, instant by instant, specifies the value of the parameter θ. Moreover, the training of these neural networks, typically formulated as empirical risk minimisation, can be reinterpreted as an optimal control problem. Given a labelled dataset $\{(X_0^i, Y_0^i)\}_{i=1}^N$ of size $N \geq 1$, the depth of the time-continuous neural network (1.2) is denoted by $T > 0$. Then, training this network amounts to learning the control signals $\theta \in L^2([0,T], \mathbb{R}^m)$ in such a way that the terminal output X_T^i of (1.2) is close to it corresponding label Y_0^i for all $i = 1, \dots, N$, with respect to some distortion measure $\ell(\cdot,\cdot) \in C^1$. A typical choice is $\ell(x,y) := |x-y|^2$, which is often referred to as the *squared loss function* in the machine learning literature. Therefore, it is possible to formulate the following optimal control problem

$$\inf_{\theta \in L^2([0,T];\mathbb{R}^m)} J^N(\theta) := \begin{cases} \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \ell\big(X^i(T), Y^i(T)\big) + \lambda \int_0^T |\theta(t)|^2 \, dt, \\ \text{s.t.} \begin{cases} \dot{X}^i(t) = \mathscr{F}(t, X^i(t), \theta(t)), & \dot{Y}^i(t) = 0, \\ \big(X^i(t), Y^i(t)\big)\big|_{t=0} = \big(X_0^i, Y_0^i\big), & i \in \{1, \dots, N\}, \end{cases} \end{cases}$$

where, differently from (1.2), we admit here the explicit dependence of the dynamics on the time variable. Notice that the objective function also comprises Tikhonov regularisation, tuned by the parameter λ, which plays a crucial role in the analysis of this control problem. The benefit of interpreting the training process in this manner results in the possibility of exploiting established results from the branch of mathematical control theory, to better understand this process. A key component of optimal control theory is a set of necessary conditions, known as Pontryagin Maximum Principle (PMP), that must be satisfied by any (local) minimiser θ. These conditions were introduced in [37] and have served as inspiration for the development of innovative algorithms [33] and network structures [12] within the machine learning community.

This work specifically addresses a variant of the optimal control problem presented above, in which the focus is on the case of an infinitely large dataset. This formulation gives rise to what is commonly known as a *mean-field optimal control problem*, where the term "mean-field" emphasises the description of a multiparticle system through its averaged effect. In this context, the focus is on capturing the collective behaviour of the system rather than individual particle-level dynamics, by considering the population as a whole. As a consequence, the parameter θ is shared by the entire population of input-target pairs, and the optimal control is required to depend on the initial distribution $\mu_0(x,y) \in \mathscr{P}(\mathbb{R}^d \times \mathbb{R}^d)$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

of the input-target pairs. Therefore, the optimal control problem needs to be defined over spaces of probability measures, and it is formulated as follows:

$$\inf_{\theta\in L^2([0,T];\,\mathbb{R}^m)} J(\theta) := \begin{cases} \displaystyle\int_{\mathbb{R}^{2d}} \ell(x,y)\, d\mu_T(x,y) + \lambda \int_0^T |\theta(t)|^2\, dt, \\ \text{s.t.} \begin{cases} \partial_t \mu_t(x,y) + \nabla_x \cdot (\mathscr{F}(t,x,\theta_t)\mu_t(x,y)) = 0 \quad t\in[0,T], \\ \mu_t|_{t=0}(x,y) = \mu_0(x,y), \end{cases} \end{cases}$$

This area of study has gained attention in recent years, and researchers have derived the corresponding Pontryagin Maximum Principle in various works, such as [19] and [8]. It is worth mentioning that there are other types of mean-field analyses of neural networks, such as the well-known work [36], which focuses on mean-field at the parameter level, where the number of parameters is assumed to be infinitely large. However, our approach in this work takes a different viewpoint, specifically focusing on the control perspective in the case of an infinitely large dataset.

One of the contributions of this paper is providing a more accessible derivation of the necessary conditions for optimality, such as the well-known Pontryagin Maximum Principle. Namely, we characterise the stationary points of the cost functional, and we are able to recover the PMP that was deduced in [8] under the assumption of large values of the regularisation parameter λ, and whose proof relied on an infinite-dimensional version of the Lagrange multiplier rule. This alternative perspective offers a clearer and more intuitive understanding of the PMP, making it easier to grasp and apply it in practical scenarios.

In addition, we aim at generalising the applicability of the results presented in [8] by considering a possibly non-convex regime, corresponding to small values of the parameter $\lambda > 0$. As mentioned earlier, the regularisation coefficient λ plays a crucial role in determining the nature of the cost function. Indeed, when λ is sufficiently large, the cost function is convex on the sublevel sets, and it is possible to prove the existence and uniqueness of the solution of the optimal control problem that arises from training NeurODEs. Additionally, in this highly-regularized scenario, desirable properties of the solution, such as its continuous dependence on the initial data and a bound on the generalisation capabilities of the networks, have been derived in [8].

However, in practical applications, a large regularisation parameter may cause a poor performance of the trained NeurODE on the task. In other words, in the highly-regularized case, the cost functional is unbalanced towards the L^2-penalization, at the expenses of the term that promotes that each datum X_0^i is driven as close as possible to the corresponding target Y_0^i. This motivated us to investigate the case of low Tikhonov regularisation. While we cannot globally recover the same results as in the highly-regularized regime, we find interesting results concerning local minimisers. Moreover, we also show that the (mean field) optimal control problem related to the training of the NeurODE induces a gradient flow in the space of admissible controls. The perspective of the gradient flow leads us to consider the well-known minimising movement scheme and to introduce a proximal stabilisation term to the cost function in numerical experiments. This approach effectively addresses the well-known instability issues (see [14]) that arise when solving numerically optimal control problems (or when training NeurODEs) with iterative methods based on the PMP. It is important to note that our stabilisation technique differs from previous methods, such as the one introduced in [33].

1.1. From NeurODEs to AutoencODEs

Despite their huge success, it should be noted that NeurODEs (as well as ResNets, their discrete-time counterparts) in their original form face a limitation in capturing one of the key aspects of modern machine learning architectures, namely the discrepancy in dimensionality between consecutive layers. As observed above, the use of skip connections with identity mappings requires a 'rectangular' shape of the network, where the width of the layers are all identical and constant with respect to the input's

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

dimension. This restriction poses a challenge when dealing with architectures that involve layers with varying dimensions, which are common in many state-of-the-art models. Indeed, the inclusion of layers with different widths can enhance the network's capacity to represent complex functions and to capture intricate patterns within the data. In this framework, Autoencoders have emerged as a fundamental class of models specifically designed to learn efficient representations of input data by capturing meaningful features through an encoder-decoder framework. More precisely, the encoder compresses the input data into a lower-dimensional latent space, while the decoder reconstructs the original input from the compressed representation. The concept of Autoencoders was first introduced in the 1980s in [39], and since then, it has been studied extensively in various works, such as [30], among many others. Nowadays, Autoencoders have found numerous applications, including data compression, dimensionality reduction, anomaly detection and generative modelling. Their ability to extract salient features and capture underlying patterns in an unsupervised manner makes them valuable tools in scenarios where labelled training data is limited or unavailable. Despite their huge success in practice, there is currently a lack of established theory regarding the performance guarantees of these models.

Prior works, such as [20], have extended the control-theoretic analysis of NeurODEs to more general width-varying neural networks. Their model is based on an integro-differential equation that was first suggested in [34] in order to study the continuum limit of neural networks with respect to width and depth. In such an equation the state variable has a dependency on both time and space since the changing dimension over time is viewed as an additional spatial variable. In [20, Section 6] the continuous space-time analogue of ResNets proposed in [34] has been considered and discretized in order to model variable width ResNets of various types, including convolutional neural networks. The authors assume a simple time-dependent grid, and use forward difference discretization for the time derivative and Newton-Cotes for discretizing the integral term, but refer to more sophisticated moving grids in order to possibly propose new types of architectures. In this setting, they are also able to derive some stability estimates and generalisation properties in the overparametrized regime, making use of turnpike theory in optimal control [22]. In principle, there could be several different ways to model width-varying neural networks with dynamical systems, e.g., forcing some structure on the control variables, or formulating a viability problem. In this last case, a possibility could be to require admissible trajectories to visit some lower-dimensional subsets during the evolution. For an introduction to viability theory, we recommend the monograph [4], while we refer to [7, 9] for recent results on viability theory for differential inclusions in Wasserstein spaces.

In contrast, our work proposes a simpler extension of the control-theoretical analysis. It is based on a novel design of the vector field that drives the dynamics, allowing us to develop a continuous-time model capable of accommodating various types of width-varying neural networks. This approach has the advantage of leveraging insights and results obtained from our previous work [8]. Moreover, the simplicity of our model facilitates the implementation of residual networks with variable width and allows us to test their performance in machine learning tasks. In order to capture width-varying neural networks, we need to extend the previous control-theoretical framework to a more general scenario, in particular, we need to relax some of the assumptions of [8]. This is done in Subsection 2.2, where we introduce discontinuous-in-time dynamics that can describe a wider range of neural network architectures. By doing so, we enable the study of Autoencoders (and, potentially, of other width-varying architectures) from a control-theoretic point of view, with the perspective of getting valuable insights into their behaviour.

Furthermore, we also generalise the types of activation functions that can be employed in the network. The previous work [8] primarily focused on sigmoid functions, which do not cover the full range of activations commonly employed in practice. Our objective is to allow for unbounded activation functions, which are often necessary for effectively solving certain tasks. By considering a broader set of activation functions, we aim to enhance the versatility and applicability of our model.

Furthermore, in contrast to [8], we introduce a stabilisation method to allow the numerical resolution of the optimal control problem in the low-regularized regime, as previously discussed. This stabilisation technique provides the means to test the architecture with our training approach on various tasks: from

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

low-dimensional experiments, which serve to demonstrate the effectiveness of our method, to more sophisticated and high-dimensional tasks such as image reconstruction. In Section 5, we present all the experiments and highlight noteworthy behaviours that we observe. An in-depth exploration of the underlying reasons for these behaviours is postponed to future works.

The structure of the paper is the following: Section 2 discusses the dynamical model of NeurODEs and extends it to the case of width-varying neural networks, including Autoencoders, which we refer to as AutoencODEs. In Section 3, we present our mean-field analysis, focusing on the scenario of an infinitely large dataset. We formulate the mean-field optimal control problem, we derive a set of necessary optimality conditions, and we provide a convergence result for the finite-particles approximation. At the end of this section, we compare our findings with the ones previously obtained in [8]. Section 4 covers the implementation and the description of the training procedure, and we compare it with other methods for NeurODEs existing in the literature. Finally, in Section 5, we present the results of our numerical experiments, highlighting interesting properties of the AutoencODEs that we observe.

1.2. Measure-theoretic preliminaries

Given a metric space (X, d_X), we denote by $\mathscr{M}(X)$ the space of signed Borel measures in X with finite total variation, and by $\mathscr{P}(X)$ the space of probability measures, while $\mathscr{P}_c(X) \subset \mathscr{P}(X)$ represents the set of probability measures with compact support. Furthermore, $\mathscr{P}_c^N(X) \subset \mathscr{P}_c(X)$ denotes the subset of empirical or atomic probability measures. Given $\mu \in \mathscr{P}(X)$ and $f : X \to Y$, with f $\mu-$measurable, we denote with $f_\# \mu \in \mathscr{P}(Y)$ the push-forward measure defined by $f_\#\mu(B) = \mu(f^{-1}(B))$ for any Borel set $B \subset Y$. Moreover, we recall the change-of-variables formula

$$\int_Y g(y)\, d\big(f_\#\mu\big)(y) = \int_X g \circ f(x)\, d\mu(x) \tag{1.3}$$

whenever either one of the integrals makes sense.

We now focus on the case $X = \mathbb{R}^d$ and briefly recall the definition of the Wasserstein metrics of optimal transport in the following definition, and refer to [2, Chapter 7] for more details.

Definition 1. *Let $1 \le p < \infty$ and $\mathscr{P}_p(\mathbb{R}^d)$ be the space of Borel probability measures on $\mathbb{R}^d$ with finite p-moment. In the sequel, we endow the latter with the p-Wasserstein metric*

$$W_p^p(\mu, \nu) := \inf \left\{ \int_{\mathbb{R}^{2d}} |z - \hat{z}|^p \, d\pi(z, \hat{z}) \;\middle|\; \pi \in \Pi(\mu, \nu) \right\},$$

where $\Pi(\mu, \nu)$ denotes the set of transport plan between μ and ν, that is the collection of all Borel probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ with marginals μ and ν in the first and second component respectively.

It is a well-known result in optimal transport theory that when $p = 1$ and $\mu, \nu \in \mathscr{P}_c(\mathbb{R}^d)$, then the following alternative representation holds for the Wasserstein distance

$$W_1(\mu, \nu) = \sup \left\{ \int_{\mathbb{R}^d} \varphi(x)\, d\big(\mu - \nu\big)(x) \;\middle|\; \varphi \in \mathrm{Lip}(\mathbb{R}^d),\ \mathrm{Lip}(\varphi) \le 1 \right\}, \tag{1.4}$$

by Kantorovich's duality [2, Chapter 6]. Here, $\mathrm{Lip}(\mathbb{R}^d)$ stands for the space of real-valued Lipschitz-continuous functions on $\mathbb{R}^d$, and $\mathrm{Lip}(\varphi)$ is the Lipschitz constant of a mapping φ defined ad

$$Lip(\varphi) := \sup_{x,y \in \mathbb{R}^d, x \neq y} \frac{\|\varphi(x) - \varphi(y)\|}{\|x - y\|}.$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

2. Dynamical model of NeurODEs

2.1. Notation and basic facts

In this paper, we consider controlled dynamical systems in $\mathbb{R}^d$, where the velocity field is prescribed by a function $\mathscr{F} : [0, T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ that satisfies these basic assumptions.

Assumption 1. *The vector field $\mathscr{F} : [0, T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ satisfies the following:*

(i) For every $x \in \mathbb{R}^d$ and every $\theta \in \mathbb{R}^m$, the map $t \mapsto \mathscr{F}(t, x, \theta)$ is measurable in t.

(ii) For every $R > 0$, there exists a constant $L_R > 0$ such that, for every $\theta \in \mathbb{R}^m$, it holds

$$|\mathscr{F}(t, x_1, \theta) - \mathscr{F}(t, x_2, \theta)| \leq L_R(1 + |\theta|)|x_1 - x_2|, \quad \text{for a.e. } t \in [0, T] \text{ and every } x_1, x_2 \in B_R(0),$$

from which it follows that $|\mathscr{F}(t, x, \theta)| \leq L_R(1 + |x|)(1 + |\theta|)$ for a.e. $t \in [0, T]$.

(iii) For every $R > 0$, there exists a constant $L_R > 0$ such that, for every $\theta_1, \theta_2 \in \mathbb{R}^m$, it holds

$$|\mathscr{F}(t, x, \theta_1) - \mathscr{F}(t, x, \theta_2)| \leq L_R(1 + |\theta_1| + |\theta_2|)|\theta_1 - \theta_2|, \quad \text{for a.e. } t \in [0, T] \text{ and every } x \in B_R(0).$$

The control system that we are going to study is

$$\begin{cases} \dot{x}(t) = \mathscr{F}(t, x(t), \theta(t)), & \text{a.e. in } [0, T], \\ x(0) = x_0, \end{cases} \tag{2.1}$$

where $\theta \in L^2([0, T], \mathbb{R}^m)$ is the control that drives the dynamics. Owing to Assumption 1, the classical Carathéodory Theorem (see [27, Theorem 5.3]) guarantees that, for every $\theta \in L^2([0, T], \mathbb{R}^m)$ and for every $x_0 \in \mathbb{R}^d$, the Cauchy problem (2.1) has a unique solution $x : [0, T] \to \mathbb{R}^d$. Hence, for every $(t, \theta) \in [0, T] \times L^2([0, T], \mathbb{R}^m)$, we introduce the flow map $\Phi^\theta_{(0,t)} : \mathbb{R}^d \to \mathbb{R}^d$ defined as

$$\Phi^\theta_{(0,t)}(x_0) := x(t), \tag{2.2}$$

where $t \mapsto x(t)$ is the absolutely continuous curve that solves (2.1), with Cauchy datum $x(0) = x_0$ and corresponding to the admissible control $t \mapsto \theta(t)$. Similarly, given $0 \leq s < t \leq T$, we write $\Phi^\theta_{(s,t)} : \mathbb{R}^d \to \mathbb{R}^d$ to denote the flow map obtained by prescribing the Cauchy datum at the more general instant $s \geq 0$. We now present the properties of the flow map defined in (2.2) that describes the evolution of the system: we show that is well-posed, and we report some classical properties.

Proposition 2.1. *For every $t \in [0, T]$ and for every $\theta \in L^2([0, T], \mathbb{R}^m)$, let $\mathscr{F}$ satisfy Assumption 1. Then, the flow $\Phi^\theta_{(0,t)} : \mathbb{R}^d \to \mathbb{R}^d$ is well-defined for any $x_0 \in \mathbb{R}^d$ and it satisfies the following properties.*

- *For every $R > 0$ and $\rho > 0$, there exists a constant $\bar{R} > 0$ such that*
$$\left|\Phi^\theta_{(0,t)}(x)\right| \leq \bar{R}$$
for every $x \in B_R(0)$ and every $\theta \in L^2([0, T], \mathbb{R}^m)$ such that $||\theta||_{L^2} \leq \rho$.
- *For every $R > 0$ and $\rho > 0$, there exists a constant $\bar{L} > 0$ such that, for every $t \in [0, T]$, it holds*
$$\left|\Phi^\theta_{(0,t)}(x_1) - \Phi^\theta_{(0,t)}(x_2)\right| \leq \bar{L}|x_1 - x_2|$$
for every $x_1, x_2 \in B_R(0)$ and every $\theta \in L^2([0, T], \mathbb{R}^m)$ such that $||\theta||_{L^2} \leq \rho$.
- *For every $R > 0$ and $\rho > 0$, there exists a constant $\bar{L} > 0$ such that, for every $t_1, t_2 \in [0, T]$, it holds*
$$\left|\Phi^\theta_{(0,t_2)}(x) - \Phi^\theta_{(0,t_1)}(x)\right| \leq \bar{L}|t_2 - t_1|^{\frac{1}{2}}$$
for every $x \in B_R(0)$ and every $\theta \in L^2([0, T], \mathbb{R}^m)$ such that $||\theta||_{L^2} \leq \rho$.
- *For every $R > 0$ and $\rho > 0$, there exists a constant $\bar{L} > 0$ such that, for every $t \in [0, T]$, it holds*
$$\left|\Phi^{\theta_1}_{(0,t)}(x) - \Phi^{\theta_2}_{(0,t)}(x)\right|_2 \leq \bar{L}\|\theta_1 - \theta_2\|_{L^2}$$
for every $x \in B_R(0)$ and every $\theta_1, \theta_2 \in L^2([0, T], \mathbb{R}^m)$ such that $\|\theta_1\|_{L^2}, \|\theta_2\|_{L^2} \leq \rho$.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

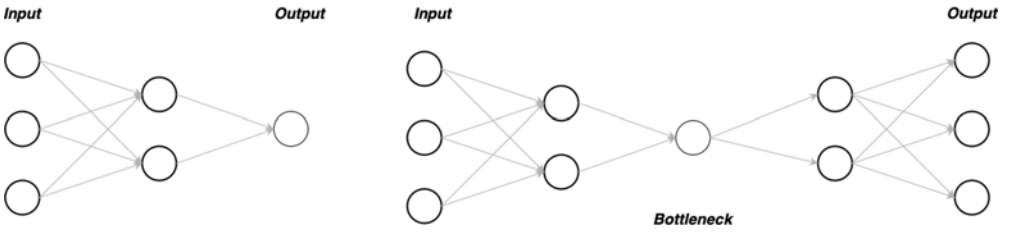

Figure 1. *Left: network with an encoder structure. Right: autoencoder.*

Proof. The proof is postponed to the Appendix (see Lemmata A.1, A.2, A.3, A.4). □

Even though the framework introduced in Assumption 1 is rather general, in this paper we specifically have in mind the case where the mapping $\mathscr{F} : [0, T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ represents the feed-forward dynamics associated to ResNets. In this scenario, the parameter $\theta \in \mathbb{R}^m$ encodes the *weights* and *shifts* of the network, i.e., $\theta = (W, b)$, where $W \in \mathbb{R}^{d\times d}$ and $b \in \mathbb{R}^d$. Moreover, the mapping $\mathscr{F}$ has the form:

$$\mathscr{F}(t, x, \theta) = \sigma(Wx + b),$$

where $\sigma : \mathbb{R}^d \to \mathbb{R}^d$ is a non-linear function acting component-wise, often called in literature *activation function*. In this work, we consider sigmoidal-type activation functions, such as the hyperbolic tangent function:

$$\sigma(x) = \tanh(x),$$

as well as smooth approximations of the Rectified Linear Unit (ReLU) function, which is defined as:

$$\sigma(x) = \max\{0, x\}. \tag{2.3}$$

We emphasise the need to consider smoothed versions of the ReLU function due to additional differentiability requirements on $\mathscr{F}$, which will be further clarified in Assumption 2. Another useful activation function covered by Assumption 2 is the Leaky ReLU function:

$$\sigma(x) = \max\{0, x\} - \max\{-\alpha x, 0\} \tag{2.4}$$

where $\alpha \in [0, 1]$ is a predetermined parameter that allows the output of the function to have negative values. The smooth approximations of (2.3) and (2.4) that we consider will be presented in Section 4.

2.2. From NeurODEs to AutoencODEs

As explained in the Introduction, NeurODEs and ResNets –their discrete-time counterparts– face the limitation of a 'rectangular' shape of the network because of formulas (1.2) and (1.1), respectively. To overcome this fact, we aim at designing a continuous-time model capable of describing width-varying neural networks, with a particular focus on Autoencoders, as they represent the prototype of neural networks whose layers operate between spaces of different dimensions. Indeed, Autoencoders consist of an *encoding phase*, where the layers' dimensions progressively decrease until reaching the 'latent dimension' of the network. Subsequently, in the *decoding phase*, the layers' widths are increased until the same dimensionality as the input data is restored. For this reason, Autoencoders are prominent examples of width-varying neural networks, since the changes in layers' dimensions lie at the core of their functioning. Sketches of encoders and Autoencoders are presented in Figure 1. Finally, we insist on the fact that our model can encompass as well other types of architectures. In this regard, in Remark 2.2 we discuss how our approach can be extended to U-nets.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

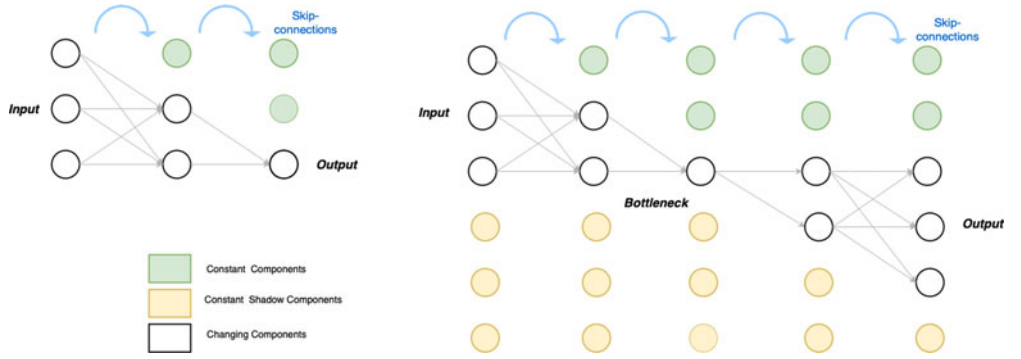

Figure 2. *Left: embedding of an encoder into a dynamical system. Right: model for an autoencoder.*

Encoder.

Our goal is to first model the case of a network, which sequentially reduces the dimensionality of the layers' outputs. For this purpose, we artificially force some of the components not to evolve anymore, while we let the others be active part of the dynamics. More precisely, given an input variable $x_0 \in \mathbb{R}^d$, we denote with $(\mathscr{I}_j)_{j=0,\dots,r}$ an increasing filtration, where each element $\mathscr{I}_j$ contains the sets of indices whose corresponding components are *inactive*, i.e., they are constant and do not contribute to the dynamics. Clearly, since the layers' width will decrease sequentially, the filtration of inactive components $\mathscr{I}_j$ will increase, i.e.

$$\emptyset =: \mathscr{I}_0 \subsetneq \mathscr{I}_1 \subsetneq \ldots \subsetneq \mathscr{I}_r \subsetneq \{1,\dots,d\}, \quad r < d, \quad j = 0,\dots,r.$$

On the other hand, the sets of indices of *active* components define a decreasing filtration $\mathscr{A}_j := \{1,\dots,d\} \setminus \mathscr{I}_j$ for $j = 0,\dots,r$. As opposed to before, the sets of active components $(\mathscr{A}_j)_{j=0,\dots,r}$ satisfy

$$\{1,\dots,d\} =: \mathscr{A}_0 \supsetneq \mathscr{A}_1 \supsetneq \ldots \supsetneq \mathscr{A}_r \supsetneq \emptyset, \quad r < d, \quad j = 0,\dots,r.$$

We observe that, for every $j = 0,\dots,r$, the sets $\mathscr{A}_j$ and $\mathscr{I}_j$ provide a partition of $\{1,\dots,d\}$. A visual representation of this model for encoders is presented on the left side of Figure 2.

Now, in the time interval $[0,T]$, let us consider $r+1$ nodes $0 = t_0 < t_1 < \ldots < t_r < t_{r+1} = T$. For $j = 0,\dots,r$, we denote with $[t_j, t_{j+1}]$ the sub-interval and, for every $x \in \mathbb{R}^d$, we use the notation $x_{\mathscr{I}_j} := (x_i)_{i\in\mathscr{I}_j}$ and $x_{\mathscr{A}_j} := (x_i)_{i\in\mathscr{A}_j}$ to access the components of x belonging to $\mathscr{I}_j$ and $\mathscr{A}_j$, respectively. Hence, the controlled dynamics for any $t \in [t_j, t_{j+1}]$ can be described by

$$\begin{cases} \dot{x}_{\mathscr{I}_j}(t) = 0, \\ \dot{x}_{\mathscr{A}_j}(t) = \mathscr{G}_j(x_{\mathscr{A}_j}(t), \theta(t)), \end{cases} \tag{2.5}$$

where $\mathscr{G}_j : \mathbb{R}^{|\mathscr{A}_j|} \times \mathbb{R}^m \to \mathbb{R}^{|\mathscr{A}_j|}$, for $j = 0,\dots,r$ and $x(0) = x_{\mathscr{A}_0}(0) = x_0$. Furthermore, the dynamical system describing the encoding part is

$$\begin{cases} \dot{x}(t) = \mathscr{F}(t, x(t), \theta(t)), \quad \text{a.e } t \in [0,T], \\ x(0) = x_0 \end{cases}$$

where, for $t \in [t_j, t_{j+1}]$, we define the discontinuous vector field as follows

$$\left(\mathscr{F}(t,x,\theta)\right)_k = \begin{cases} \left(\mathscr{G}\left(x_{\mathscr{A}_j}, \theta\right)\right)_k, & \text{if } k \in \mathscr{A}_j, \\ 0, & \text{if } k \in \mathscr{I}_j. \end{cases} \tag{2.6}$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Remark 2.1. Notice that $\theta(t) \in \mathbb{R}^m$ for every $t \in [0, T]$, according to the model that we have just described. However, it is natural to expect that, since x has varying active components, in a similar way the controlled dynamics $\mathscr{F}(t, x, \theta)$ shall not explicitly depend at every $t \in [0, T]$ on every component of θ.

Autoencoder.

We now extend the previous model to the case of networks, which not only decrease the dimensionality of the layers but are also able to increase the layers' width in order to restore the original dimension of the input data. Here we denote by $z_0 \in \mathbb{R}^{\tilde{d}}$ the input variable, and we fictitiously augment the input's dimension, so that we consider the initial datum $x_0 = (z_0, \underline{0}) \in \mathbb{R}^d = \mathbb{R}^{\tilde{d}} \times \mathbb{R}^{\tilde{d}}$, where $\underline{0} \in \mathbb{R}^{\tilde{d}}$. We make use of the following notation for every $x \in \mathbb{R}^d$:

$$x = \left((z_i)_{i=1,\dots,\tilde{d}}, \left(z_i^H\right)_{i=1,\dots,\tilde{d}}\right)$$

where z^H is the augmented (or *shadow*) part of the vector x. In this model, the time horizon $[0, T]$ is splitted using the following time-nodes:

$$0 = t_0 \le t_1 \le \dots \le t_r \le \dots \le t_{2r} \le t_{2r+1} := T$$

where t_r, which was the end of the encoder in the previous model, is now the instant corresponding to the bottleneck of the autoencoder. Similarly, as before, we introduce two families of partitions of $\{1, \dots, \tilde{d}\}$ modelling the active and non-active components of, respectively, z and z^H. The first filtrations are relative to the encoding phase and they involve the component of z:

$$\begin{cases} \mathscr{I}_{j-1} \subsetneq \mathscr{I}_j & \text{if } 1 \le j \le r, \\ \mathscr{I}_j = \mathscr{I}_{j-1} & \text{if } j > r, \end{cases} \qquad \begin{cases} \mathscr{A}_{j-1} \supsetneq \mathscr{A}_j & \text{if } 1 \le j \le r, \\ \mathscr{A}_j = \mathscr{A}_{j-1} & \text{if } j > r. \end{cases}$$

where $\mathscr{I}_0 := \emptyset$, $\mathscr{I}_r \subsetneq \{1, \dots, \tilde{d}\}$ and $\mathscr{A}_0 = \{1, \dots, \tilde{d}\}$, $\mathscr{A}_r \supsetneq \emptyset$. The second filtrations, that aim at modelling the decoder, act on the shadow part of x, i.e., they involve the components of z^H:

$$\begin{cases} \mathscr{I}_{j-1}^H = \{1, \dots, d\} & \text{if } 1 \le j \le r, \\ \mathscr{I}_j^H \subsetneq \mathscr{I}_{j-1}^H & \text{if } r < j \le 2r, \end{cases} \qquad \begin{cases} \mathscr{A}_{j-1}^H = \emptyset & \text{if } 1 \le j \le r, \\ \mathscr{A}_j^H \supsetneq \mathscr{A}_{j-1}^H & \text{if } r < j \le 2r. \end{cases}$$

While the encoder structure acting on the input data z_0 is the same as before, in the decoding phase we aim at activating the components that have been previously turned off during the encoding. However, since the information contained in the original input z_0 should be first compressed and then decompressed, we should not make use of the values of the components that we have turned off in the encoding and hence, we cannot re-activate them. Therefore, in our model the dimension is restored by activating components of z^H, the shadow part of x, which we recall was initialised equal to $\underline{0} \in \mathbb{R}^{\tilde{d}}$. This is the reason why we introduce sets of active and inactive components also for the shadow part of the state variable. A sketch of this type of model is presented on the right of Figure 2. Moreover, in order to be consistent with the classical structure of an autoencoder, the following identities must be satisfied:

1. $\mathscr{A}_j \cap \mathscr{A}_j^H = \emptyset$ for every $j = 1, \dots, 2r$,
2. $\mathscr{A}_{2r} \cup \mathscr{A}_{2r}^H = \{1, \dots, \tilde{d}\}$.

The first identity formalises the constraint that the active component of z and those of z^H cannot overlap and must be distinct, while the second identity imposes that, at the end of the evolution, the active components in z and z^H should sum up exactly to $1, \dots, \tilde{d}$. Furthermore, from the first identity we derive that $\mathscr{A}_j \subseteq (\mathscr{A}_j^H)^C = \mathscr{I}_j^H$ and, similarly, $\mathscr{A}_j^H \subseteq \mathscr{I}_j$ for every $j = 1, \dots, 2r$. Moreover, $\mathscr{A}_r$ satisfies the inclusion $\mathscr{A}_r \subseteq \mathscr{A}_j$ for every $j = 1, \dots, 2r$, which is consistent with the fact that layer with the smallest width is located in the bottleneck, i.e., in the interval $[t_r, t_{r+1}]$. Finally, from the first and the second assumption, we obtain that $\mathscr{A}_{2r}^H = \mathscr{I}_{2r}$, i.e., the final active components of z^H coincide with the inactive

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

components of z, and, similarly, $\mathscr{I}_{2r}^H = \mathscr{A}_{2r}$. Finally, to access the active components of $x = (z, z^H)$, we make use of the following notation:

$$x_{\mathscr{A}_j} = (z_k)_{k\in\mathscr{A}_j}, \quad x_{\mathscr{A}_j^H} = \left(z_k^H\right)_{k\in\mathscr{A}_j^H} \quad \text{and} \quad x_{\mathscr{A}_j,\mathscr{A}_j^H} = \left(z_{\mathscr{A}_j}, z^H_{\mathscr{A}_j^H}\right),$$

and we do the same for the inactive components:

$$x_{\mathscr{I}_j} = (z_k)_{k\in\mathscr{I}_j}, \quad x_{\mathscr{I}_j^H} = \left(z_k^H\right)_{k\in\mathscr{I}_j^H} \quad \text{and} \quad x_{\mathscr{I}_j,\mathscr{I}_j^H} = \left(z_{\mathscr{I}_j}, z^H_{\mathscr{I}_j^H}\right).$$

We are now in position to write the controlled dynamics in the interval $t_j \le t \le t_{j+1}$:

$$\begin{cases} \dot{x}_{\mathscr{I}_j,\mathscr{I}_j^H}(t) = 0, \\ \dot{x}_{\mathscr{A}_j,\mathscr{A}_j^H}(t) = \mathscr{G}_j(x_{\mathscr{A}_j,\mathscr{A}_j^H}(t), \theta(t)), \end{cases} \tag{2.7}$$

where $\mathscr{G}_j : \mathbb{R}^{|\mathscr{A}_j|+|\mathscr{A}_j^H|} \times \mathbb{R}^m \to \mathbb{R}^{|\mathscr{A}_j|+|\mathscr{A}_j^H|}$, for $j = 0, \dots, 2r$, and $x^H_{\mathscr{I}_0}(0) = \underline{0}$, $x_{\mathscr{A}_0}(0) = x_0$. As before, we define the discontinuous vector field $\mathscr{F}$ for $t \in [t_j, t_{j+1}]$, as follows

$$\left(\mathscr{F}(t, x, \theta)\right)_k = \begin{cases} \left(\mathscr{G}\left(x_{\mathscr{A}_j,\mathscr{A}_j^H}, \theta\right)\right)_k, & \text{if } k \in \mathscr{A}_j \cup \mathscr{A}_j^H \\ 0, & \text{if } k \in \mathscr{I}_j \cup \mathscr{I}_j^N. \end{cases} \tag{2.8}$$

Hence, we are now able to describe any type of width-varying neural network through a continuous-time model depicted by the following dynamical system

$$\begin{cases} \dot{x}(t) = \mathscr{F}(t, x(t), \theta(t)) \quad \text{a.e. in } [0, T], \\ x(0) = x_0. \end{cases}$$

It is essential to highlight the key difference between the previous NeurODE model in (2.6) and the current model: the vector field $\mathscr{F}$ now explicitly depends on the time variable t to account for sudden dimensionality drops, where certain components are forced to remain constant. As a matter of fact, the resulting dynamics exhibit high discontinuity in the variable t. To the best of our knowledge, this is the first attempt to consider such discontinuous dynamics in NeurODEs. Previous works, such as [18, 26], typically do not include an explicit dependence on the time variable in the right-hand side of NeurODEs, or they assume a continuous dependency on time, as in [8]. Furthermore, it is worth noting that the vector field $\mathscr{F}$ introduced to model autoencoders satisfies the general assumptions outlined in Assumption 1 at the beginning of this section.

Remark 2.2. The presented model, initially designed for Autoencoders, can be easily extended to accommodate various types of width-varying neural networks, including architectures with long skip connections such as U-nets [38]. While the specific details of U-nets are not discussed in detail, their general structure is outlined in Figure 3. U-nets consist of two main components: the contracting path (encoder) and the expansive path (decoder). These paths are symmetric, with skip connections between corresponding layers in each part. Within each path, the input passes through a series of convolutional layers, followed by a non-linear activation function (often ReLU), and other operations (e.g., max pooling) which are not encompassed by our model. The long skip connections that characterise U-nets require some modifications to the model of autoencoder described above. If we denote with $\tilde{d}_i$ for $i = 0, \dots, r$ the dimensionality of each layer in the contracting path, we have that $\tilde{d}_{2r-i} = \tilde{d}_i$ for every $i = 0, \dots, r$. Then, given an initial condition $z_0 \in \mathbb{R}^{\tilde{d}_0}$, we embed it into the augmented state variable

$$x_0 = (z_0, \underline{0}), \text{ where } \underline{0} \in \mathbb{R}^{\tilde{d}_1+\dots+\tilde{d}_r}.$$

As done in the previous model for autoencoder, we consider time-nodes $0 = t_0 < \dots < t_{2r} = T$, and in each sub-interval we introduce a controlled dynamics with the scheme of active/inactive components depicted in Figure 3.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

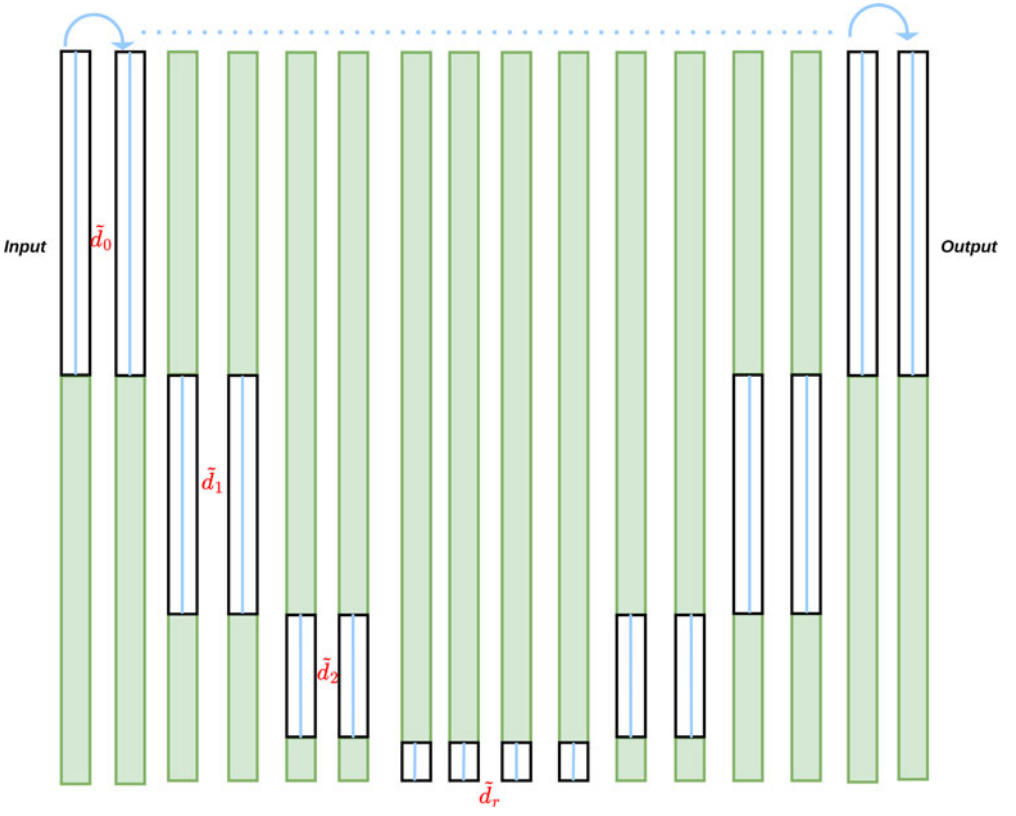

Figure 3. *Embedding of the U-net into a higher-dimensional dynamical system.*

3. Mean-field analysis

In this section, we extend the dynamical model introduced in Section 2 to its mean-field limit, which corresponds to the scenario of an infinitely large dataset. Within this framework, we formulate the training of NeurODEs and AutoencODEs as a mean-field optimal control problem and provide the associated necessary optimality conditions. It is worth noting that our analysis covers both the high-regularized regime, as studied in previous work [8], as well as the low-regularized regime, which has not been extensively addressed before. In this regard, we dedicate a subsection to a detailed comparison with the results obtained in [8]. Additionally, we investigate the case of finite-particles approximation and we establish a quantitative bound on the generalisation capabilities of these networks.

3.1. Mean-field dynamical model

In this section, we employ the same viewpoint as in [8], and we consider the case of a dataset with an infinite number of observations. In our framework, each datum is modelled as a point $x_0 \in \mathbb{R}^d$, and it comes associated to its corresponding label $y_0 \in \mathbb{R}^d$. Notice that, in principle, in Machine Learning applications the label (or *target*) datum y_0 may have dimension different from d. However, the labels' dimension is just a matter of notation and does not represent a limit of our model. Following [8], we consider the curve $t \mapsto (x(t), y(t))$, which satisfies

$$\dot{x}(t) = \mathscr{F}(t, x(t), \theta(t)) \qquad \text{and} \qquad \dot{y}(t) = 0 \tag{3.1}$$

for a.e. $t \in [0, T]$, and $(x(0), y(0)) = (x_0, y_0)$. We observe that the variable y corresponding to the labels is not changing, nor it is affecting the evolution of the variable x. We recall that the flow associated with

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

the dynamics of the variable x is denoted by $\Phi^{\theta}_{(0,t)} : \mathbb{R}^d \to \mathbb{R}^d$ for every $t \in [0,T]$, and it has been defined in (2.2). Moreover, in regards to the full dynamics prescribed by (3.1), for every admissible control $\theta \in L^2([0,T],\mathbb{R}^m)$ we introduce the extended flow $\mathbf{\Phi}^{\theta}_{(0,t)} : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d \times \mathbb{R}^d$, which reads

$$\mathbf{\Phi}^{\theta}_{(0,t)}(x_0, y_0) = \left(\Phi^{\theta}_{(0,t)}(x_0), y_0\right) \tag{3.2}$$

for every $t \in [0,T]$ and for every $(x_0, y_0) \in \mathbb{R}^d \times \mathbb{R}^d$. We now consider the case of an infinite number of labelled data $\left(X^i_0, Y^i_0\right)_{i\in I}$, where I is an infinite set of indexes. In our mathematical model, we understand this data distribution as a compactly-supported probability measure $\mu_0 \in \mathscr{P}_c\left(\mathbb{R}^d \times \mathbb{R}^d\right)$. Moreover, for every $t \in [0,T]$, we denote by $t \mapsto \mu_t$ the curve of probability measures in $\mathscr{P}_c\left(\mathbb{R}^d \times \mathbb{R}^d\right)$ that models the evolution of the solutions of (3.1) corresponding to the Cauchy initial conditions $\left(X^i_0, Y^i_0\right)_{i\in I}$. In other words, the curve $t \mapsto \mu_t$ satisfies the following continuity equation:

$$\partial_t \mu_t(x,y) + \nabla_x \cdot \left(\mathscr{F}(t,x,\theta_t)\mu_t(x,y)\right) = 0, \qquad \mu_t|_{t=0}(x,y) = \mu_0(x,y), \tag{3.3}$$

understood in the sense of distributions, i.e.

Definition 2. *For any given $T > 0$ and $\theta \in L^2([0,T],\mathbb{R}^m)$, we say that $\mu \in \mathscr{C}\left([0,T], \mathscr{P}_c\left(\mathbb{R}^{2d}\right)\right)$ is a weak solution of (3.3) on the time interval $[0,T]$ if*

$$\int_0^T \int_{\mathbb{R}^{2d}} \left(\partial_t \psi(t,x,y) + \nabla_x \psi(t,x,y) \cdot \mathscr{F}(t,x,\theta_t)\right) d\mu_t(x,y)\, dt = 0, \tag{3.4}$$

for every test function $\psi \in \mathscr{C}^1_c\left((0,T) \times \mathbb{R}^{2d}\right)$.

Let us now discuss the existence and the characterisation of the solution.

Proposition 3.1. *Under Assumptions 1, for every $\mu_0 \in \mathscr{P}_c(\mathbb{R}^{2d})$ we have that (3.3) admits a unique solution $t \mapsto \mu_t$ in the sense of Definition 2. Moreover, we have that for every $t \in [0,T]$*

$$\mu_t = \mathbf{\Phi}^{\theta}_{(0,t)\#}\mu_0. \tag{3.5}$$

Proof. Existence and uniqueness of the measure solution of (3.3) follow from [1, Proposition 2.1, Theorem 3.1 and Remark 2.1]. □

From the characterisation of the solution of (3.3) provided in (3.5), it follows that the curve $t \mapsto \mu_t$ inherits the properties of the flow map Φ^{θ} described in Proposition 2.1. These facts are collected in the next result.

Proposition 3.2. *Let us fix $T > 0$ and $\mu_0 \in \mathscr{P}_c(\mathbb{R}^{2d})$, and let us consider $\mathscr{F} : [0,T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ satisfying Assumption 1. Let $\theta \in L^2([0,T],\mathbb{R}^m)$ be an admissible control, and let $t \mapsto \mu_t$ be the corresponding solution of (3.3). Then, the curve $t \mapsto \mu_t$ satisfies the properties listed below.*

- *For every $R > 0$ and $\rho > 0$, there exists $\bar{R} > 0$ such that, for every $t \in [0,T]$, it holds that*

  $$supp(\mu_t) \subset B_{\bar{R}}(0)$$

  for every $\theta \in L^2([0,T],\mathbb{R}^m)$ such that $\|\theta\|_{L^2} \le \rho$, and for every μ_0 such that $supp(\mu_0) \subset B_R(0)$.
- *For every $R > 0$ and $\rho > 0$, there exists $\bar{L} > 0$ such that, for every $t \in [0,T]$, it holds that*

  $$W_1(\mu_t, \nu_t) \le \bar{L} W_1(\mu_0, \nu_0)$$

  for every $\theta \in L^2([0,T],\mathbb{R}^m)$ such that $\|\theta\|_{L^2} \le \rho$, and for every initial conditions μ_0, ν_0 such that the supports satisfy $supp(\mu_0), supp(\nu_0) \subset B_R(0)$, where $\mu_t = \mathbf{\Phi}^{\theta}_{(0,t)\#}\mu_0$ and $\nu_t = \mathbf{\Phi}^{\theta}_{(0,t)\#}\nu_0$.
- *For every $R > 0$ and $\rho > 0$, there exists $\bar{L} > 0$ such that, for every $t_1, t_2 \in [0,T]$, it holds that*

  $$W_1(\mu_{t_1}, \mu_{t_2}) \le \bar{L} \cdot |t_1 - t_2|^{\frac{1}{2}}$$

  for every $\theta \in L^2([0,T],\mathbb{R}^m)$ such that $\|\theta\|_{L^2} \le \rho$, and for every μ_0 such that $supp(\mu_0) \subset B_R(0)$.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

- *For every $R > 0$ and $\rho > 0$, there exists $\bar{L} > 0$ such that, for every $t \in [0, T]$, it holds that*

$$W_1(\mu_t, \nu_t) \leq \bar{L}\|\theta_1 - \theta_2\|_{L^2}$$

for every $\theta_1, \theta_2 \in L^2([0, T], \mathbb{R}^m)$ such that $\|\theta\|_{L^2}, \|\theta_2\|_{L^2} \leq \rho$, and for every initial condition μ_0 such that $supp(\mu_0) \subset B_R(0)$, where $\mu_t = \mathbf{\Phi}^{\theta_1}_{(0,t)\#}\mu_0$ and $\nu_t = \mathbf{\Phi}^{\theta_2}_{(0,t)\#}\mu_0$.

Proof. All the results follow from Proposition 3.1 and from the properties of the flow map presented in Proposition 2.1, combined with the Kantorovich duality (1.4) for the distance W_1, and the change-of-variables formula (1.3). Since the argument is essentially the same for all the properties, we detail the computations only for the second point, i.e., the Lipschitz-continuous dependence on the initial distribution. Owing to (1.4), for any $t \in [0, T]$, for any $\varphi \in \mathrm{Lip}(\mathbb{R}^{2d})$ such that its Lipschitz constant $Lip(\varphi) \leq 1$, it holds that

$$W_1(\mu_t, \nu_t) \leq \int_{\mathbb{R}^{2d}} \varphi(x, y)\, d(\mu_t - \nu_t)(x, y) = \int_{\mathbb{R}^{2d}} \varphi\left(\Phi^{\theta}_{(0,t)}(x), y\right)\, d(\mu_0 - \nu_0)(x, y) \leq \bar{L} W_1(\mu_0, \nu_0),$$

where the equality follows from the definition of push-forward and from (3.2), while the constant $\bar{L}$ in the second inequality descends from the local Lipschitz estimate of $\Phi^{\theta}_{(0,t)}$ established in Proposition 2.1. □

3.2. Mean-field optimal control

Using the transport equation (3.3), we can now formulate the mean-field optimal control problem that we aim to address. To this end, we introduce the functional $J : L^2([0, T], \mathbb{R}^m) \to \mathbb{R}$, defined as follows:

$$J(\theta) = \begin{cases} \displaystyle\int_{\mathbb{R}^{2d}} \ell(x, y)\, d\mu_T(x, y) + \lambda \int_0^T |\theta(t)|^2\, dt, \\ \text{s.t.} \begin{cases} \partial_t \mu_t(x, y) + \nabla_x \cdot (\mathscr{F}(t, x, \theta_t)\mu_t(x, y)) = 0 \quad t \in [0, T], \\ \mu_t|_{t=0}(x, y) = \mu_0(x, y), \end{cases} \end{cases} \tag{3.6}$$

for every admissible control $\theta \in L^2([0, T], \mathbb{R}^m)$. The objective is to find the optimal control θ^* that minimises $J(\theta^*)$, subject to the PDE constraint (3.3) being satisfied by the curve $t \mapsto \mu_t$. The term 'mean-field' emphasises that θ is shared by an entire population of input-target pairs, and the optimal control must depend on the distribution of the initial data. We observe that when the initial measure μ_0 is empirical, i.e.

$$\mu_0 := \mu_0^N = \frac{1}{N} \sum_{i=1}^{N} \delta_{\left(x_0^i, y_0^i\right)},$$

then minimisation of (3.6) reduces to a classical finite particle optimal control problem with ODE constraints.

We now state the further regularity hypotheses that we require, in addition to the one contained in Assumption 1.

Assumption 2. *For any given $T > 0$, the vector field $\mathscr{F}$ satisfies the following.*

(iv) For every $R > 0$ there exists a constant $L_R > 0$ such that, for every $x_1, x_2 \in B_R(0)$, it holds

$$|\nabla_x \mathscr{F}(t, x_1, \theta) - \nabla_x \mathscr{F}(t, x_2, \theta)| \leq L_R(1 + |\theta|^2)|x_1 - x_2|, \quad \text{for a.e. } t \in [0, T] \text{ and every } \theta \in \mathbb{R}^m.$$

(v) There exists another constant $L_R > 0$ such that, for every $\theta_1, \theta_2 \in \mathbb{R}^m$, it holds

$$|\nabla_\theta \mathscr{F}(t, x, \theta_1) - \nabla_\theta \mathscr{F}(t, x, \theta_2)| \leq L_R|\theta_1 - \theta_2|, \quad \text{for a.e. } t \in [0, T] \text{ and } x \in B_R(0).$$

From this, it follows that $|\nabla_\theta \mathscr{F}(t, x, \theta)| \leq L_R(1 + |\theta|)$ for every $x \in B_R(0)$ and for every $\theta \in \mathbb{R}^m$.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

(vi) There exists another constant $L_R > 0$ such that, for every $\theta_1, \theta_2 \in \mathbb{R}^m$, it holds

$$|\nabla_x \mathscr{F}(t,x,\theta_1) - \nabla_x \mathscr{F}(t,x,\theta_2)| \le L_R(1+|\theta_1|+|\theta_2|)|\theta_1-\theta_2|, \quad \text{for a.e. } t\in[0,T] \text{ and } x \in B_R(0).$$

From this, it follows that $|\nabla_x \mathscr{F}(t,x,\theta)| \le L_R(1+|\theta|^2)$ for every $x \in B_R(0)$ and for every $\theta \in \mathbb{R}^m$.

(vii) There exists another constant $L_R > 0$ such that

$$|\nabla_\theta \mathscr{F}(t,x_1,\theta) - \nabla_\theta \mathscr{F}(t,x_2,\theta)| \le L_R(1+|\theta|)|x_1-x_2|, \quad \text{for a.e. } t\in[0,T] \text{ and } x_1,x_2 \in B_R(0).$$

Additionally, it is necessary to specify the assumptions on the function ℓ that quantifies the discrepancy between the output of the network and its corresponding label.

Assumption 3. *The function $\ell : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}_+$ is C^1-regular and non-negative. Moreover, for every $R > 0$, there exists a constant $L_R > 0$ such that, for every $x_1, x_2 \in B_R(0)$, it holds*

$$|\nabla_x \ell(x_1,y_1) - \nabla_x \ell(x_2,y_2)| \le L_R\left(|x_1-x_2| + |y_1-y_2|\right). \tag{3.7}$$

Remark 3.1. We formulate here some explicit examples of controlled dynamics and loss function that satisfy the hypotheses listed in Assumptions 1–3. As regards the velocity field $\mathscr{F} : [0,T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$, if we denote with $\mathscr{A}$ the active components at the instant $t \in [0,T]$, then, using the same notations as in (2.6) and in (2.8), we have that

$$(\mathscr{F}(t,x,\theta))_k = \sigma\left(W_{k,\mathscr{A}} \cdot x_{\mathscr{A}} + b_k\right)$$

where $\theta = (W,b) \in \mathbb{R}^{d\times d} \times \mathbb{R}^d$, and where $\sigma : \mathbb{R} \to \mathbb{R}$ is the activation function. The case of a smooth and bounded activation was already considered in [8], and we refer the reader to [8, Remark 3.1] for examples of bounded functions that satisfy the assumptions. In this paper, we allow for activation functions that have sub-linear growth in the argument. Namely, in Section 5 we shall make use of a smoothed version of the Leaky ReLu that reads as

$$\sigma(z) = \alpha z + \frac{1-\alpha}{s}\log\left(1+e^{sz}\right), \quad z \in \mathbb{R}, \tag{3.8}$$

where $\alpha \in [0,1)$ and $s > 0$ are hyper-parameters. Computing the first and the second derivatives of σ, we obtain that

$$\sigma'(z) = \alpha + (1-\alpha)\frac{e^{sz}}{1+e^{sz}}, \quad \sigma''(z) = \frac{se^{sz}}{(1+e^{sz})^2}, \tag{3.9}$$

and it follows that $|\sigma'(z)| \le 1$ and $|\sigma''(z)| \le s$ for every $z \in \mathbb{R}$. Then, considering $k_1, k_2 \in \mathscr{A}$, we have that

$$\frac{\partial}{\partial x_{k_2}}\left(\mathscr{F}(t,x,\theta)\right)_{k_1} = \sigma'\left(W_{k_1,\mathscr{A}} \cdot x_{\mathscr{A}} + b_{k_1}\right) W_{k_1,k_2},$$

and it follows that

$$\left|\frac{\partial}{\partial x_{k_2}}\left(\mathscr{F}(t,x,\theta)\right)_{k_1} - \frac{\partial}{\partial x_{k_2}}\left(\mathscr{F}(t,y,\theta)\right)_{k_1}\right| \le s|W|^2|x-y|,$$

$$\left|\frac{\partial}{\partial x_{k_2}}\left(\mathscr{F}(t,x,(W,b))\right)_{k_1} - \frac{\partial}{\partial x_{k_2}}\left(\mathscr{F}(t,x,(W',b'))\right)_{k_1}\right| \le \left(1+s|x||W'|\right)|W-W'| + s|W'||b-b'|,$$

where we used that $\theta = (W,b), \theta' = (W',b')$. These show, respectively, that Assumption 2-(iv) and Assumption 2-(vi) are satisfied if we use (3.8) as the activation function. Then, we observe that

$$\frac{\partial}{\partial W_{k_1,k_2}}\left(\mathscr{F}(t,x,\theta)\right)_{k_1} = \sigma'\left(W_{k_1,\mathscr{A}} \cdot x_{\mathscr{A}} + b_{k_1}\right) x_{k_2},$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

yielding

$$\left|\frac{\partial}{\partial W_{k_1,k_2}}\big(\mathscr{F}(t,x,(W,b))\big)_{k_1}-\frac{\partial}{\partial W_{k_1,k_2}}\big(\mathscr{F}(t,x,(W',b'))\big)_{k_1}\right|\leq s|x|^2|W-W'|+s|x||b-b'|$$

$$\left|\frac{\partial}{\partial W_{k_1,k_2}}\big(\mathscr{F}(t,x,\theta)\big)_{k_1}-\frac{\partial}{\partial W_{k_1,k_2}}\big(\mathscr{F}(t,y,\theta)\big)_{k_1}\right|\leq\big(1+s|y||W|\big)|x-y|,$$

where we used that $\theta=(W,b),\theta'=(W',b')$. Similarly, as before, the last two inequalities establish, respectively, Assumption 2-(v) and Assumption 2-(vii). We report that the derivatives in the biases $\frac{\partial}{\partial b_{k_2}}\big(\mathscr{F}(t,x,\theta)\big)_{k_1}$ can be estimated with analogous expressions. Finally, we observe that Assumption 3 holds whenever $\ell:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ is of class C^2 in its variables, as for instance it is the case for $\ell(x,y)=|x-y|^2$.

Let us begin by establishing a regularity result for the reduced final cost, which refers to the cost function without the regularisation term.

Lemma 3.3 (Differentiability of the cost). *Let $T,R>0$ and $\mu_0\in\mathscr{P}_c(\mathbb{R}^{2d})$ be such that $supp(\mu_0)\subset B_R(0)$, and let us consider $\mathscr{F}:[0,T]\times\mathbb{R}^d\times\mathbb{R}^m\to\mathbb{R}^d$ and $\ell:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ that satisfy, respectively, Assumptions 1-2 and Assumption 3. Then, the reduced final cost*

$$J_\ell:\theta\in L^2([0,T];\mathbb{R}^m)\mapsto\begin{cases}\displaystyle\int_{\mathbb{R}^{2d}}\ell(x,y)\,d\mu_T^\theta(x,y),\\ s.t.\quad\begin{cases}\partial_t\mu_t^\theta(x,y)+\nabla_x\big(\mathscr{F}(t,x,\theta_t)\mu_t^\theta(x,y)\big)=0,\\ \mu_t^\theta|_{t=0}(x,y)=\mu_0(x,y),\end{cases}\end{cases}\tag{3.10}$$

is Fréchet-differentiable. Moreover, using the standard Hilbert space structure of $L^2([0,T],\mathbb{R}^m)$, we can represent the differential of J_ℓ at the point θ_0 as the function:

$$\nabla_\theta J_\ell(\theta):t\mapsto\int_{\mathbb{R}^{2d}}\nabla_\theta\mathscr{F}^\top\left(t,\Phi^{\theta_0}_{(0,t)}(x),\theta(t)\right)\cdot\mathscr{R}^\theta_{(t,T)}(x)^\top\cdot\nabla_x\ell^\top\left(\Phi^\theta_{(0,T)}(x),y\right)\,d\mu_0(x,y)\tag{3.11}$$

for a.e. $t\in[0,T]$.

Before proving the statement, we need to introduce the linear operator $\mathscr{R}^\theta_{\tau,s}(x):\mathbb{R}^d\to\mathbb{R}^d$ with $\tau,s\in[0,T]$, that is related to the linearisation along a trajectory of the dynamics of the control system (2.1), and that appears in (3.11). Given an admissible control $\theta\in L^2([0,T],\mathbb{R}^m)$, let us consider the corresponding trajectory curve $t\mapsto\Phi^\theta_{(0,t)}(x)$ for $t\in[0,T]$, i.e., the solution of (2.1) starting at the point $x\in\mathbb{R}^d$ at the initial instant $t=0$. Given any $\tau\in[0,T]$, we consider the following linear ODE in the phase space $\mathbb{R}^{d\times d}$:

$$\begin{cases}\frac{d}{ds}\mathscr{R}^\theta_{(\tau,s)}(x)=\nabla_x\mathscr{F}\left(s,\Phi^\theta_{(0,s)}(x),\theta(s)\right)\cdot\mathscr{R}^\theta_{(\tau,s)}(x)\quad\text{for a.e. } s\in[0,T],\\ \mathscr{R}^\theta_{(\tau,\tau)}(x)=\mathrm{Id}.\end{cases}\tag{3.12}$$

We insist on the fact that, when we write $\mathscr{R}^\theta_{(\tau,s)}(x)$, x denotes the starting point of the trajectory along which the dynamics have been linearised. We observe that, using Assumption 2-$(iv)-(vi)$ and Caratheodory Theorem, it follows that (3.12) admits a unique solution, for every $x\in\mathbb{R}^d$ and for every $\tau\in[0,T]$. Since it is an elementary object in Control Theory, the properties of $\mathscr{R}^\theta$ are discussed in the Appendix (see Proposition A.7). We just recall here that the following relation is satisfied:

$$\mathscr{R}^\theta_{\tau,s}(x)=\nabla_x\Phi^\theta_{(\tau,s)}\big|_{\Phi^\theta_{(0,\tau)}(x)}\tag{3.13}$$

for every $\tau,s\in[0,T]$ and for every $x\in\mathbb{R}^d$ (see, e.g., [10, Theorem 2.3.2]). Moreover, for every $\tau,s\in[0,T]$ the following identity holds:

$$\mathscr{R}^\theta_{\tau,s}(x)\cdot\mathscr{R}^\theta_{s,\tau}(x)=\mathrm{Id},$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

i.e., the matrices $\mathscr{R}^{\theta}_{\tau,s}(x), \mathscr{R}^{\theta}_{s,\tau}(x)$ are one the inverse of the other. From this fact, it is possible to deduce that

$$\frac{\partial}{\partial \tau}\mathscr{R}^{\theta}_{\tau,s}(x) = -\mathscr{R}^{\theta}_{\tau,s}(x)\cdot\nabla_x \mathscr{F}\left(\tau, \Phi^{\theta}_{(0,\tau)}(x), \theta(\tau)\right) \tag{3.14}$$

for almost every $\tau, s \in [0,T]$ (see, e.g., [10, Theorem 2.2.3] for the details).

Proof of Lemma 3.3. Let us fix an admissible control $\theta \in L^2([0,T];\mathbb{R}^m)$ and let $\mu^{\theta}_{\cdot} \in \mathscr{C}^0([0,T];\mathscr{P}_c(\mathbb{R}^{2d}))$ be the unique solution of the continuity equation (3.3), corresponding to the control θ and satisfying $\mu^{\theta}|_{t=0} = \mu_0$. According to Proposition 3.1, this curve can be expressed as $\mu^{\theta}_t = \mathbf{\Phi}^{\theta}_{(0,t)\#}\mu_0$ for every $t \in [0,T]$, where the map $\mathbf{\Phi}^{\theta}_{(0,t)} = \left(\Phi^{\theta}_{(0,t)}, \mathrm{Id}\right) : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ has been introduced in (3.2) as the flow of the extended control system (3.1). In particular, we can rewrite the terminal cost J_ℓ defined in (3.10) as

$$J_\ell(\theta) = \int_{\mathbb{R}^{2d}} \ell\left(\Phi^{\theta}_{(0,T)}(x), y\right) d\mu_0(x,y).$$

In order to compute the gradient $\nabla_\theta J_\ell$, we preliminarily need to focus on the differentiability with respect to θ of the mapping $\theta \mapsto \ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)$, when (x,y) is fixed. Indeed, given another control $\vartheta \in L^2([0,T];\mathbb{R}^m)$ and $\varepsilon > 0$, from Proposition A.6 it descends that

$$\begin{aligned}\Phi^{\theta+\varepsilon\vartheta}_{(0,T)}(x) &= \Phi^{\theta}_{(0,T)}(x) + \varepsilon\xi^{\theta}(T) + o_\theta(\varepsilon)\\ &= \Phi^{\theta}_{(0,T)}(x) + \varepsilon\int_0^T \mathscr{R}^{\theta}_{(s,T)}(x)\nabla_\theta \mathscr{F}\left(s, \Phi^{\theta}_{(0,s)}(x), \theta(s)\right)\vartheta(s)ds + o_\theta(\varepsilon) \qquad \text{as } \varepsilon \to 0,\end{aligned} \tag{3.15}$$

where $o_\theta(\varepsilon)$ is uniform for every $x \in B_R(0) \subset \mathbb{R}^d$, and as ϑ varies in the unit ball of L^2. Owing to Assumption 3, for every $x, y, v \in B_R(0)$ we observe that

$$|\ell(x+\varepsilon v + o(\varepsilon), y) - \ell(x,y) - \varepsilon\nabla_x\ell(x,y)\cdot v| \le |\nabla_x\ell(x,y)|o(\varepsilon) + \frac{1}{2}L_R|\varepsilon v + o(\varepsilon)|^2 \qquad \text{as } \varepsilon \to 0. \tag{3.16}$$

Therefore, by combining (3.15) and (3.16), we obtain that

$$\begin{aligned}&\ell\left(\Phi^{\theta+\varepsilon\vartheta}_{(0,T)}(x), y\right) - \ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)\\ &\quad= \varepsilon\int_0^T \left(\nabla_x\ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)\cdot\mathscr{R}^{\theta}_{(s,T)}(x)\cdot\nabla_\theta\mathscr{F}\left(s, \Phi^{\theta}_{(0,s)}(x), \theta(s)\right)\right)\cdot\vartheta(s)ds + o_\theta(\varepsilon).\end{aligned}$$

Since the previous expression is uniform for $x, y \in B_R(0)$, then if we integrate both sides of the last identity with respect to μ_0, we have that

$$\begin{aligned}&J_\ell(\theta+\varepsilon\vartheta) - J_\ell(\theta)\\ &\quad= \varepsilon\int_{\mathbb{R}^{2d}}\int_0^T \left(\nabla_x\ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)\cdot\mathscr{R}^{\theta}_{(s,T)}(x)\cdot\nabla_\theta\mathscr{F}\left(s, \Phi^{\theta}_{(0,s)}(x), \theta(s)\right)\right)\cdot\vartheta(s)\,ds\,d\mu_0(x,y) + o_\theta(\varepsilon).\end{aligned} \tag{3.17}$$

This proves the Fréchet differentiability of the functional J_ℓ at the point θ. We observe that, from Proposition 2.1, Proposition A.7 and Assumption 2, it follows that the function $s \mapsto \nabla_x\ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)\cdot\mathscr{R}^{\theta}_{(s,T)}(x)\cdot\nabla_\theta\mathscr{F}\left(s, \Phi^{\theta}_{(0,s)}(x), \theta(s)\right)$ is uniformly bounded in L^2, as x, y vary in $B_R(0) \subset \mathbb{R}^d$. Then, using Fubini Theorem, the first term of the expansion (3.17) can be rewritten as

$$\int_0^T \left(\int_{\mathbb{R}^{2d}} \nabla_x\ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)\cdot\mathscr{R}^{\theta}_{(s,T)}(x)\cdot\nabla_\theta\mathscr{F}\left(s, \Phi^{\theta}_{(0,s)}(x), \theta(s)\right)\,d\mu_0(x,y)\right)\cdot\vartheta(s)\,ds.$$

Hence, from the previous asymptotic expansion and from Riesz Representation Theorem, we deduce (3.11). □

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

We now prove the most important result of this subsection, concerning the Lipschitz regularity of the gradient $\nabla_\theta J_\ell$.

Proposition 3.4. *Under the same assumptions and notations as in Lemma 3.3, we have that the gradient $\nabla_\theta J_\ell : L^2([0,T],\mathbb{R}^m) \to L^2([0,T],\mathbb{R}^m)$ is Lipschitz continuous on every bounded set of L^2. More precisely, given $\theta_1, \theta_2 \in L^2([0,T];\mathbb{R}^m)$, there exists a constant $\mathscr{L}(T,R,\|\theta_1\|_{L^2},\|\theta_2\|_{L^2}) > 0$ such that*

$$\left\|\nabla_\theta J_\ell(\theta_1) - \nabla_\theta J_\ell(\theta_2)\right\|_{L^2} \leq \mathscr{L}(T,R,\|\theta_1\|_{L^2},\|\theta_2\|_{L^2})\left\|\theta_1-\theta_2\right\|_{L^2}.$$

Proof. Let us consider two admissible controls $\theta_1,\theta_2 \in L^2([0,T],\mathbb{R}^m)$ such that $\|\theta_1\|_{L^2},\|\theta_1\|_{L^2} \leq C$. In order to simplify the notations, given $x \in B_R(0) \subset \mathbb{R}^d$, we define the curves $x_1 : [0,T] \to \mathbb{R}^d$ and $x_2 : [0,T] \to \mathbb{R}^d$ as

$$x_1(t) := \Phi^{\theta_1}_{(0,t)}(x), \quad x_2(t) := \Phi^{\theta_2}_{(0,t)}(x)$$

for every $t \in [0,T]$, where the flows $\Phi^{\theta_1}, \Phi^{\theta_2}$ where introduced in (2.2). We recall that, in virtue of Proposition 2.1, $x_1(t), x_2(t) \in B_{\bar R}(0)$ for every $t \in [0,1]$. Then, for every $y \in B_R(0)$, we observe that

$$\begin{aligned}
&\left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big)\mathscr{R}^{\theta_1}_{(t,T)}(x)^\top\nabla_x\ell^\top\big(x_1(T),y\big) - \nabla_\theta\mathscr{F}^\top\big(t,x_1(t),\theta_2(t)\big)\mathscr{R}^{\theta_2}_{(t,T)}(x)^\top\nabla_x\ell^\top\big(x_2(T),y\big)\right| \\
&\quad\leq \left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big)\right|\left|\mathscr{R}^{\theta_1}_{(t,T)}(x)^\top\right|\left|\nabla_x\ell^\top\big(x_1(T),y\big) - \nabla_x\ell^\top\big(x_2(T),y\big)\right| \\
&\qquad+\left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big)\right|\left|\mathscr{R}^{\theta_1}_{(t,T)}(x)^\top - \mathscr{R}^{\theta_2}_{(t,T)}(x)^\top\right|\left|\nabla_x\ell^\top\big(x_2(T),y\big)\right| \\
&\qquad+\left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big) - \nabla_\theta\mathscr{F}^\top\big(t,x_2(t),\theta_2(t)\big)\right|\left|\mathscr{R}^{\theta_2}_{(t,T)}(x)^\top\right|\left|\nabla_x\ell^\top\big(x_2(T),y\big)\right| \qquad (3.18)
\end{aligned}$$

for a.e. $t \in [0,T]$. We bound separately the three terms at the right-hand side of (3.18). As regards the first addend, from Assumption 2-(v), Assumption 3, Proposition A.7 and Lemma A.4, we deduce that there exists a positive constant $C_1 > 0$ such that

$$\begin{aligned}
&\left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big)\right|\left|\mathscr{R}^{\theta_1}_{(t,T)}(x)^\top\right|\left|\nabla_x\ell^\top\big(x_1(T),y\big) - \nabla_x\ell^\top\big(x_2(T),y\big)\right| \\
&\qquad\leq C_1\,(1+|\theta_1(t)|)\,\|\theta_1-\theta_2\|_{L^2} \qquad (3.19)
\end{aligned}$$

for a.e. $t \in [0,T]$. Similarly, using again Assumption 2-(v), Assumption 3 and Proposition A.7 on the second addend at the right-hand side of (3.18), we obtain that there exists $C_2 > 0$ such that

$$\left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big)\right|\left|\mathscr{R}^{\theta_1}_{(t,T)}(x)^\top - \mathscr{R}^{\theta_2}_{(t,T)}(x)^\top\right|\left|\nabla_x\ell^\top\big(x_2(T),y\big)\right| \leq C_2\,(1+|\theta_1(t)|)\,\|\theta_1-\theta_2\|_{L^2} \qquad (3.20)$$

for a.e. $t \in [0,T]$. Moreover, the third term can be bounded as follows:

$$\begin{aligned}
&\left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big) - \nabla_\theta\mathscr{F}^\top\big(t,x_2(t),\theta_2(t)\big)\right|\left|\mathscr{R}^{\theta_2}_{(t,T)}(x)^\top\right|\left|\nabla_x\ell^\top\big(x_2(T),y\big)\right| \\
&\qquad\leq C_3\,[(1+|\theta_1(t)|)\|\theta_1-\theta_2\|_{L^2} + |\theta_1(t)-\theta_2(t)|] \qquad (3.21)
\end{aligned}$$

for a.e. $t \in [0,T]$, where we used Assumption 2-$(v)-(vii)$, Proposition A.7 and Lemma A.4. Therefore, combining (3.18)–(3.21), we deduce that

$$\begin{aligned}
&\left|\nabla_\theta \mathscr{F}^\top\big(t,x_1(t),\theta_1(t)\big)\mathscr{R}^{\theta_1}_{(t,T)}(x)^\top\nabla_x\ell^\top\big(x_1(T),y\big) - \nabla_\theta\mathscr{F}^\top\big(t,x_1(t),\theta_2(t)\big)\mathscr{R}^{\theta_2}_{(t,T)}(x)^\top\nabla_x\ell^\top\big(x_2(T),y\big)\right| \\
&\qquad\leq \bar C\,[(1+|\theta_1(t)|)\|\theta_1-\theta_2\|_{L^2} + |\theta_1(t)-\theta_2(t)|] \qquad (3.22)
\end{aligned}$$

for a.e. $t \in [0,T]$. We observe that the last inequality holds for every $x,y \in B_R(0)$. Therefore, if we integrate both sides of (3.22) with respect to the probability measure μ_0, recalling the expression of the gradient of J_ℓ reported in (3.11), we have that

$$|\nabla_\theta J_\ell(\theta_1)[t] - \nabla_\theta J_\ell(\theta_1)[t]| \leq \bar C\,[(1+|\theta_1(t)|)\|\theta_1-\theta_2\|_{L^2} + |\theta_1(t)-\theta_2(t)|] \qquad (3.23)$$

for a.e. $t \in [0,T]$, and this concludes the proof. □

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

From the previous result, we can deduce that the terminal cost $J_\ell : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ is locally semiconvex.

Corollary 3.5 (Local semiconvexity of the cost functional). *Under the same assumptions and notations as in Lemma 3.3, let us consider a bounded subset $\Gamma \subset L^2([0,T];\mathbb{R}^m)$. Then, $\nabla_\theta J : L^2([0,T]) \to L^2([0,T])$ is Lipschitz continuous on Γ. Moreover, there exists a constant $\mathscr{L}(T,R,\Gamma) > 0$ such that the cost functional $J : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ defined in (3.6) satisfies the following semiconvexity estimate:*

$$J\big((1-\zeta)\theta_1 + \zeta\theta_2\big) \le (1-\zeta)J(\theta_1) + \zeta J(\theta_2) - (2\lambda - \mathscr{L}(T,R,\Gamma))\tfrac{\zeta(1-\zeta)}{2}\|\theta_1-\theta_2\|_2^2 \tag{3.24}$$

for every $\theta_1,\theta_2 \in \Gamma$ and for every $\zeta \in [0,1]$. In particular, if $\lambda > \frac{1}{2}\mathscr{L}(T,R,\Gamma)$, the cost functional J is strictly convex over Γ.

Proof. We recall that $J(\theta) = J_\ell(\theta) + \lambda\|\theta\|_{L^2}^2$, where J_ℓ has been introduced in (3.10). Owing to Proposition 3.4, it follows that $\nabla_\theta J_\ell$ is Lipschitz continuous on Γ with constant $\mathscr{L}(T,R,\Gamma)$. This implies that J is Lipschitz continuous as well on Γ. Moreover, it descends that

$$J_\ell\big((1-\zeta)\theta_1 + \zeta\theta_2\big) \le (1-\zeta)J_\ell(\theta_1) + \zeta J_\ell(\theta_2) + \mathscr{L}(T,R,\Gamma)\tfrac{\zeta(1-\zeta)}{2}\|\theta_1-\theta_2\|_{L^2}^2$$

for every $\theta_1,\theta_2 \in \Gamma$ and for every $\zeta \in [0,1]$. On the other hand, recalling that

$$\|(1-\zeta)\theta_1 + \zeta\theta_2\|_{L^2}^2 = (1-\zeta)\|\theta_1\|_{L^2}^2 + \zeta\|\theta_2\|_{L^2}^2 - \zeta(1-\zeta)\|\theta_1-\theta_2\|_{L^2}^2$$

for every $\theta_1,\theta_2 \in L^2$, we immediately deduce (3.24). □

Remark 3.2. When the parameter $\lambda > 0$ that tunes the L^2-regularization is large enough, we can show that the functional J defined by (3.6) admits a unique global minimiser. Indeed, since the control identically 0 is an admissible competitor, we have that

$$\inf_{\theta\in L^2} J(\theta) \le J(0) = J_\ell(0),$$

where we observe that the right-hand side is not affected by the value of λ. Hence, recalling that $J(\theta) = J_\ell(\theta) + \lambda\|\theta\|_{L^2}^2$, we have that the sublevel set $\{\theta : J(\theta) \le J_\ell(0)\}$ is included in the ball $B_\lambda := \{\theta : \|\theta\|_{L^2}^2 \le \frac{1}{\lambda}J_\ell(0)\}$. Since these balls are decreasing as λ increases, owing to Corollary 3.5, we deduce that there exists a parameter $\bar\lambda > 0$ such that the cost functional J is strongly convex when restricted to $B_{\bar\lambda}$. Then, Lemma 3.3 guarantees that the functional $J : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ introduced in (3.6) is continuous with respect to the strong topology of L^2, while the convexity implies that it is weakly lower semi-continuous as well. Being the ball $B_{\bar\lambda}$ weakly compact, we deduce that the restriction to $B_{\bar\lambda}$ of the functional J admits a unique minimiser θ^*. However, since $B_{\bar\lambda}$ includes the sublevel set $\{\theta : J(\theta) \le J_\ell(0)\}$, it follows that θ^* is actually the unique global minimiser. It is interesting to observe that, even though λ is chosen large enough to ensure existence (and uniqueness) of the global minimiser, it is not possible to conclude that the functional J is globally convex. This is essentially due to the fact that Corollary 3.5 holds only on bounded subsets of L^2.

Taking advantage of the representation of the gradient of the terminal cost J_ℓ provided by (3.11), we can formulate the necessary optimality conditions for the cost J introduced in (3.6). In order to do that, we introduce the function $p : [0,T]\times\mathbb{R}^d\times\mathbb{R}^d \to \mathbb{R}^d$ as follows:

$$p_t(x,y) := \nabla_x\ell\left(\Phi^\theta_{(0,T)}(x), y\right)\cdot\mathscr{R}^\theta_{(t,T)}(x), \tag{3.25}$$

where $\mathscr{R}^\theta_{(t,T)}(x)$ is defined according to (3.12). We observe that p (as well as $\nabla_x\ell$) should be understood as a row vector. Moreover, using (3.14), we deduce that, for every $x,y\in\mathbb{R}^d$, the $t\mapsto p_t(x,y)$ is solving the following backward Cauchy problem:

$$\frac{\partial}{\partial t}p_t(x,y) = -p_t(x,y)\cdot\nabla_x\mathscr{F}\left(t,\Phi^\theta_{(0,t)}(x),\theta(t)\right), \qquad p_T(x,y) = \nabla_x\ell\left(\Phi^\theta_{(0,T)}(x),y\right). \tag{3.26}$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Hence, we can equivalently rewrite $\nabla_\theta J_\ell$ using p:

$$\nabla_\theta J_\ell(\theta)[t] = \int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top \left(t, \Phi^\theta_{(0,t)}(x), \theta(t)\right) \cdot p_t^\top(x,y)\, d\mu_0(x,y) \tag{3.27}$$

for almost every $t \in [0,T]$. Therefore, recalling that $J(\theta) = J_\ell(\theta) + \lambda \|\theta\|^2_{L^2}$, we deduce that the stationary condition $\nabla_\theta J(\theta^*) = 0$ can be rephrased as

$$\begin{cases} \partial_t \mu_t^*(x,y) + \nabla_x \cdot \left(\mathscr{F}(t,x,\theta^*(t))\mu_t^*(x,y)\right) = 0, & \mu_t^*|_{t=0}(x,y) = \mu_0(x,y), \\ \partial_t p_t^*(x,y) = -p_t^*(x,y) \cdot \nabla_x \mathscr{F}\left(t, \Phi^{\theta^*}_{(0,t)}(x), \theta^*(t)\right), & p_t^*|_{t=T}(x,y) = \nabla_x \ell\left(\Phi^{\theta^*}_{(0,T)}(x), y\right), \\ \theta^*(t) = -\dfrac{1}{2\lambda} \displaystyle\int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top \left(t, \Phi^{\theta^*}_{(0,t)}(x), \theta^*(t)\right) \cdot p_t^{*\top}(x,y)\, d\mu_0(x,y). \end{cases} \tag{3.28}$$

Remark 3.3. The computation of p through the backward integration of (3.26) can be interpreted as the control-theoretic equivalent of the "back-propagation of the gradients". We observe that, in order to check whether (3.28) is satisfied, it is sufficient to evaluate p^* only on $\mathrm{supp}(\mu_0)$. Moreover, the evaluation of p^* on different points $(x_1,y_1), (x_2,y_2) \in \mathrm{supp}(\mu_0)$ involves the resolution of two uncoupled backward ODEs. This means that, when dealing with a measure μ_0 that charges only finitely many points, we can solve the equation (3.26) in parallel for every point in $\mathrm{supp}(\mu_0)$.

In virtue of Proposition 3.4, we can study the gradient flow induced by the cost functional $J : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ on its domain. More precisely, given an admissible control $\theta_0 \in L^2([0,T],\mathbb{R}^m)$, we consider the gradient flow equation:

$$\begin{cases} \dot\theta(\omega) = -\nabla_\theta J(\theta(\omega)) \quad \text{for } \omega \geq 0, \\ \theta(0) = \theta_0. \end{cases} \tag{3.29}$$

In the next result, we show that the gradient flow equation (3.29) is well-posed and that the solution is defined for every $\omega \geq 0$. In the particular case of linear-control systems, the properties of the gradient flow trajectories have been investigated in [42].

Lemma 3.6. *Let $T, R > 0$ and $\mu_0 \in \mathscr{P}_c(\mathbb{R}^{2d})$ be a probability measure such that $supp(\mu_0) \subset B_R(0)$, and let us consider $\mathscr{F} : [0,T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ and $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ that satisfy, respectively, Assumptions 1–2 and 3. Then, for every $\theta_0 \in L^2([0,T],\mathbb{R}^m)$, the gradient flow equation (3.29) admits a unique solution $\omega \mapsto \theta(\omega)$ of class C^1 that is defined for every $\omega \in [0,+\infty)$.*

Proof. Let us consider $\theta_0 \in L^2([0,T],\mathbb{R}^m)$, and let us introduce the sublevel set

$$\Gamma := \{\theta \in L^2([0,T],\mathbb{R}^m) : J(\theta) \leq J(\theta_0)\},$$

where J is the functional introduced in (3.6) defining the mean-field optimal control problem. Using the fact that the end-point cost $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ is non-negative, we deduce that $\Gamma \subset \left\{\theta \in L^2([0,T],\mathbb{R}^m) : \|\theta\|^2_{L^2} \leq \frac{1}{\lambda} J(\theta_0)\right\}$. Hence, from Proposition 3.4 it follows that the gradient field $\nabla_\theta J$ is Lipschitz (and bounded) on Γ. Hence, using a classical result on ODE in Banach spaces (see, e.g., [31, Theorem 5.1.1]), it follows that the initial value problem (3.29) admits a unique small-time solution $\omega \mapsto \theta(\omega)$ of class C^1 defined for $\omega \in [-\delta, \delta]$, with $\delta > 0$. Moreover, we observe that

$$\frac{d}{d\omega} J(\theta(\omega)) = \langle \nabla_\theta J(\theta(\omega)), \dot\theta(\omega)) \rangle = -\|\nabla_\theta J(\theta(\omega))\|_{L^2} \leq 0,$$

and this implies that $\theta(\omega) \in \Gamma$ for every $\omega \in [0,\delta]$. Hence, it is possible to recursively extend the solution to every interval of the form $[0,M]$, with $M > 0$. □

We observe that, under the current working assumptions, we cannot provide any convergence result for the gradient flow trajectories. This is not surprising since, when the regularisation parameter $\lambda > 0$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

is small, it is not even possible to prove that the functional J admits minimisers. Indeed, the argument presented in Remark 3.2 requires the regularisation parameter λ to be sufficiently large.

We conclude the discussion with an observation on a possible discretization of (3.29). If we fix a sufficiently small parameter $\tau > 0$, given an initial guess θ_0, we can consider the sequence of controls $(\theta_k^\tau)_{k\geq 0} \subset L^2([0,T],\mathbb{R}^m)$ defined through the Minimizing Movement Scheme:

$$\theta_0^\tau = \theta_0, \qquad \theta_{k+1}^\tau \in \arg\min_\theta \left[J(\theta) + \frac{1}{2\tau}\|\theta - \theta_k^\tau\|_{L^2}^2 \right] \quad \text{for every } k \geq 0. \tag{3.30}$$

Remark 3.4. We observe that the minimisation problems in (3.30) are well-posed as soon as the functionals $\theta \mapsto J_{\theta_k}^\tau(\theta) := J(\theta) + \frac{1}{2\tau}\|\theta - \theta_k^\tau\|_{L^2}^2$ are strictly convex on the bounded sublevel set $K_{\theta_0} := \left\{\theta : J(\theta) \leq J\left(\theta_0^\tau\right)\right\}$, for every $k \geq 0$. Hence, the parameter $\tau > 0$ can be calibrated by means of the estimates provided by Corollary 3.5, considering the bounded set K_{θ_0}. Then, using and inductive argument, it follows that, for every $k \geq 0$, the functional $J_{\theta_k}^\tau : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ admits a unique global minimiser θ_{k+1}^τ. Also for $J_{\theta_k}^\tau$ we can derive the necessary conditions for optimality satisfied by θ_{k+1}, which are analogous to the ones formulated in (3.28), and which descend as well from the identity $\nabla_\theta J_{\theta_k}^\tau\left(\theta_{k+1}^\tau\right) = 0$:

$$\begin{cases} \partial_t \mu_t(x,y) + \nabla_x \cdot \left(\mathscr{F}\left(t,x,\theta_{k+1}^\tau(t)\right)\mu_t(x,y)\right) = 0, & \mu_t|_{t=0}(x,y) = \mu_0(x,y), \\ \partial_t p_t(x,y) = -p_t(x,y)\cdot\nabla_x \mathscr{F}\left(t, \Phi_{(0,t)}^{\theta_{k+1}^\tau}(x), \theta_{k+1}^\tau(t)\right), & p_t|_{t=T}(x,y) = \nabla_x \ell\left(\Phi_{(0,T)}^{\theta_{k+1}^\tau}(x), y\right), \\ \theta_{k+1}^\tau(t) = -\dfrac{1}{1+2\lambda\tau}\left(\theta_k^\tau(t) - \tau\displaystyle\int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top\left(t, \Phi_{(0,t)}^{\theta_{k+1}^\tau}(x), \theta_{k+1}^\tau(t)\right)\cdot p_t^\top(x,y)\, d\mu_0(x,y)\right). \end{cases} \tag{3.31}$$

Finally, we observe that the mapping $\Lambda^\tau : L^2([0,T],\mathbb{R}^m) \to L^2([0,T],\mathbb{R}^m)$ defined for a.e. $t \in [0,T]$ as

$$\Lambda_{\theta_k^\tau}^\tau(\theta)[t] := -\frac{1}{1+2\lambda\tau}\left(\theta_k^\tau(t) - \tau\int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top\left(t, \Phi_{(0,t)}^\theta(x), \theta(t)\right)\cdot p_t^\top(x,y)\, d\mu_0(x,y)\right) \tag{3.32}$$

is a contraction on K_{θ_0} as soon as

$$\frac{\tau}{1+2\lambda\tau}\mathrm{Lip}\left(\nabla_\theta J_\ell|_{K_{\theta_0}}\right) < 1.$$

For every $\tau > 0$ such that the sequence $\left(\theta_k^\tau\right)_{k\geq 0}$ is defined, we denote with $\tilde\theta^\tau : [0,+\infty) \to L^2([0,T],\mathbb{R}^m)$ the piecewise affine interpolation obtained as

$$\tilde\theta^\tau(\omega) = \theta_k^\tau + \frac{\theta_{k+1}^\tau - \theta_k^\tau}{\tau}(\omega - k\tau) \quad \text{for } \omega \in [k\tau,(k+1)\tau]. \tag{3.33}$$

We finally report a classical result concerning the convergence of the piecewise affine interpolation $\tilde\theta^\tau$ to the gradient flow trajectory solving (3.29).

Proposition 3.7. *Under the same assumptions and notations as in Lemma 3.6, let us consider an initial point $\theta_0 \in L^2([0,T],\mathbb{R}^m)$ and a sequence $(\tau_j)_{j\in\mathbb{N}}$ such that $\tau_j \to 0$ as $j \to \infty$, and let $\left(\tilde\theta^{\tau_j}\right)_{j\in\mathbb{N}}$ be the sequence of piecewise affine curves defined by (3.33). Then, for every $\Omega > 0$, there exists a subsequence $\left(\tilde\theta^{\tau_{j_k}}\right)_{k\in\mathbb{N}}$ converging uniformly on the interval $[0,\Omega]$ to the solution of (3.29) starting from θ_0.*

Proof. The proof follows directly from [41, Proposition 2.3]. □

3.3. Finite-particles approximation

In this section, we study the stability of the mean-field optimal control problem (3.6) with respect to finite-samples distributions. More precisely, assume that we are given samples $\left\{\left(X_0^i, Y_0^i\right)\right\}_{i=1}^N$ of size $N \geq 1$ independently and identically distributed according to $\mu_0 \in \mathscr{P}_c(\mathbb{R}^{2d})$, and consider the empirical loss minimisation problem

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

$$\inf_{\theta\in L^2([0,T];\mathbb{R}^m)} J^N(\theta) := \begin{cases} \dfrac{1}{N}\displaystyle\sum_{i=1}^{N} \ell\big(X^i(T), Y^i(T)\big) + \lambda \int_0^T |\theta(t)|^2\, dt \\ \text{s.t.} \begin{cases} \dot{X}^i(t) = \mathscr{F}(t, X^i(t), \theta(t)), & \dot{Y}^i(t) = 0, \\ \big(X^i(t), Y^i(t)\big)\big|_{t=0} = \big(X_0^i, Y_0^i\big), & i \in \{1, \dots, N\}. \end{cases} \end{cases} \tag{3.34}$$

By introducing the empirical measure $\mu_0^N \in \mathscr{P}_c^N(\mathbb{R}^{2d})$, defined as

$$\mu_0^N := \frac{1}{N}\sum_{i=1}^{N} \delta_{\left(x_0^i, y_0^i\right)},$$

the cost function in (3.34) can be rewritten as

$$J^N(\theta) = \int_{\mathbb{R}^{2d}} \ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)\, d\mu_0^N(x,y) + \lambda\|\theta\|_{L^2}^2 \tag{3.35}$$

for every $\theta \in L^2([0,T],\mathbb{R}^m)$, and the empirical loss minimisation problem in (3.34) can be recast as a mean-field optimal control problem with initial datum μ_0^N. In this section, we are interested in studying the asymptotic behaviour of the functional J^N as N tends to infinity. More precisely, we consider a sequence of probability measures $\left(\mu_0^N\right)_{N\geq 1}$ such that μ_0^N charges uniformly N points, and such that

$$W_1\big(\mu_0^N, \mu_0\big) \underset{N\to+\infty}{\longrightarrow} 0.$$

Then, in Proposition 3.8, we study the uniform convergence of J^N and of $\nabla_\theta J^N$ to J and $\nabla_\theta J^N$, respectively, where $J : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ is the functional defined in (3.6) and corresponding to the limiting measure μ_0. Moreover, in Theorem 3.9, assuming the existence of a region where the functionals J^N are uniformly strongly convex, we provide an estimate of the so-called *generalisation error* in terms of the distance $W_1\big(\mu_0^N, \mu_0\big)$.

Proposition 3.8 (Uniform convergence of J^N and $\nabla_\theta J^N$). *Let us consider a probability measure $\mu_0 \in \mathscr{P}_c(\mathbb{R}^{2d})$ and a sequence $\left(\mu_0^N\right)_{N\geq 1}$ such that $\mu_0^N \in \mathscr{P}_c^N(\mathbb{R}^{2d})$ for every $N \geq 1$. Let us further assume that $W_1\big(\mu_0^N, \mu_0\big) \to 0$ as $N \to \infty$, and that there exists $R > 0$ such that $supp(\mu_0), supp\big(\mu_0^N\big) \subset B_R(0)$ for every $N \geq 1$. Given $T > 0$, let $\mathscr{F} : [0,T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ and $\ell : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ satisfy, respectively, Assumptions 1–2 and Assumption 3, and let $J, J^N : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ be the cost functionals defined in (3.6) and (3.34), respectively. Then, for every bounded subset $\Gamma \subset L^2([0,T],\mathbb{R}^m)$, we have that*

$$\lim_{N\to\infty} \sup_{\theta\in\Gamma} |J^N(\theta) - J(\theta)| = 0 \tag{3.36}$$

and

$$\lim_{N\to\infty} \sup_{\theta\in\Gamma} \|\nabla_\theta J^N(\theta) - \nabla_\theta J(\theta)\|_{L^2} = 0, \tag{3.37}$$

where J was introduced in (3.6), and J^N is defined as in (3.35).

Proof. Since we have that $J(\theta) = J_\ell(\theta) + \lambda\|\theta\|_{L^2}^2$ and $J^N(\theta) = J_\ell^N(\theta) + \lambda\|\theta\|_{L^2}^2$, it is sufficient to prove (3.36)–(3.37) for J_ℓ and J_ℓ^N, where we set

$$J_\ell^N(\theta) := \int_{\mathbb{R}^{2d}} \ell\left(\Phi^{\theta}_{(0,T)}(x), y\right)\, d\mu_0^N(x,y)$$

for every $\theta \in L^2$ and for every $N \geq 1$. We first observe that, for every $\theta \in L^2([0,T],\mathbb{R}^m)$ such that $\|\theta\|_{L^2} \leq \rho$, from Proposition 3.2 it follows that $\mathrm{supp}(\mu_0), \mathrm{supp}\big(\mu_0^N\big) \subset B_{\bar{R}}(0)$, for some $\bar{R} > 0$. Then, denoting with

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

$t \mapsto \mu_t^N$ and $t \mapsto \mu_t$ the solutions of the continuity equation (3.3) driven by the control θ and with initial datum, respectively, μ_0^N and μ_0, we compute

$$\begin{aligned}\left|J_\ell^N(\theta) - J_\ell(\theta)\right| &= \left|\int_{\mathbb{R}^{2d}} \ell\left(\Phi^\theta_{(0,T)}(x), y\right)\ \left(d\mu_0^N - d\mu_0\right)(x,y)\right| = \left|\int_{\mathbb{R}^{2d}} \ell(x,y)\left(d\mu_T^N - d\mu_T\right)(x,y)\right| \\ &\leq \bar{L}_1\bar{L}_2 W_1\left(\mu_0^N, \mu_0\right), \end{aligned} \tag{3.38}$$

where we have used (3.2) and Proposition 3.1 in the second identity, and we have indicated with $\bar{L}_1$ the Lipschitz constant of ℓ on $B_{\bar{R}}(0)$, while $\bar{L}_2$ descends from the continuous dependence of solutions of (3.3) on the initial datum (see Proposition 3.2). We insist on the fact that both $\bar{L}_1, \bar{L}_2$ depend on ρ, i.e., the upper bound on the L^2-norm of the controls.

We now address the uniform convergence of $\nabla_\theta J_\ell^N$ to $\nabla_\theta J_\ell$ on bounded sets of L^2. As before, let us consider an admissible control θ such that $\|\theta\|_{L^2} \leq \rho$. Hence, using the representation provided in (3.11), for a.e. $t \in [0,T]$ we have:

$$\begin{aligned}&\left|\nabla_\theta J_\ell^N(\theta)[t] - \nabla_\theta J_\ell(\theta)[t]\right| = \\ &\left|\int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top\left(t, \Phi^{\theta_0}_{(0,t)}(x), \theta_0(t)\right) \cdot \mathscr{R}^{\theta_0}_{(t,T)}(x)^\top \cdot \nabla_x \ell^\top\left(\Phi^{\theta_0}_{(0,T)}(x), y\right)\left(d\mu_0^N - d\mu_0\right)(x,y)\right|, \end{aligned} \tag{3.39}$$

In order to prove uniform convergence in L^2 norm, we have to show that the integrand is Lipschitz continuous in (x,y) for a.e. $t \in [0,T]$, where the Lipschitz constant has to be L^2-integrable as a function of the t variable. First of all, by combining Assumption 2$-(v)$ and Lemma A.2, we can prove that there exists constants $C_1, \bar{L}_3 > 0$ (depending on ρ) such that

$$\begin{aligned}\left|\nabla_\theta \mathscr{F}\left(t, \Phi^\theta_{(0,t)}(x), \theta(t)\right)\right| &\leq C_1(1 + |\theta(t)|), \\ \left|\nabla_\theta \mathscr{F}\left(t, \Phi^\theta_{(0,t)}(x_1), \theta(t)\right) - \nabla_\theta \mathscr{F}\left(t, \Phi^\theta_{(0,t)}(x_2), \theta(t)\right)\right| &\leq \bar{L}_3\bar{L}_2(1 + |\theta(t)|)|x_1 - x_2| \end{aligned} \tag{3.40}$$

for a.e. $t \in [0,T]$. We recall that the quantity $\bar{L}_2 > 0$ (that already appeared in (3.38)) represents the Lipschitz constant of the flow $\Phi_{(0,t)}$ with respect to the initial datum. Moreover, from Proposition A.7, it descends that

$$\begin{aligned}\left|\mathscr{R}^\theta_{(t,T)}(x)\right| &\leq C_2, \\ \left|\mathscr{R}^\theta_{(t,T)}(x_1) - \mathscr{R}^\theta_{(t,T)}(x_2)\right| &\leq \bar{L}_4|x_1 - x_2| \end{aligned} \tag{3.41}$$

for every $t \in [0,T]$, where the constants $C_2, \bar{L}_4$ both depend on ρ. Finally, owing to Assumption 3 and Proposition 2.1, we deduce

$$\begin{aligned}\left|\nabla_x \ell\left(\Phi^\theta_{(0,T)}(x), y\right)\right| &\leq C_3, \\ \left|\nabla_x \ell\left(\Phi^\theta_{(0,T)}(x_1), y_1\right) - \nabla_x \ell\left(\Phi^\theta_{(0,T)}(x_2), y_2\right)\right| &\leq \bar{L}_5\left(\bar{L}_2|x_1 - x_2| + |y_1 - y_2|\right) \end{aligned} \tag{3.42}$$

for every $x, y \in B_R(0)$, where the constants $C_3, \bar{L}_2$ and the Lipschitz constant $\bar{L}_5$ of $\nabla_x \ell$ depend, once again, on ρ. Combining (3.40), (3.41) and (3.42), we obtain that there exists a constant $\tilde{L}_\rho > 0$ such that

$$\left|\nabla_\theta J^N[t] - \nabla_\theta J[t]\right| \leq \tilde{L}_\rho(1 + |\theta(t)|) W_1\left(\mu_0^N, \mu_0\right),$$

for a.e. $t \in [0,T]$. Observing that the right-hand side is L^2-integrable in t, the previous inequality yields

$$\left\|\nabla_\theta J^N - \nabla_\theta J\right\|_{L^2} \leq \tilde{L}_\rho(1 + \rho) W_1\left(\mu_0^N, \mu_0\right),$$

and this concludes the proof. □

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

In the next result, we provide an estimate of the *generalisation error* in terms of the distance $W_1\left(\mu_0^N, \mu_0\right)$. In this case, the important assumption is that there exists a sequence $\left(\theta^{*,N}\right)_{N\geq 1}$ of local minimisers of the functionals $(J^N)_{N\geq 1}$, and that it is contained in a region where $(J^N)_{N\geq 1}$ are uniformly strongly convex.

Theorem 3.9. *Under the same notations and hypotheses as in Proposition 3.8, let us further assume that the functional J admits a local minimiser θ^* and, similarly, that, for every $N \geq 1$, $\theta^{*,N}$ is a local minimiser for J^N. Moreover, we require that there exists a radius $\rho > 0$ such that, for every $N \geq \bar{N}$, $\theta^{*,N} \in B_\rho(\theta^*)$ and the functional J^N is $\eta-$strongly convex in $B_\rho(\theta^*)$, with $\eta > 0$. Then, there exists a constant $C > 0$ such that, for every $N \geq \bar{N}$, we have*

$$\left| \int_{\mathbb{R}^{2d}} \ell(x,y)\, d\mu_T^{\theta^{*,N}}(x,y) - \int_{\mathbb{R}^{2d}} \ell(x,y)\, d\mu_T^{\theta^*}(x,y) \right| \leq C\left(W_1\left(\mu_0^N, \mu_0\right) + \frac{1}{\sqrt{\eta}} \sqrt{W_1\left(\mu_0^N, \mu_0\right)} \right). \tag{3.43}$$

Proof. According to our assumptions, the control $\theta^{*,N} \in B_\rho(\theta^*)$ is a local minimiser for J^N, and, being J^N strongly convex on $B_\rho(\theta^*)$ for $N \geq \bar{N}$, we deduce that $\left\{\theta^{*,N}\right\} = \arg\min_{B_\rho(\theta^*)} J^N$. Furthermore, from the η-strong convexity of J^N, it follows that for every $\theta_1, \theta_2 \in B_\rho(\theta^*)$, it holds

$$\left\langle \nabla_\theta J^N(\theta_1) - \nabla_\theta J^N(\theta_2),\ \theta_1 - \theta_2 \right\rangle \geq \eta ||\theta_1 - \theta_2||_{L^2}^2.$$

According to Proposition 3.8, we can pass to the limit in the latter and deduce that

$$\left\langle \nabla_\theta J(\theta_1) - \nabla_\theta J(\theta_2),\ \theta_1 - \theta_2 \right\rangle \geq \eta ||\theta_1 - \theta_2||_{L^2}^2$$

for every $\theta_1, \theta_2 \in B_\rho(\theta^*)$. Hence, J is η-strongly convex in $B_\rho(\theta^*)$ as well, and that $\{\theta^*\} = \arg\min_{B_\rho(\theta^*)} J$. Therefore, from the η-strong convexity of J^N and J, we obtain

$$J^N(\theta^*) - J^N\left(\theta^{*,N}\right) \geq \frac{\eta}{2} \left|\left|\theta^{*,N} - \theta^*\right|\right|_{L^2}^2$$

$$J\left(\theta^{*,N}\right) - J^N(\theta^*) \geq \frac{\eta}{2} \left|\left|\theta^{*,N} - \theta^*\right|\right|_{L^2}^2.$$

Summing the last two inequalities, we deduce that

$$\eta \left|\left|\theta^{*,N} - \theta^*\right|\right|_{L^2}^2 \leq \left(J^N(\theta^*) - J(\theta^*)\right) + \left(J^N\left(\theta^{*,N}\right) - J\left(\theta^{*,N}\right)\right) \leq 2C_1 W_1\left(\mu_0^N, \mu_0\right), \tag{3.44}$$

where the second inequality follows from the local uniform convergence of Proposition 3.8. We are now in position to derive a bound on the generalisation error:

$$\begin{aligned}
\left| \int_{\mathbb{R}^{2d}} \ell(x,y) \left(d\mu_T^{\theta^{*,N}} - d\mu_T^{\theta^*} \right)(x,y) \right| &= \left| \int_{\mathbb{R}^{2d}} \ell\left(\Phi_{(0,T)}^{\theta^{*,N}}(x), y \right) d\mu_0^N(x,y) - \int_{\mathbb{R}^{2d}} \ell\left(\Phi_{(0,T)}^{\theta^*}(x), y \right) d\mu_0(x,y) \right| \\
&\leq \int_{\mathbb{R}^{2d}} \left| \ell\left(\Phi_{(0,T)}^{\theta^{*,N}}(x), y \right) - \ell\left(\Phi_{(0,T)}^{\theta^*}(x), y \right) \right| d\mu_0^N(x,y) \\
&\quad + \left| \int_{\mathbb{R}^{2d}} \ell\left(\Phi_{(0,T)}^{\theta^*}(x), y \right) \left(d\mu_0^N(x,y) - d\mu_0(x,y) \right) \right| \\
&\leq \bar{L} \sup_{x \in \mathrm{supp}\left(\mu_0^N\right)} \left| \Phi_{(0,T)}^{\theta^{*,N}}(x) - \Phi_{(0,T)}^{\theta^*}(x) \right| + \bar{L}_R W_1\left(\mu_0^N, \mu_0\right),
\end{aligned} \tag{3.45}$$

where $\bar{L}$ and $\bar{L}_R$ are constants coming from Assumption 3 and Proposition 2.1. Then, we combine Proposition 2.1 with the estimate in (3.44), in order to obtain

$$\sup_{x \in \mathrm{supp}\left(\mu_0^N\right)} \left| \Phi_{(0,T)}^{\theta^{*,N}}(x) - \Phi_{(0,T)}^{\theta^*}(x) \right| \leq C_2 \|\theta^{*,N} - \theta^*\|_{L^2} \leq C_2 \sqrt{\frac{2C_1}{\eta} W_1\left(\mu_0^N, \mu_0\right)}.$$

Finally, from the last inequality and (3.45), we deduce (3.43). □

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Remark 3.5. Since the functional $J : L^2([0,T],\mathbb{R}^m) \to \mathbb{R}$ defined in (3.6) is continuous (and, in particular, lower semi-continuous) with respect to the strong topology of L^2, the locally uniform convergence of the functionals J^N to J (see Proposition 3.8) implies that J^N is Γ-converging to J with respect to the strong topology of L^2. However, this fact is of little use, since the functionals J, J^N are not strongly coercive. On the other hand, if we equip L^2 with the weak topology, in general, the functional J is not lower semi-continuous. In our framework, the only circumstance where one can hope for Γ-convergence with respect to the weak topology corresponds to the highly-regularized scenario, i.e., when the parameter $\lambda > 0$ is sufficiently large. Therefore, in the situations of practical interest when λ is small, we cannot rely on this tool, and the crucial aspect is that the dynamics (2.1) is non-linear with respect to the control variable. Indeed, in the case of affine-control systems considered in [44], it is possible to establish Γ-convergence results in the L^2-weak topology (see [43] for an application to diffeomorphisms approximation). Finally, we report that in [46], in order to obtain the L^2-strong equi-coercivity of the functionals, the authors introduced in the cost the H^1-seminorm of the controls.

3.4. Convex regime and previous result

In order to conclude our mean-field analysis, we now compare our results with the ones obtained in the similar framework of [8], where the regularisation parameter λ was assumed to be *sufficiently large*, leading to a convex regime in the sublevel sets (see Remark 3.2). We recall below the main results presented in [8].

Theorem 3.10. *Given $T, R, R_T > 0$, and an initial datum $\mu_0 \in \mathscr{P}_c(\mathbb{R}^{2d})$ with $supp(\mu_0) \subset B_R(0)$, let us consider a terminal condition $\psi_T : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ such that $supp(\psi_T) \subset B_{R_T}(0)$ and $\psi_T(x,y) = \ell(x,y)$ $\forall x, y \in B_R(0)$. Let $\mathscr{F}$ satisfy [8, Assumptions 1-2] and $\ell \in C^2(\mathbb{R}^d \times \mathbb{R}^d, \mathbb{R})$. Assume further that $\lambda > 0$ is large enough. Then, there exists a triple $(\mu^*, \theta^*, \psi^*) \in \mathscr{C}([0,T], \mathscr{P}_c(\mathbb{R}^{2d})) \times Lip([0,T],\mathbb{R}^m) \times \mathscr{C}^1([0,T], \mathscr{C}^2_c(\mathbb{R}^{2d}))$ solution of*

$$\begin{cases} \partial_t \mu_t^*(x,y) + \nabla_x \cdot (\mathscr{F}(t,x,\theta^*(t))\mu_t^*(x,y)) = 0, & \mu_t^*|_{t=0}(x,y) = \mu_0(x,y), \\ \partial_t \psi_t^*(x,y) + \nabla_x \psi_t^*(x,y) \cdot \mathscr{F}(t,x,\theta^*(t)) = 0, & \psi_t^*|_{t=T}(x,y) = \ell(x,y), \\ \theta^{*\top}(t) = -\dfrac{1}{2\lambda} \displaystyle\int_{\mathbb{R}^{2d}} \nabla_x \psi_t^*(x,y) \cdot \nabla_\theta \mathscr{F}(t,x,\theta^*(t))\, d\mu_t^*(x,y), \end{cases} \tag{3.46}$$

where $\psi^* \in \mathscr{C}^1([0,T], \mathscr{C}^2_c(\mathbb{R}^{2d}))$ is in characteristic form. Moreover, the control solution θ^* is unique in a ball $\Gamma_C \subset L^2([0,T],\mathbb{R}^m)$ and continuously dependent on the initial datum μ_0.

We observe that the condition on $\lambda > 0$ to be large enough is crucial to obtain local convexity of the cost functional and, consequently, existence and uniqueness of the solution. However, in the present paper, we have not made assumptions on the magnitude of λ, hence, as it was already noticed in Remark 3.2, we might end up in a non-convex regime. Nevertheless, in Proposition 3.11, we show that, in the case of λ sufficiently large, the previous approach and the current one are "equivalent".

Proposition 3.11. *Under the same hypotheses as in Theorem 3.10, let $J : L^2([0,T]) \to \mathbb{R}$ be the functional defined in (3.6). Then, θ^* satisfies (3.46) if and only if it is a critical point for J.*

Proof. According to Lemma (3.3), the gradient of the functional J at $\theta \in L^2([0,T],\mathbb{R}^m)$ is defined for a.e. $t \in [0,T]$ as

$$\nabla_\theta J(\theta)[t] = \int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top \left(t, \Phi^\theta_{(0,t)}(x), \theta(t)\right) \cdot \mathscr{R}^\theta_{(t,T)}(x)^\top \cdot \nabla_x \ell^\top \left(\Phi^\theta_{(0,T)}(x), y\right) d\mu_0(x,y) + 2\lambda\theta(t).$$

Hence, if we set the previous expression equal to zero, we obtain the characterisation of the critical point

$$\theta(t) = -\frac{1}{2\lambda} \int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top \left(t, \Phi^\theta_{(0,t)}(x), \theta(t)\right) \cdot \mathscr{R}^\theta_{(t,T)}(x)^\top \cdot \nabla_x \ell^\top \left(\Phi^\theta_{(0,T)}(x), y\right) d\mu_0(x,y) \tag{3.47}$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

for a.e. $t \in [0, T]$. On the other hand, according to Theorem 3.10, the optimal θ satisfies for a.e. $t \in [0, T]$ the following

$$\theta(t) = -\frac{1}{2\lambda} \int_{\mathbb{R}^{2d}} \left(\nabla_x \psi_t(x, y) \cdot \nabla_\theta \mathscr{F}(t, x, \theta(t))\right)^\top d\mu_t(x, y)$$

$$= -\frac{1}{2\lambda} \int_{\mathbb{R}^{2d}} \nabla_\theta \mathscr{F}^\top \left(t, \Phi^\theta_{(0,t)}(x), \theta(t)\right) \cdot \nabla_x \psi_t^\top \left(\Phi^\theta_{(0,t)}(x), y\right) d\mu_0(x, y). \tag{3.48}$$

Hence, to conclude that $\nabla_\theta J = 0$ is equivalent to condition stated in Theorem 3.10, we are left to show that

$$\mathscr{R}^\theta_{(t,T)}(x)^\top \cdot \nabla_x \ell^\top \left(\Phi^\theta_{(0,T)}(x), y\right) = \nabla_x \psi_t^\top \left(\Phi^\theta_{(0,t)}(x), y\right), \tag{3.49}$$

where the operator $\mathscr{R}^\theta_{(t,T)}(x)$ is defined as the solution of (3.12). First of all, we recall that $(t, x, y) \mapsto \psi\left(t, \Phi^\theta_{(0,t)}(x), y\right)$ is defined as the characteristic solution of the second equation in (3.46) and, as such, it satisfies

$$\psi_t(x, y) = \ell\left(\Phi^\theta_{(t,T)}(x), y\right),$$

for every $t \in [0, T]$ and for every $x, y \in B_{R_T}(0)$. By taking the gradient with respect to x, we obtain that

$$\nabla_x \psi_t(x, y) = \nabla_x \ell\left(\Phi^\theta_{(t,T)}(x), y\right)\right) \cdot \nabla_x \Phi^\theta_{(t,T)}\big|_x,$$

for all $x, y \in B_{R_T}(0)$. Hence, using (3.13), we deduce that

$$\nabla_x \psi_t\left(\Phi^\theta_{(0,t)}(x), y\right) = \nabla_x \ell\left(\Phi^\theta_{(t,T)} \circ \Phi^\theta_{(0,t)}(x), y\right)\right) \cdot \nabla_x \Phi^\theta_{(t,T)}\big|_{\Phi^\theta_{(0,t)}(x)} = \nabla_x \ell\left(\Phi^\theta_{(0,T)}(x), y\right)\right) \cdot \mathscr{R}^\theta_{(t,T)}(x)$$

which proves (3.49). □

4. Algorithm

In this section, we present our training procedure, which is derived from the necessary optimality conditions related to the minimising movement scheme (see (3.31)). Since the mean-field optimal control problem as presented in (3.6) is numerically intractable (especially in high-dimension), in the practice, we always consider the functional corresponding to the finite-particles approximation (see (3.34)). For its resolution, we employ an algorithm belonging to the family of shooting methods, which consists of the forward evolution of the trajectories, the backward evolution of the adjoint variables, and the update of the controls. Variants of this method have already been employed in different works, e.g. [6, 8, 14, 26, 33], with the name of *method of successive approximations*, and they have been proven to be an alternative way of performing the training of NeurODEs for a range of tasks, including high-dimensional problems.

In our case, we start with a random guess for the control parameter $\theta_0 \in L^2([0, T], \mathbb{R}^m)$. Subsequently, we solve the necessary optimality conditions specified in equation (3.31) for a suitable $\tau > 0$ to obtain an updated control parameter θ_1. More precisely, since the last identity in (3.31) has the form $\theta_1 = \Lambda^\tau_{\theta_0}(\theta_1)$, the computation of θ_1 is performed via fixed-point iterations of the mapping $\Lambda^\tau_{\theta_0}$, which is defined as in (3.32). In this regard, we recall that $\Lambda^\tau_{\theta_0}$ is a contraction if τ is small enough. The scheme that we implemented is presented is Algorithm 1.

Remark 4.1. It is interesting to observe that, in the highly-regularized regime considered in [8], the authors managed to obtain a contractive map directly from the necessary conditions for optimality, and they did not need to consider the minimising movements scheme. This is rather natural since, when the parameter $\lambda > 0$ that tunes the L^2-penalization is large enough, the functional associated with the optimal control problem is strongly convex in the sublevel set corresponding to the control $\theta \equiv 0$, as discussed in Remark 3.2. However, as reported in [8], determining the appropriate value for λ in each application can be challenging. On the other hand, from the practitioners' perspective, dealing with high regularisation is not always desirable, since the machine learning task that the system should learn is encoded in the

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Algorithm 1: Shooting method

Data: $\{(X_0^i, Y_0^i)\}_{i=0}^N$ data with labels;
$\mathscr{F}$ controlled vector field ;
θ^0 initial guess for controls;
N_{iter} number of shooting iterations;
dt time-discretization of the interval $[0, T]$;
λ regularization parameter;
τ memory parameter;
Result: $\theta^{N_{iter}}$

$N_t \leftarrow \frac{T}{dt}$;
for $k = 1, \dots, N_{iter}$ **do**
  for $j = 1, \dots, N_t$ **do**
    for $i = 1, \dots, N$ **do**
      $X^i(t_{j+1}) \leftarrow X^i(t_j) + dt\ \mathscr{F}(t_j, X^i(t_j), \theta^k(t_j))$; // Solve forward
    end
  end
  for $i = 1, \dots, N$ **do**
    $P^i(N_t) \leftarrow -\nabla_x \ell(X^i(N_t), Y^i(0))$; //Update the co-state at the final time
  end
  for $j = N_t, \dots, 1$ **do**
    for $i = 1, \dots, N$ **do**
      $P^i(t_j) \leftarrow P^i(t_{j+1}) + dt\ P^i(t_{j+1}) \cdot \nabla_x \mathscr{F}(t_{j+1}, X^i(t_{j+1}), \theta^k(t_{j+1}))$; //Solve backward
    end
  end
  for $j = 1, \dots, N_t$ **do**
    $I^{\theta^{k+1}} \leftarrow \sum_{i=0}^N \nabla_\theta \mathscr{F}(t_j, X^i(t_j), \theta^{k+1}(t_j))^\top \cdot P^i(t_j)^\top$; // Approximate the integral (3.27)
    $\theta^{k+1}(t_j) \leftarrow \Lambda\big[-\frac{1}{1+2\lambda\tau}(\theta^k(t_j) - \frac{\tau}{N} I^{\theta^{k+1}})\big]$; // Update the control via fixed-point Λ
  end
end

final-time cost. The authors highlighted the complexity involved in selecting a regularisation parameter that is large enough to achieve contractivity while ensuring that the resulting controls are not excessively small (due to high regularisation) and of little use.

These considerations motivated us to consider a scenario where the regularisation parameter does not need to be set sufficiently large. From a numerical perspective, the parameter τ in equation (3.31) (coming from the minimising movement scheme) plays the role of the *learning rate*, and it provides the lacking amount of convexity, addressing the stability issues related to the resolution of optimal control problems in non-convex regime. These kinds of instabilities were already known in the Soviet literature on numerical optimal control (see the review paper [14]), and various solutions have been proposed to address them. For example, in [40], the authors proposed an iterative method based on the Maximum Principle and on an augmented Hamiltonian, with an approach that is somehow reminiscent of minimising movements. More recently, in the framework of NeurODEs, in [33], it was proposed another stabilisation strategy, which is different from ours since it enforces similarity between the evolution of state and co-state variables after the control update. Implicitly, the approach of [33] leads to a penalisation of significant changes in the controls. On the other hand, in our approach, this penalisation is more explicit, and it is enforced via the memory term of the minimising movement scheme. To the best of our knowledge, this is the first instance where a regularisation based on the minimising movement scheme is employed for training NeurODEs.

Remark 4.2. Although we formulate and analyse theoretically our problem within the mean-field framework, it is not advantageous to numerically solve the forward equation as a partial differential equation. In [8], various numerical methods for solving PDEs were employed and compared. However, these methods encounter limitations when applied to high-dimensional data, which is often the case in Machine Learning scenarios. Therefore, in this study, we employ a particle method to solve both the forward partial differential equation and the backward dynamics. This particle-based approach involves

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

reformulating the PDE as a system of ordinary differential equations in which particles represent mathematical collocation points that discretize the continuous fields. By employing this particle method, we address the challenges associated with high-dimensional data, enabling efficient numerical solutions for the forward and backward dynamics.

To conclude this section, we briefly present the forward and the backward systems that are solved during the execution of the method. For the sake of simplicity, we will focus on the case of an encoder. The objective is to minimise the following function:

$$J(\theta) = \frac{1}{N}\sum_{i=1}^{N} \ell\left(X^i_{\mathscr{A}_r}(T), Y^i(0)\right) + \frac{\lambda}{2}\|\theta\|_2^2, \tag{4.1}$$

where $\mathscr{A}_r$ denotes the active indices in the bottleneck, i.e. at $t_r = T$, of the state-vector $X^i(T)$. The latter denotes the encoded output at time T for the i-th particle, while $Y^i(0)$ represents the corresponding target at time 0 (which we recall is the same at time T, being $\dot{Y}^i \equiv 0$ for every $i = 1, \dots, N$). For each i-th particle and every t such that $t_j \le t \le t_{j+1}$, the forward dynamics can be described as follows:

$$\begin{cases} \dot{X}^i_{\mathscr{I}_j}(t) = 0, \\ \dot{X}^i_{\mathscr{A}_j}(t) = \mathscr{G}_j\left(t, X^i_{\mathscr{A}_j}(t), \theta(t)\right), \end{cases} \tag{4.2}$$

subject to the initial condition $X^i(0) = X^i_{\mathscr{A}_0}(0) = X^i_0 \in \mathbb{R}^d$. In the same interval $t_j \le t \le t_{j+1}$, the backward dynamics reads

$$\begin{cases} \dot{P}^i_{\mathscr{I}_j}(t) = 0, \\ \dot{P}^i_{\mathscr{A}_j}(t) = -P^i_{\mathscr{A}_j}(t) \cdot \nabla_{x_{\mathscr{A}_j}} \mathscr{G}_j\left(t, X^i_{\mathscr{A}_j}(t), \theta(t)\right), \end{cases} \tag{4.3}$$

where the final co-state is

$$P^i(T) = \begin{cases} -\partial_k \ell\left(X^i_{\mathscr{A}_r}(T), Y^i(0)\right), & \text{if } k \in \mathscr{A}_r, \\ 0, & \text{if } k \notin \mathscr{A}_r. \end{cases}$$

We notice that, for $t_j \le t \le t_{j+1}$ and every $i \in \{0, \dots, N\}$, we have

$$\mathscr{F}(t, X^i(t), \theta(t)) = \mathscr{F}\left(t, \left(X^i_{\mathscr{A}_r}, X^i_{\mathscr{I}_j}\right)(t), \theta(t)\right) = \begin{pmatrix} \mathscr{G}_j(t, X^i_{\mathscr{A}_j}(t), \theta(t)) \\ 0 \end{pmatrix}, \tag{4.4}$$

and, consequently, we deduce that

$$\nabla_x \mathscr{F}(t, X^i(t), \theta(t)) = \begin{pmatrix} \nabla_{x_{\mathscr{A}_j}} \mathscr{G}_j\left(t, X^i_{\mathscr{A}_j}(t), \theta(t)\right) & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.5}$$

where the null blocks are due to the fact that for $t_j \le t \le t_{j+1}$, $\nabla_x \mathscr{F}_k(t, x, \theta) = 0$ if $k \in \mathscr{I}_j$, and $\nabla_{x_{\mathscr{I}_j}} \mathscr{G}_j(t, x, \theta) = 0$. In the case of an Autoencoder, the structure of the forward and backward dynamics is analogous.

Remark 4.3. From the calculations reported above, it is evident that the matrices and the vectors involved in our forward and backward dynamics are quite sparse (see (4.5) and (4.4)), and that the state and co-state variables contain components that are constant in many sub-intervals (see (4.2) and (4.3)). Hence, in the practical implementation, especially when dealing with an Autoencoder, we do not actually need to double the original state variables and introduce the shadow ones, but we can simply overwrite those values and, in this way, we obtain a more memory-efficient code. A similar argument holds as well for the co-state variables. Moreover, we expect the control variable θ to have several null components during the evolution. This relates to Remark 2.1 and descends from the fact that, even though in our

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

model $\theta \in \mathbb{R}^m$ for every $t \in [0, T]$, in the internal sub-intervals $[t_j, t_{j+1}]$ only few of its components are influencing the dynamics. Hence, owing to the L^2-squared regularisation on θ, it results that if in an interval $[t_j, t_{j+1}]$ a certain component of θ is not affecting the velocity, then it is convenient to keep it null.

Before delving into the numerical experiments, it is essential to highlight the distinctions between our AutoencODEs and standard Autoencoders. A defining characteristic of AutoencODEs is the incorporation of skip connections at every layer, even in those where the dimensionality is not constant, resulting in a distinct architecture compared to traditional Autoencoders. Moreover, there is a notable difference in training methodologies. Standard Autoencoders leverage stochastic gradient descent variations like Adam for training (see e.g. [24, Chapter 8]), while our model is optimised without any stochastic perturbations of the PMP. Due to these differences in both architectures and training techniques, a straightforward comparison between the two models is currently unfeasible and remains a topic for future exploration. Indeed, it is not possible to compare the two models in terms of velocity of convergence or performance, due to the fact that the training of standard Autoencoders is highly optimised in any machine learning library, while our model is still subject to ongoing adjustments and refinements. Nonetheless, our third numerical example on the MNIST dataset underscores discernible differences between the behaviours of these models.

Finally, we mention that a possibility for enhancing our approach involves refining the training technique by considering a stochastic version of the PMP. Specifically, during each iteration of the shooting method in Algorithm 1, subsampling a data batch for updating the controls shows promising results in terms of expedited convergence and enhanced generalisation, akin to the benefits stochastic gradient descent offers for gradient methods. However, a comprehensive analysis of this method extends beyond the purpose of this paper.

5. Numerical experiments

In this section, we present a series of numerical examples to illustrate the practical application of our approach. We consider datasets of varying dimensions, ranging from low-dimensional data to a more typical Machine Learning dataset such as MNIST. Additionally, we provide justifications and insights into some of the choices made in our theoretical analysis. For instance, we examine the process of choosing the components to be deactivated during the modelling phase, and we investigate whether this hand-picked selection can lead to any issues or incorrect results. In this regard, in our first experiment concerning a classification task, we demonstrate that this a priori choice does not pose any problem, as the network effectively learns to separate the dataset into two classes before accurately classifying them. Furthermore, as we already pointed out, we have extended some of the assumptions from [8] to accommodate the use of a smooth approximation of the ReLU function. This extension is not merely a theoretical exercise, since in our second numerical example, we show how valuable it is to leverage unbounded activation functions. While both of these examples involve low-dimensional data and may not be representative of typical tasks for an Autoencoder architecture, we address this limitation in our third experiment by performing a reconstruction task on the MNIST dataset. Lastly, we present noteworthy results obtained from analysing the performance of MNIST, highlighting specific behaviours that warrant further investigation in future research.

The layers of the networks that we employ in all our experiments have the form:

$$\mathbb{R}^d \ni X = \left(X_{\mathscr{A}_j}, X_{\mathscr{I}_j}\right)^\top \mapsto \phi_n^{W,b}(X) = \left(X_{\mathscr{A}_j}, X_{\mathscr{I}_j}\right)^\top + h\Big(\sigma\left(W_{\mathscr{A}_j} \cdot X_{\mathscr{A}_j} + b_{\mathscr{A}_j}\right), 0\Big)^\top,$$

where $\mathscr{A}_j, \mathscr{I}_j$ are, respectively, the sets of active and inactive components at the layer n, $b_{\mathscr{A}_j}$ are the components of $b \in \mathbb{R}^d$ belonging to $\mathscr{A}_j$, while $W_{\mathscr{A}_j}$ is the square sub-matrix of $W \in \mathbb{R}^{d\times d}$ corresponding to the active components. Finally, the activation function σ will be specified case by case.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

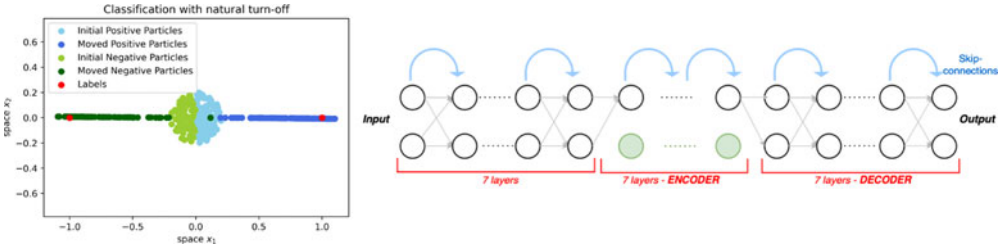

Figure 4. *Left: classification task performed when the turned off component is the natural one. Right: sketch of the AutoencODE architecture considered.*

5.1. Bidimensional classification

In our initial experiment, we concentrate on a bidimensional classification task that has been extensively described in [8]. Although this task deviates from the typical application of Autoencoders, where the objective is data reconstruction instead of classification, we believe it gives valuable insights into how our model works. The objective is to classify particles sampled from a standard Gaussian distribution in $\mathbb{R}^2$ based on the sign of their first component. Given an initial data point $x_0 \in \mathbb{R}^2$, denoted by $x_0[i]$ with $i = 1, 2$ representing its $i-$th component, we assign a positive label $+1$ to it if $x_0[1] > 0$, and a negative label -1 otherwise. To incorporate the labels into the Autoencoder framework, we augment the labels to obtain a positive label $[1, 0]$ and a negative one $[-1, 0]$. In such a way, we obtain target vectors in $\mathbb{R}^2$, i.e., with the same dimension as the input data points in the first layer.

The considered architecture is an Autoencoder comprising twenty-one layers, corresponding to $T = 2$ and $dt = 0.05$. The first seven layers maintain a constant active dimension equal to 2, followed by seven layers of active dimension 1. Finally, the last seven layers, representing the prototype of a decoder, have again constant active dimension 2, restoring the initial one. A sketch of the architecture is presented on the right side of Figure 4.

We underline that we make use of the observation presented in Remark 4.3 to construct the implemented network, and we report that we employ the hyperbolic tangent as activation function.

The next step is to determine which components to deactivate, i.e., we have to choose the sets $\mathcal{I}_j$ for $j = 1, \dots, 2r$: the natural choice is to deactivate the second component since the information, which the classification is based on, is contained in the first component (the sign) of the input data-points. Since we use the memory-saving regime of Remark 4.3, we observe that, in the decoder, the particles are 'projected' onto the x-axis, as their second component is deactivated and set equal to 0. Then, in the decoding phase, both the components have again the possibility of evolving. This particular case is illustrated on the left side of Figure 4.

Now, let us consider a scenario where the network architecture remains the same, but instead of deactivating the second component, we turn off the first component. This has the effect of "projecting" the particles onto the y-axis in the encoding phase. The results are presented in Figure 5, where an interesting effect emerges.

In the initial phase (left), where the particles can evolve in the whole space $\mathbb{R}^2$, the network is capable of rearranging the particles in order to separate them. More precisely, in this part, the relevant information for the classification (i.e., the sign of the first component), is transferred to the second component, which will not be deactivated. Therefore, once the data points are projected onto the y-axis in the bottleneck (middle), two distinct clusters are already formed, corresponding to the two classes of particles. Finally, when the full dimension is restored, the remaining task consists of moving these clusters towards the respective labels, as demonstrated in the plot on the right of Figure 5. This numerical evidence confirms that our a priori choice (even when it is very unnatural) of the components to be deactivated does not affect the network's ability to learn and classify the data. Finally, while studying this low-dimensional numerical example, we test one of the assumptions that we made in the theoretical setting. In particular,

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Table 1. *Minimum and maximum eigenvalues of the Hessian matrix across epochs.*

Epochs	0	80	160	240	320	400	480	560	640	720
Min Eig.	–1.72e-2	–1.19e-2	–1.09e-2	–8.10e-3	–3.44e-3	–6.13e-3	6.80e-4	7.11e-4	7.25e-4	7.33e-4
Max Eig.	3.78e-2	2.84e-1	7.30e-1	9.34e-1	1.11	1.18	1.22	1.25	1.26	1.27

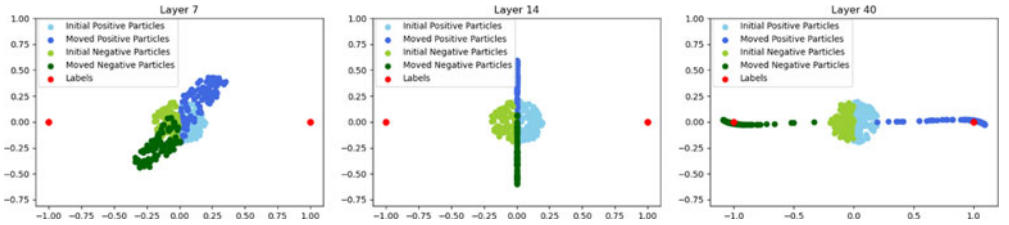

Figure 5. *Left: initial phase, i.e., separation of the data along the y-axis. Center: encoding phase, i.e., only the second component is active. Right: decoding phase and classification result after the 'unnatural turn off'. Notice that, for a nice clustering of the classified data, we have increased the number of layers from* 20 *to* 40*. However, we report that the network accomplishes the task even if we use the same structure as in Figure* 4*.*

we want to check if it is reasonable to assume that the cost landscape is convex around local minima, as assumed in Theorem 3.9. In Table 1, we report the smallest and highest eigenvalues of the Hessian matrix of the loss function recorded during the training process, i.e., starting from a random initial guess, until the convergence to an optimal solution.

5.2. Parabola reconstruction

In our second numerical experiment, we focus on the task of reconstructing a two-dimensional parabola. To achieve this, we sample points from the parabolic curve and we use them as the initial data for our network. The network architecture consists of a first block of seven layers with active dimension 2, followed by seven additional layers with active dimension 1. Together, these two blocks represent the encoding phase in which the set of active components are $\mathscr{A}_j = \{0\}$ for $j = 7, \ldots, 14$. Similarly, as in the previous example, the points at the 7-th layer are "projected" onto the $x-$axis, and for the six subsequent layers, they are constrained to stay in this subspace. After the 14-th layer, the original active dimension is restored, and the particles can move in the whole space $\mathbb{R}^2$, aiming at reaching their original positions. Despite the low dimensionality of this task, it provides an interesting application that allows us to observe the distinct phases of our mode, which are presented in Figure 6.

Notably, in the initial seven layers, the particles show quite tiny movements (top left of Figure 6). This is because the relevant information to reconstruct the position is encoded in the first component, which is kept active in the bottleneck. On the other hand, if in the encoder we chose to deactivate the first component instead of the second one, we would expect that the points need to move considerably before the projection takes place, as was the case in the previous classification task. During the second phase (top right of Figure 6), the particles separate along the x-axis, preparing for the final decoding phase, which proves to be the most challenging to learn (depicted in the bottom left of Figure 6). Based on our theoretical knowledge and the results from initial experiments, we attempt to improve the performance of the AutoencODE network by modifying its structure. One possible approach is to design the network in a way that allows more time for the particles to evolve during the decoding phase while reducing the time spent in the initial and bottleneck phases. Indeed, we try to use 40 layers instead of 20, and most of the new ones are allocated in the decoding phase. The result is illustrated in the bottom right of

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

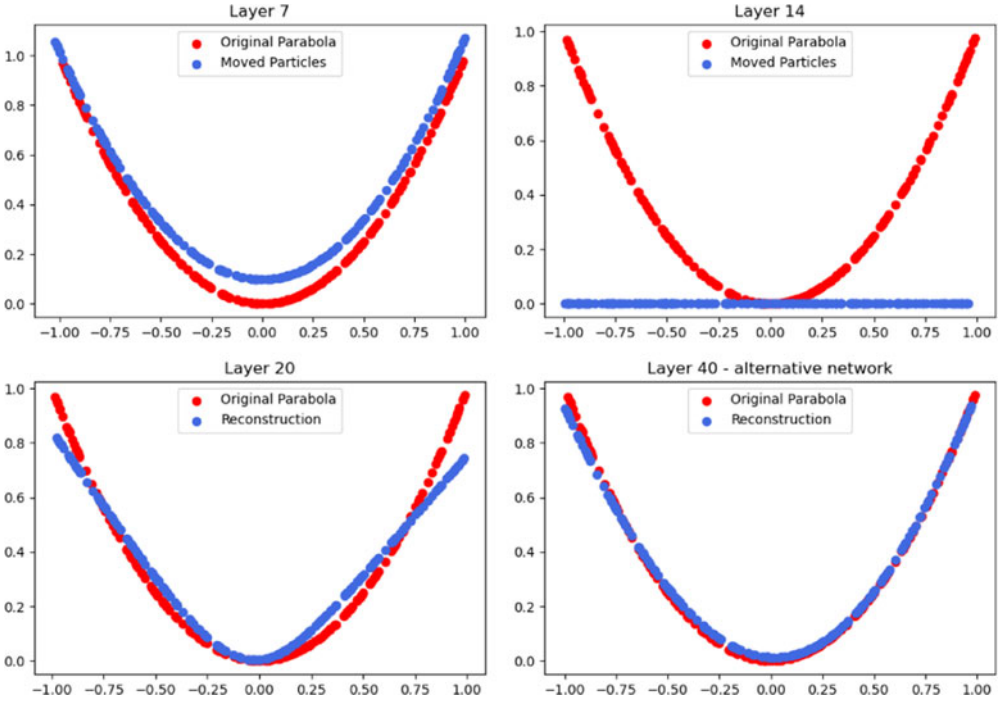

Figure 6. *Top left: initial phase. Top rights: encoding phase. Bottom left: decoding phase. Bottom right: network's reconstruction with alternative architecture.*

Figure 6, where we observe that changing the network's structure has a significant positive impact on the reconstruction quality, leading to better results. This result is inspired by the heuristic observation that the particles 'do not need to move' in the first two phases. On this point, a more theoretical analysis of the network's structure will be further discussed in the next paragraph, where we perform a sanity check, and relate the need for extra layers to the Lipschitz constant of the trained network.

This experiment highlights an important observation regarding the choice of activation functions. Specifically, it becomes evident that certain bounded activation functions, such as the hyperbolic tangent, are inadequate for moving the particles back to their original positions during the decoding phase. The bounded nature of these activation functions limits their ability to move a sufficiently large range of values, which can lead to the points getting stuck at suboptimal positions and failing to reconstruct the parabolic curve accurately. To overcome this limitation and achieve successful reconstruction, it is necessary to employ unbounded activation functions that allow for a wider range of values, in particular the well-known Leaky Relu function. An advantage of our approach is that our theory permits the use of smooth approximations for well-known activation functions, such as the Leaky ReLU (2.4). Specifically, we employ the following smooth approximation of the Leaky ReLU function:

$$\sigma_{smooth}(x) = \alpha x + (1-\alpha)\frac{1}{s}\log\left(1+e^{sx}\right), \tag{5.1}$$

where s approaching infinity ensures convergence to the original Leaky ReLU function. While alternative approximations are available, we employed (5.1) in our study. This observation emphasises the importance of considering the characteristics and properties of activation functions when designing and training neural networks, and it motivates our goal in this work to encompass unbounded activation functions in our working assumptions.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

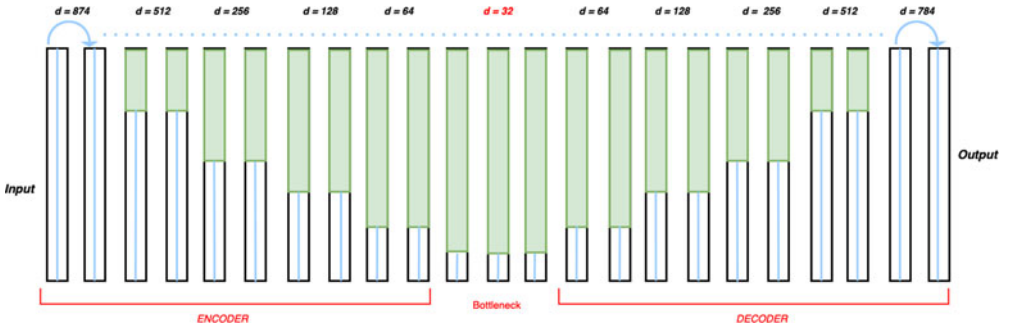

Figure 7. *Architecture used for the MNIST reconstruction task. The inactive nodes are marked in green.*

5.3. MNIST reconstruction

In this experiment, we apply the AutoencODE architecture and our training method to the task of reconstructing images from the MNIST dataset. The MNIST dataset contains 70, 000 greyscale images of handwritten digits ranging from zero to nine. Each image has a size of 28×28 pixels and has been normalised. This dataset is commonly used as a benchmark for image classification tasks or for evaluating image recognition and reconstruction algorithms. However, our objective in this experiment is not to compare our reconstruction error with state-of-the-art results, but rather to demonstrate the applicability of our method to high-dimensional data, and to highlight interesting phenomena that we encounter. In general, when performing an autoencoder reconstruction task, the goal is to learn a lower-dimensional representation of the data that captures its essential features. On the other hand, determining the dimension of the lower-dimensional representation, often referred to as the *latent dimension*, requires setting a hyperparameter, i.e., the width of the bottleneck's layers, which might depend on the specific application.

We now discuss the architecture we employed and the choice we made for the latent dimension. Our network consists of twenty-three layers, with the first ten layers serving as encoder, where the dimension of the layers is gradually reduced from the initial value $d_0 = 784$ to a latent dimension of $d_r = 32$. Then, this latent dimension is kept in the bottleneck for three layers, and the last ten layers act as decoder, and, symmetrically to the encoder, it increases the width of the layers from 32 back to $d_{2r} = 784$. Finally, for each layer, we employ a smooth version of the Leaky Relu, see (5.1), as activation function. The architecture is visualised in Figure 7, while the achieved reconstruction results are presented in Figure 8. We observe that, once again, we made use of Remark 4.3 for the implementation of the AutoencoODE-based model.

Latent dimensionality in the bottleneck.

One of the first findings that we observe in our experiments pertains to the latent dimension of the network and to the intrinsic dimension of the dataset. The problem of determining the intrinsic dimension has been object of previous studies such as [16, 17, 47], where it was estimated to be approximately equal to 13 in the case of MNIST dataset. On this interesting topic, we also report the paper [32], where a maximum likelihood estimator was proposed and datasets of images were considered, and the recent contribution [35]. Finally, the model of the *hidden manifold* has been formulated and studied in [23].

Notably, our network exhibits an interesting characteristic in which, starting from the initial guess of weights and biases initialised at 0, the training process automatically identifies an intrinsic dimensionality of 13. Namely, we observe that the latent vectors of dimension 32 corresponding to each image in the dataset are sparse vectors with 13 non-zero components, forming a consistent support across all latent vectors derived from the original images. To further analyse this phenomenon, we compute the means of all the latent vectors for each digit and compare them, as depicted in the left and middle of

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

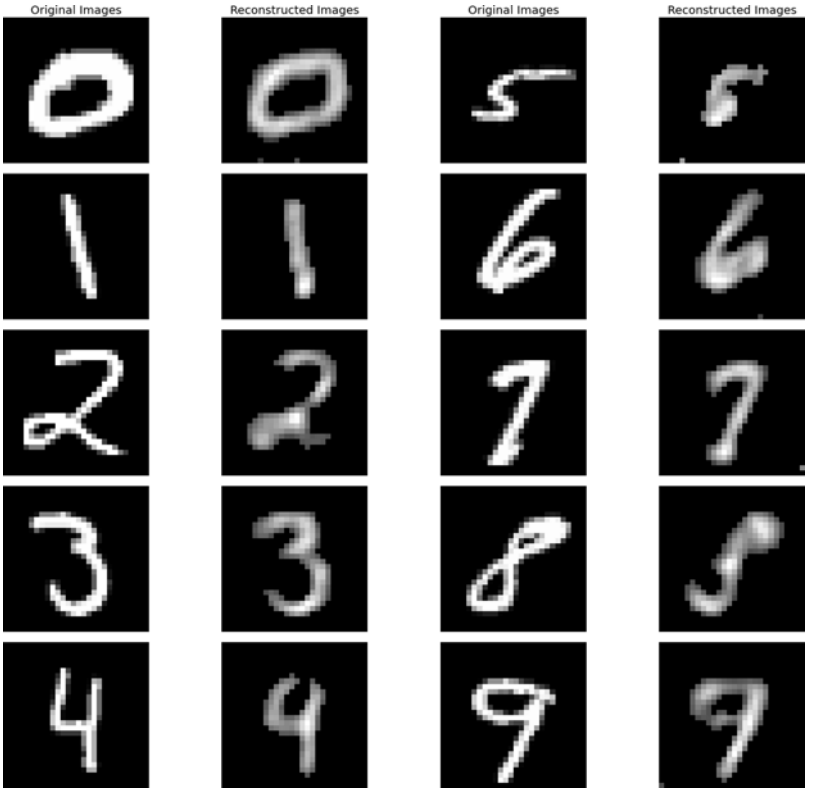

Figure 8. *Reconstruction of some numbers achieved by AutoencODE.*

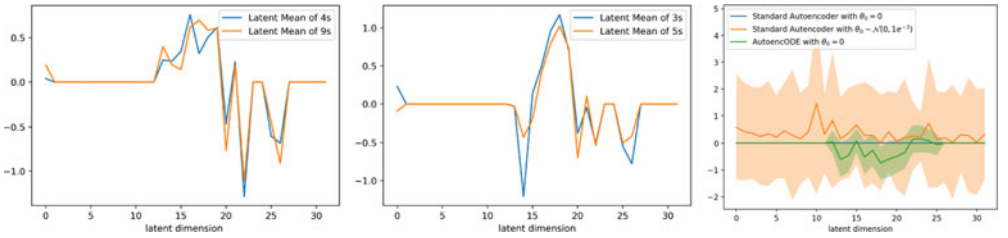

Figure 9. *Left: comparing two similar latent means. Center: again two similar latent means. Right: mean and standard deviation of the encoded vectors in the bottleneck.*

Figure 9. These mean vectors always have exactly the same support of dimension 13, and, interestingly, we observe that digits that share similar handwritten shapes, such as the numbers 4 and 9 or digits 3 and 5, actually have latent means that are close to each other. Additionally, we explore the generative capabilities of our network by allowing the latent means to evolve through the decoding phase, aiming to generate new images consistent with the mean vector. This intriguing behaviour of our network warrants further investigation into its ability to detect the intrinsic dimension of the input data, and into the exploration of its generative potential. Previous studies have demonstrated that the ability of neural networks to converge to simpler solutions is significantly influenced by the initial parameter values (see e.g. [15]). Indeed, in our case, we have observed that this phenomenon only occurs when initialising the parameters with zeros.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Remark 5.1. If we compare our results with those of a standard Autoencoder, we observe the absence of the emergence of the latent dimensionality within the bottleneck. For the comparison, we employed a classical width-varying Autoencoder (i.e., without residual-like skipping connections), with the same number of layers and trainable parameters as the AutoencODE represented in Figure 7. We used the Leaky ReLu as the activation function, in every layer. When training standard Autoencoders, it is well-known that constant initialisation schemes perform poorly, leading to uniform gradients and neurones converging to exactly the same features during training. Conversely, when starting with low-magnitude values of parameters, e.g., normally distributed with a standard deviation of 10^{-3}, the training is successful in terms of reconstruction, but the resultant encoded vectors lack sparsity. In AutoencODEs, the combination of our novel architecture with our training methodology allows us to optimise the cost even with a zero initialisation. Remarkably, this not only proves to be effective but also facilitates the representation of data within the bottleneck through sparse vectors. These vectors exhibit a support size, which seems to be intricately linked to the dataset's true dimensionality.

These considerations are illustrated on the right of Figure 9, where we depict both the mean and the associated standard deviation (visualized as a shadow surrounding the mean) of the encoded vectors obtained after training with different initializations of the parameters. Indeed, we observed that in standard Autoencoders initialised with zero, the resulting network fails to adequately reconstruct the data. Instead, it predominantly learns a 'mean' reconstruction, making the standard deviation not visibly apparent. The results achieved with a small initialisation effectively reconstruct the data. However, they fall short of producing sparse vectors in the bottleneck, a feature successfully achieved by AutoencODEs in the third case.

Sanity check of the network's architecture.

An advantage of interpreting neural networks as discrete approximations of dynamical systems is that we can make use of typical results of numerical resolutions of ODEs in order to better analyse our results. Indeed, we notice that, according to well-known results, in order to solve a generic ODEs, we need to take as discretization step-size dt a value smaller than the inverse of the Lipschitz constant of the vector field driving the dynamics. We recall that the quantity dt is related to the number of layers of the network through the relation $n_{\text{layers}} = \frac{T}{dt}$, where T is the right-extreme of the evolution interval $[0, T]$.

In our case, we choose *a priori* the amplitude of dt, we train the network and once we have computed θ^*, we can compare *a posteriori* the discretization step-size chosen at the beginning with the quantity $\Delta = \frac{1}{Lip(\mathscr{F}(t,X,\theta^*)}$ for each time-node t and every datum x.

In Figure 10, we show the time discretization dt in orange and in blue the quantity Δ, for the case of a wrongly constructed autoencoder (on the left) and the correct one (on the right). From these plots, we can perform a 'sanity check' and we can make sure that the number of layers that we chose is sufficient to solve the task. Indeed, in the wrong autoencoder on the left, we see that in the last layer, the quantity Δ is smaller than dt, and this violates the condition that guarantees the stability of the explicit Euler discretization.

Indeed, the introduction of two symmetric layers to the network (corresponding to the plot on the right of Figure 10) allows the network to satisfy everywhere the relation $\Delta > dt$. Moreover, we also notice that during the encoding phase, the inverse of the Lipschitz constant of $\mathscr{F}$ is quite high, which means that the vector field does not need to move a lot of the points. This suggests that we could get rid of some of the layers in the encoder and only keep the necessary ones, i.e., the ones in the decoder where Δ is small and a finer discretization step size is required. We report that this last observation is consistent with the results recently obtained in [11]. Finally, we also draw attention to the work [45], which shares a similar spirit with our experiments, since the Lipschitz constant of the layers is the main subject of investigation. In their study, the authors employ classical results on the numerical integration of ordinary differential equations in order to understand how to constrain the weights of the network with the aim of designing stable architectures. This approach leads to networks with non-expansive properties, which is highly

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

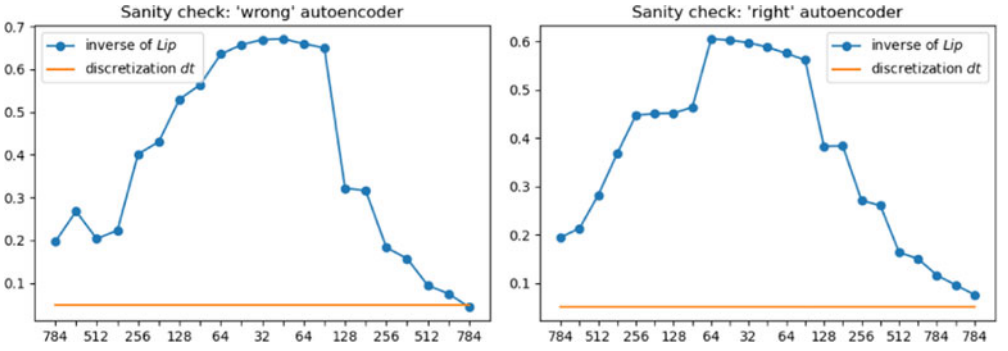

Figure 10. *Left: wrong autoencoder detected with the analysis of* Δ*. Right: correct version of the same autoencoder.*

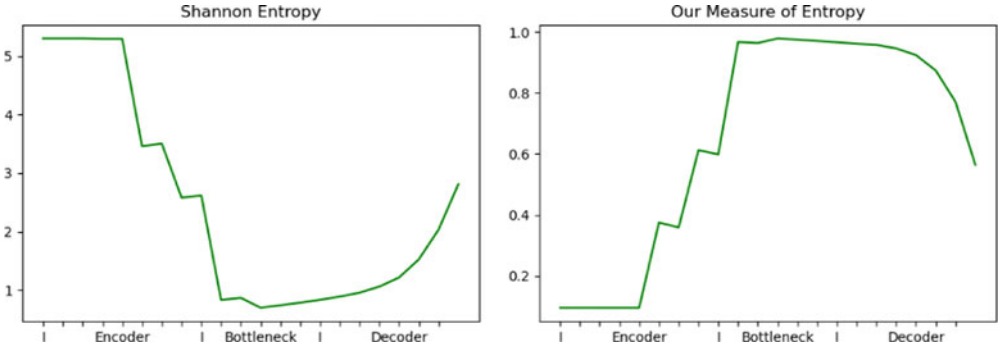

Figure 11. *Left: Shannon entropy across layers. Right: our measure of entropy across layers.*

advantageous for mitigating instabilities in various scenarios, such as testing adversarial examples [25], training generative adversarial networks [3] or solving inverse problems using deep learning.

Entropy across layers.

We present our first experiments on the study of the information propagation within the network, where some intriguing results appear. This phenomenon is illustrated in Figure 11, where we examine the entropy across the layers after the network has been trained. We introduce two different measures of entropy, depicted in the two graphs of the figure. In first place, we consider the well-known Shannon entropy, denoted as $H(E)$, which quantifies the information content of a discrete random variable E, distributed according to a discrete probability measure $p{:}\Omega \to [0,1]$ such that $p(e) = p(E = e)$. The Shannon entropy is computed as follows:

$$H(E) = \mathbb{E}\big[-\log\left(p(E)\right)\big] = \sum_{e \in E} -p(e)\log\left(p(e)\right)$$

In our context, the random variable of interest is $E = \sum_{j=1}^{N} \mathbb{1}_{|X_0^i - X_0^j| \leq \epsilon}$, where X_0^i represents a generic image from the MNIST dataset. Additionally, we introduce another measure of entropy, denoted as $\mathscr{E}$, which quantifies the probability that the dataset can be partitioned into ten clusters corresponding to the ten different digits. This quantity has been introduced in [21] and it is defined as

$$\mathscr{E} = \mathbb{P}\left(X \in \bigcup_{i=1}^{k} B_\varepsilon\left(X_0^i\right)\right),$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

where $\varepsilon > 0$ is a small radius, and $X_0^1, \ldots, X_0^k$ are samplings from the dataset. Figure 11 suggests the existence of a distinct pattern in the variation of information entropy across the layers, which offers a hint for further investigations.

Let us first focus on the Shannon entropy: as the layers' dimensionality decreases in the encoding phase, there is an expected decrease of entropy, reflecting the compression and reduction of information in the lower-dimensional representation. The bottleneck layer, where the dimension is kept constant, represents a critical point where the entropy reaches a minimum. This indicates that the information content is highly concentrated and compressed in this latent space. Then, during the decoding phase, the Shannon entropy does not revert to its initial value but instead exhibits a slower increase. This behaviour suggests that the network retains some of the learned structure and information from the bottleneck layer. Something similar happens for the second measure of entropy: at the beginning, the data is unlikely to be highly clustered, since two distinct images of the same digit may be quite distant from the other. In the inner layers, this probability increases until it reaches its maximum (rather close to 1) in the bottleneck, where the data can then be fully partitioned into clusters of radius ϵ. As for the Shannon entropy, the information from the bottleneck layer is retained during the decoding phase, which is why the entropy remains constant for a while and then decreases back in a slower manner.

It is worth noticing that in both cases, the entropy does not fully return to its initial level. This might be attributed to the phenomenon of mode collapse, where the network fails to capture the full variability in the input data and instead produces similar outputs for different inputs, hence inducing some sort of *implicit bias*. Mode collapse is often considered undesirable in generative models, as it hinders the ability to generate diverse and realistic samples. However, in the context of understanding data structure and performing clustering, the network's capability to capture the main modes or clusters of the data can be seen as a positive aspect. The network learns to extract salient features and represent the data in a compact and informative manner, enabling tasks such as clustering and classification. Further investigation is needed to explore the relationship between the observed entropy patterns, mode collapse and the overall performance of the network on different tasks.

6. Conclusion

We have extended the well-established continuous-time model for Neural ODEs to the case of networks with layers of varying width, by introducing a properly controlled dynamical system with explicit dependence on the time variable. For the analysis of such controlled systems, we have employed a mean-field control framework to investigate the limiting case when the size of the training dataset tends to infinity, and we have obtained results on the generalisation capabilities of these networks. In addition to what was already available in the literature, in our analysis, we do not need to require the presence of high Tikhonov regularisation. Moreover, we have developed a training method tailored to this kind of network that is based on the necessary conditions formulated in the theoretical sections. This training procedure has been applied in the experiments to solve various tasks, and the promising results have been collected together. We have also noticed some interesting arising behaviours (dimensionality detection) that deserve further investigation in future work.

Acknowledgements. The authors would like to thank Prof. Giuseppe Savaré for the fruitful discussions during his permanence in Munich. Moreover, the authors are grateful to Dr. Oleh Melnyk for the suggestion on the extension of the dynamical model to the U-net architecture.

Financial support. This work has been funded by the German Federal Ministry of Education and Research and the Bavarian State Ministry for Science and the Arts through the Munich Center for Machine Learning (award number 511-75228-4/6-MCML23B). C.C. and M.F. also acknowledge the partial support of the DFG Project "Implicit Bias and Low Complexity Networks" within the DFG SPP 2298 "Theoretical Foundations of Deep Learning". A.S. acknowledges the partial support from INdAM-GNAMPA. The Open Access funding for this publication has been provided by Technical University of Munich within an agreement between Technical University of Munich and Cambridge University Press.

Competing interests. The authors declare none.

References

[1] Ambrosio, L., Caffarelli, L., Crandall, M. G., Evans, L. C., Fusco, N. & Ambrosio, L. (2008) Transport equation and cauchy problem for non-smooth vector fields. In: *Calculus of Variations and Nonlinear Partial Differential Equations: With a Historical Overview by Elvira Mascolo*, pp. 1–41.

[2] Ambrosio, L., Gigli, N. & Savaré, G. (2005) *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*, Basel: Birkhäuser.

[3] Arjovsky, M., Chintala, S. & Bottou, L. (2017) Wasserstein generative adversarial networks. In: *International Conference on Machine Learning*, PMLR, pp. 214–223.

[4] Aubin, J.-P. (2010) *Viability Theory*, Birkhäuser, Boston, MA.

[5] Bengio, Y., Simard, P. & Frasconi, P. (1994) Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Networks* **5**(2), 157–166.

[6] Benning, M., Celledoni, E., Ehrhardt, M. J., Owren, B. & Schönlieb, C.-B. (2019) Deep learning as optimal control problems: models and numerical methods. *J. Comput. Dyn.* **6**(2), 6–198.

[7] Bonnet, B., Cipriani, C., Fornasier, M. & Huang, H. (2023) A measure theoretical approach to the mean-field maximum principle for training neurodes. *Nonlinear Anal.* **227**, 113161.

[8] Bonnet, B. & Frankowska, H. (2022) Viability and exponentially stable trajectories for differential inclusions in wasserstein spaces. In:*2022 IEEE 61st Conference on Decision and Control (CDC)*, IEEE, pp. 5086–5091.

[9] Bonnet-Weill, B. & Frankowska, H. (2023) On the viability and invariance of proper sets under continuity inclusions in wasserstein spaces.

[10] Bressan, A. & Piccoli, B. (2007). *Introduction to the Mathematical Theory of Control. 1*, American Institute of Mathematical Sciences, Springfield.

[11] Bungert, L., Roith, T., Tenbrinck, D. & Burger, M. (2021) Neural architecture search via bregman iterations. arXiv preprint arXiv: 2106.02479.

[12] Chang, B., Meng, L., Haber, E., Ruthotto, L., Begert, D. & Holtham, E. (2018) Reversible architectures for arbitrarily deep residual neural networks. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, p. 32.

[13] Chen, R. T., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. (2018) Neural ordinary differential equations. In: *Advances in Neural Information Processing Systems, 31*.

[14] Chernousko, F. & Lyubushin, A. (1982) Method of successive approximations for solution of optimal control problems. *Optim. Control Appl. Methods* **3**(2), 101–114.

[15] Chou, H.-H., Maly, J. & Rauhut, H. (2023) More is less: inducing sparsity via overparameterization. *Inf. Inference J. IMA* **12**(3), iaad012.

[16] Costa, J. A. & Hero, A. O. (2004) Learning intrinsic dimension and intrinsic entropy of high-dimensional datasets. In: *2004 12th European Signal Processing Conference*, pp. 369–372.

[17] Denti, F., Doimo, D., Laio, A. & Mira, A. (2022) The generalized ratios intrinsic dimension estimator. *Sci. Rep-UK* **12**(1), 20005.

[18] Esteve, C., Geshkovski, B., Pighin, D. & Zuazua, E. (2020) Large-time asymptotics in deep learning. arXiv preprint arXiv: 2008.02491.

[19] Fornasier, M., Heid, P. & Sodini, G. (2023) In preparation.

[20] Geshkovski, B. & Zuazua, E. (2022) Turnpike in optimal control of pdes, resnets, and beyond. *Acta Numer.* **31**, 135–263.

[21] Goldt, S., Mézard, M., Krzakala, F. & Zdeborová, L. (2020) Modeling the influence of data structure on learning in neural networks: the hidden manifold model. *Phys. Rev. X* **10**(4), 041044.

[22] Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep Learning*, MIT Press. *Available at* http://www.deeplearningbook.org.

[23] Goodfellow, I. J., Shlens, J. & Szegedy, C. (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[24] Haber, E. & Ruthotto, L. (2017) Stable architectures for deep neural networks. *Inverse Probl.* **34**(1), 014004.

[25] Hale, J. K. (1969) *Ordinary Differential Equations*, Wiley-Interscience, New York.

[26] He, K. & Sun, J. (2015) Convolutional neural networks at constrained time cost. In:*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5353–5360.

[27] He, K., Zhang, X., Ren, S. & Sun, J. (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

[28] Hinton, G. E., Osindero, S. & Teh, Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural Comput.* **18**(7), 1527–1554.

[29] Ladas, G. E. & Lakshmikantham, V. (1972) Differential Equations in Abstract Spaces, Vol. **85**, Academic Press, New York.

[30] Levina, E. & Bickel, P. (2004) Maximum likelihood estimation of intrinsic dimension. In: *Advances in Neural Information Processing Systems, 17*.

[31] Li, Q., Chen, L. & Tai, C. (2018) Maximum principle based algorithms for deep learning. *J. Mach. Learn. Res.* **18**, 1–29.

[32] Liu, H. & Markowich, P. (2020) Selection dynamics for deep neural networks. *J. Differ. Equations* **269**(12), 11540–11574.

[33] Macocco, I., Glielmo, A., Grilli, J. & Laio, A. (2023) Intrinsic dimension estimation for discrete metrics. *Phys. Rev. Lett.* **130**(6), 067401.

[34] Mei, S., Misiakiewicz, T. & Montanari, A. (2019) Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In: *Conference on Learning Theory*, PMLR, pp. 2388–2464.
[35] Pontryagin, L. S. (1987). *Mathematical Theory of Optimal Processes*, CRC Press, London.
[36] Ronneberger, O., Fischer, P. & Brox, T. (2015) U-net: convolutional networks for biomedical image segmentation. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pp. 234–241.
[37] Rumelhart, D. E., Hinton, G. E. & Williams, R. J. (1986) Learning representations by back-propagating errors. *Nature* **323**, 533–536.
[38] Sakawa, Y. & Shindo, Y. (1980) On global convergence of an algorithm for optimal control. *IEEE Trans. Autom. Control* **25**(6), 25–1153.
[39] Santambrogio, F. (2017) {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bull. Math. Sci.* **7**, 87–154.
[40] Scagliotti, A. (2022) A gradient flow equation for optimal control problems with end-point cost. *J. Dyn. Control Syst***29**, 521–568.
[41] Scagliotti, A. (2023) Optimal control of ensembles of dynamical systems. *ESAIM Control Optim. Calc. Var.* **29**, 22.
[42] Scagliotti, A. (2023) Deep learning approximation of diffeomorphisms via linear-control systems. *Math. Control Relat. Fields* **13**(3), 1226–1257.
[43] Sherry, F., Celledoni, E., Ehrhardt, M. J., Murari, D., Owren, B. & Schönlieb, C.-B. (2023) Designing stable neural networks using convex analysis and odes. arXiv preprint arXiv:2306.17332.
[44] Thorpe, M. & van Gennip, Y. (2023) Deep limits of residual neural networks. *Res. Math. Sci.* **10**(1), 6.
[45] W., E. (2017) A proposal on machine learning via dynamical systems. *Commun. Math. Stat.* **1**(5), 1–11.
[46] W., E., Han, J. & Li, Q. (2019) A mean-field optimal control formulation of deep learning. *Res. Math. Sci.* **6**(1), 1–41.
[47] Zheng, Y., He, T. Y. Qiu, and D. P. Wipf, *Learning manifold dimensions with conditional variational autoencoders*. Advances in Neural Information Processing Systems, **35**, (2022), 34709-34721.

A. Appendix

Lemma A.1 (Boundedness of trajectories). *Let us consider the controlled system*

$$\dot{x} = \mathscr{F}(t, x, \theta), \quad x(0) = x_0,$$

where $\mathscr{F} : [0, T] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ *satisfies Assumptions* 1, *and* $\theta \in L^2([0, T], \mathbb{R}^m)$. *Then, for every* $R > 0$ *and any* $x_0 \in B_R(0)$, *we have that* $x(t) \in B_{\bar{R}}(0)$ *for every* $t \in [0, T]$, *where* $\bar{R} = (R + L_R(1 + \|\theta\|_{L^1}))e^{L_R(1+\|\theta\|_{L^1})}$.

Proof. According to Assumption 1−(*ii*) on $\mathscr{F}$, the trajectories can be bounded as follows:

$$|x(t)| \le |x_0| + \int_0^t |\mathscr{F}(s, x(s), \theta(s))|\, ds \le |x_0| + L_R \int_0^t (1 + |x(s)|)(1 + |\theta(s)|)\, ds$$

for every $t \in [0, T]$. Using Gronwall's lemma, it follows that

$$|x(t)| \le (|x_0| + L_R\,(1 + \|\theta\|_{L^1}))\, e^{L_R\left(1+\|\theta\|_{L^1}\right)}.$$ □

Lemma A.2 (Flow's dependency on initial datum). *For every* $t \in [0, T]$, *let us consider the flow mapping* $\Phi^{\theta}_{(0,t)} : \mathbb{R}^d \to \mathbb{R}^d$ *defined in* (2.2) *and driven by the control* $\theta \in L^2([0, T], \mathbb{R}^m)$. *Let us assume that the controlled dynamics* $\mathscr{F} : [0, T] \times \mathbb{R}^d \times \mathbb{R}^m \to \mathbb{R}^d$ *satisfies Assumption* 1. *Then, for every* $R > 0$, *and every* $x_1, x_2 \in B_R(0)$, *it follows that*

$$\left|\Phi^{\theta}_{(0,t)}(x_1) - \Phi^{\theta}_{(0,t)}(x_2)\right| \le e^{L_{\bar{R}}(1+\|\theta\|_{L1})}|x_1 - x_2|,$$

where $\bar{R}$ *is defined as in Lemma* A.1, *and* $L_{\bar{R}}$ *is prescribed by Assumption* 1*-(ii).*

Proof. Let us denote with $t \mapsto x_1(t), t \mapsto x_2(t)$ the solutions of (2.1) driven by θ and starting, respectively, from $x_1(0) = x_1, x_2(0) = x_2$. Then, for every $t \in [0, T]$, we have

$$|x_1(t) - x_2(t)| \le |x_1 - x_2| + \int_0^t |\mathscr{F}(s, x_1(s), \theta(s)) - \mathscr{F}(s, x_2(s), \theta(s))|\, ds$$

$$\le |x_1 - x_2| + L_{\bar{R}} \int_0^t (1 + |\theta(s)|)|x_1(s) - x_2(s))|\, ds,$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

by using Assumption 1−(*ii*). As before, the statement follows from Gronwall's Lemma. □

Lemma A.3 (Flow's dependency on time). *Under the same assumptions and notations as in Lemma A.2, for every $R > 0$, for every $x \in B_R(0)$ and for every $\theta \in L^2([0,T],\mathbb{R}^m)$, we have that*

$$\left|\Phi^{\theta}_{(0,t_2)}(x) - \Phi^{\theta}_{(0,t_1)}(x)\right| \leq L_{\bar{R}}(1+\bar{R})(1+\|\theta\|_{L^2})|t_2 - t_1|^{\frac{1}{2}}$$

for every $0 \leq t_1 < t_2 \leq T$, where $\bar{R}$ is defined as in Lemma A.1, and $L_{\bar{R}}$ is prescribed by Assumption 1-(ii). Moreover, if $\theta \in L^2([0,T],\mathbb{R}^m) \cap L^\infty([0,T],\mathbb{R}^m)$, then, for every $0 \leq t_1 < t_2 \leq T$, it holds:

$$\left|\mathbf{\Phi}^{\theta}_{(0,t_2)}(x) - \mathbf{\Phi}^{\theta}_{(0,t_1)}(x)\right| \leq L_{\bar{R}}(1+\bar{R})(1+\|\theta\|_{L^2})|t_2 - t_1|.$$

Proof. If we denote by $t \mapsto x(t)$ the solution of (2.1) driven by the control θ, then

$$\left|x(t_2) - x(t_1)\right| \leq \int_{t_1}^{t_2} \left|\mathscr{F}(s, x(s), \theta(s)\right| ds \leq \int_{t_1}^{t_2} L_{\bar{R}}(1+\bar{R})(1+|\theta(s)|)\, ds.$$

The thesis follows by using Cauchy-Schwarz for $\theta \in L^2$, or from basic estimates if $\theta \in L^\infty$. □

Lemma A.4 (Flow's dependency on controls). *For every $t \in [0,T]$, let $\Phi^{\theta_1}_{(0,t)}, \Phi^{\theta_2}_{(0,t)} : \mathbb{R}^d \to \mathbb{R}^d$ be the flows defined in (2.2) and driven, respectively, by $\theta_1, \theta_2 \in L^2([0,T],\mathbb{R}^m)$. Let us assume that the controlled dynamics $\mathscr{F} : [0,T] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ satisfies Assumption 1. Then, for every $R > 0$ and for every $x \in B_R(0)$, it holds that*

$$\left|\Phi^{\theta_1}_{(0,t)}(x) - \Phi^{\theta_2}_{(0,t)}(x)\right| \leq L_{\bar{R}}\,(1 + \|\theta_1\|_{L^2} + \|\theta_2\|_{L^2})\, e^{L_{\bar{R}}(1+\|\theta_1\|_{L^1})}\, \|\theta_1 - \theta_2\|_{L^2},$$

where $\bar{R}$ is defined as in Lemma A.1, and $L_{\bar{R}}$ is prescribed by Assumption 1-(ii).

Proof. By using Assumption 1−(*ii*), (*iii*) and the triangle inequality, we obtain that

$$\begin{aligned}
\left|\Phi^{\theta_1}_{(0,t)}(x) - \Phi^{\theta_2}_{(0,t)}(x)\right| &\leq \int_0^t |\mathscr{F}(s, x_1(s), \theta_1(s)) - \mathscr{F}(s, x_2(s), \theta_2(s))|\, ds \\
&\leq \int_0^t |\mathscr{F}(s, x_1(s), \theta_1(s)) - \mathscr{F}(s, x_2(s), \theta_1(s))|\, ds \\
&\quad + \int_0^t |\mathscr{F}(s, x_2(s), \theta_1(s)) - \mathscr{F}(s, x_2(s), \theta_2(s))|\, ds \\
&\leq L_{\bar{R}} \int_0^t (1+\theta_1(s))|x_1(s) - x_2(s)|\, ds + L_{\bar{R}}\,(1 + \|\theta_1\|_{L^2} + \|\theta_2\|_{L^2})\, \|\theta_1 - \theta_2\|_{L^2}.
\end{aligned}$$

The statement follows again by applying Gronwall's Lemma. □

Proposition A.5 (Differentiability with respect to trajectories perturbations). *Let us assume that the controlled dynamics $\mathscr{F}$ satisfies Assumptions 1–2. Given an admissible control $\theta \in L^2([0,T],\mathbb{R}^m)$ and a trajectory $t \mapsto x(t) = \Phi^{\theta}_{(0,t)}(x_0)$ with $x_0 \in B_R(0)$, let $\xi : [0,T] \to \mathbb{R}^d$ be the solution of the linearised problem*

$$\begin{cases} \dot{\xi}(t) = \nabla_x \mathscr{F}(t, x(t), \theta(t))\xi(t), \\ \xi(\bar{t}) = v, \end{cases} \tag{A1}$$

where $\bar{t} \in [0,T]$ is the instant of perturbation and v is the direction of perturbation of the trajectory. Then, for every $t \in (\bar{t}, T)$, it holds

$$\left|\Phi^{\theta}_{(\bar{t},t)}(x(\bar{t}) + \epsilon v) - \Phi^{\theta}_{(\bar{t},t)}(x(\bar{t})) - \epsilon\xi(t)\right| \leq C|v|^2\epsilon^2$$

where C is a constant depending on $T, R, \|\theta\|_{L^2}$.

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Proof. For $t \geq \bar{t}$, let us denote with $t \mapsto y(t) := \Phi^{\theta}_{(\bar{t},t)}(x(\bar{t}) + \epsilon v)$ the solution of the modified problem, obtained by perturbing the original trajectory with ϵv at instant $\bar{t}$. Then, since ξ solves (A1), we can write

$$
\begin{aligned}
|y(t) - x(t) - \epsilon\xi(t)| &= \left|\Phi^{\theta}_{(\bar{t},t)}(x(\bar{t}) + \epsilon v) - \Phi^{\theta}_{(\bar{t},t)}(x(\bar{t})) - \epsilon\xi(t)\right| \\
&\leq \int_{\bar{t}}^{t} |\mathscr{F}(s, y(s), \theta(s)) - \mathscr{F}(s, x(s), \theta(s)) - \epsilon\nabla_x \mathscr{F}(s, x(s), \theta(s))\xi(s)| ds \\
&\leq \int_{\bar{t}}^{t} |\mathscr{F}(s, y(s), \theta(s)) - \mathscr{F}(s, x(s), \theta(s)) - \nabla_x \mathscr{F}(s, x(s), \theta(s))(y(s) - x(s))| ds \\
&\quad + \int_{\bar{t}}^{t} |\nabla_x \mathscr{F}(s, x(s), \theta(s))||y(s) - x(s) - \epsilon\xi(s)| ds \\
&\leq \int_{\bar{t}}^{t} \left[\int_0^1 |\nabla_x \mathscr{F}(s, x(s) + \tau(y(s) - x(s)), \theta(s) - \nabla_x \mathscr{F}(s, x(s), \theta(s)||y(s) - x(s)| d\tau\right] ds \\
&\quad + \int_{\bar{t}}^{t} |\nabla_x \mathscr{F}(s, x(s), \theta(s))||y(s) - x(s) - \epsilon\xi(s)| ds
\end{aligned}
$$

for every $t \geq \bar{t}$. We now address the two integrals separately. Using Assumption 2$-(iv)$ and the result of Lemma A.4, we obtain the following bound

$$
\begin{aligned}
&\int_{\bar{t}}^{t} \left[\int_0^1 |\nabla_x \mathscr{F}(s, x(s) + \tau(y(s) - x(s))), \theta(s) - \nabla_x \mathscr{F}(s, x(s), \theta(s))||y(s) - x(s)| d\tau\right] ds \\
&\leq \int_{\bar{t}}^{t} L_{\bar{R}} \left(1 + |\theta(s)|^2\right) \frac{1}{2} |y(s) - x(s)|^2 ds \\
&\leq \frac{1}{2} L_{\bar{R}} \left(1 + \|\theta\|_{L^2}^2\right) e^{2L_{\bar{R}}(1 + \|\theta\|_{L^1})} |\epsilon v|^2
\end{aligned}
$$

Similarly, for the second integral, owing to Assumption 2$-(iv)$, we can compute:

$$
\int_{\bar{t}}^{t} |\nabla_x \mathscr{F}(s, x(s), \theta(s))||y(s) - x(s) - \epsilon\xi(s)| ds \leq \int_{\bar{t}}^{t} L_{\bar{R}} \left(1 + |\theta(s)|^2\right) (1 + \bar{R})|y(s) - x(s) - \epsilon\xi(s)| ds
$$

Finally, by combining the two results together and using Gronwall's Lemma, we prove the statement. □

Proposition A.6 (Differentiability with respect to control perturbations). *Consider the solution ξ of the linearised problem*

$$
\begin{cases}
\dot{\xi}(t) = \nabla_x \mathscr{F}(t, x^{\theta}(t), \theta(t))\xi(t) + \nabla_{\theta} \mathscr{F}(t, x^{\theta}(t), \theta(t))v(t) \\
\xi(0) = 0
\end{cases}
\tag{A2}
$$

where the control θ is perturbed at the initial time with $\theta + \epsilon v$, when starting with an initial datum $x_0 \in B_R(0)$. Then,

$$
\left|\Phi^{\theta+\epsilon v}_{(0,t)}(x_0) - \Phi^{\theta}_{(0,t)}(x_0) - \epsilon\xi(t)\right| \leq C||v||_{L^2}^2 \epsilon^2 \tag{A3}
$$

where C is a constant depending on $T, \bar{R}, L_{\bar{R}}, \|\theta\|_{L^1}$. Moreover, we have that for every $t \in [0, T]$

$$
\xi(t) = \int_0^t \mathscr{R}^{\theta}_{(s,t)}(x_0) \cdot \nabla_{\theta} \mathscr{F}(s, x^{\theta}(s), \theta(s))v(s)\, ds, \tag{A4}
$$

where $\mathscr{R}^{\theta}_{(s,t)}(x_0)$ has been defined in (3.12).

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Proof. We first observe that the dynamics in (A2) are affine in the ξ variable. Moreover, Assumptions 1–2 guarantee that the coefficients are L^1-regular in time. Hence, from the classical Caratheodory Theorem we deduce the existence and the uniqueness of the solution of (A2). Finally, the identity (A4) follows as a classical application of the resolvent map $(\mathscr{R}^{\theta}_{(s,t)}(x_0)_{s,t\in[0,T]}$ (see,e.g., in [10, Theorem 2.2.3]).
Let us denote with $t \mapsto x(t)$ and $t \mapsto y(t)$ the solutions of Cauchy problem (2.1) corresponding, respectively, to the admissible controls θ and $\theta + \epsilon\nu$. In virtue of Lemma A.1, we have that there exists $\bar{R} > 0$ such that $x(t), y(t) \in B_{\bar{R}}(0)$ for every $t \in [0, T]$. Then, recalling the definition of the flow map provided in (2.2), we compute

$$
\begin{aligned}
|y(t) - x(t) - \epsilon\xi(t)| &= \left|\Phi^{\theta+\epsilon\nu}_{(0,t)}(x_0) - \Phi^{\theta}_{(0,t)}(x_0) - \epsilon\xi(t)\right| \\
&\leq \int_0^t |\mathscr{F}(s, y(s), \theta(s) + \epsilon\nu(s)) - \mathscr{F}(s, x(s), \theta(s)) - \epsilon\dot{\xi}(s)|\, ds \\
&\leq \int_0^t |\mathscr{F}(s, y(s), \theta(s) + \epsilon\nu(s)) - \mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s)) \\
&\qquad - \epsilon\nabla_x\mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s)) \cdot (y(s) - x(s))|\, ds \\
&\quad + \int_0^t |\mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s)) - \mathscr{F}(s, x(s), \theta(s)) - \epsilon\nabla_\theta\mathscr{F}(s, x(s), \theta(s)) \cdot \nu(s)|\, ds \\
&\quad + \int_0^t |\nabla_x\mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s)) - \nabla_x\mathscr{F}(s, x(s), \theta(s))||y(s) - x(s)|ds \\
&\quad + \int_0^t |\nabla_x\mathscr{F}(s, x(s), \theta(s))||y(s) - x(s) - \epsilon\xi(s)|\, ds.
\end{aligned}
$$

We now handle each term separately:

$$
\begin{aligned}
&\int_0^t |\mathscr{F}(s, y(s), \theta(s) + \epsilon\nu(s)) - \mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s)) - \epsilon\nabla_x\mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s))(y(s) - x(s))|\, ds \\
&\leq \int_0^t \left[\int_0^1 L_{\bar{R}} \left(1 + |\theta(s) + \epsilon\nu(s)|^2\right) \tau |y(s) - x(s)|^2 d\tau\right] ds \\
&\leq L^3_{\bar{R}} \left(1 + \|\theta\|_{L^2} + \epsilon \|\nu\|_{L^2}\right)^4 e^{2L_{\bar{R}}\left(1+\|\theta\|_{L^1}\right)} \|\nu\|^2_{L^2} \epsilon^2
\end{aligned}
\tag{A5}
$$

where we used Assumption 2−(*iv*) and Lemma A.4. By using Assumption 2−(*v*), we obtain the following bounds for the second integral:

$$
\begin{aligned}
&\int_0^t |\mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s)) - \mathscr{F}(s, x(s), \theta(s)) - \nabla_\theta\mathscr{F}(s, x(s), \theta(s)) \cdot \epsilon\nu(s)|\, ds \\
&\leq \int_0^t \left[\int_0^1 L_{\bar{R}} |\nu(s)|^2 \epsilon^2 \tau d\tau\right] ds = \frac{1}{2} L_{\bar{R}} \|\nu\|^2_{L^2} \epsilon^2.
\end{aligned}
\tag{A6}
$$

Similarly, the third integral can be bounded by using Assumption 2−(*vi*) and Lemma A.4, and it yields

$$
\begin{aligned}
&\int_0^t |\nabla_x\mathscr{F}(s, x(s), \theta(s) + \epsilon\nu(s)) - \nabla_x\mathscr{F}(s, x(s), \theta(s))||y(s) - x(s)|\, ds \\
&\leq \int_0^t L_{\bar{R}}(1 + |\theta(s)| + \epsilon|\nu(s)|)\epsilon|y(s) - x(s)||\nu(s)|\, ds \\
&\leq L^2_{\bar{R}} \left(1 + \|\theta\|_{L^2} + \epsilon \|\nu\|_{L^2}\right)^2 e^{L_{\bar{R}}\left(1+\|\theta\|_{L^1}\right)} \|\nu\|^2_{L^2} \epsilon^2.
\end{aligned}
\tag{A7}
$$

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

Finally, the fourth integral can be bounded using Assumption 2−(*iv*) as follows:

$$\int_0^t |\nabla_x \mathscr{F}(s, x(s), \theta(s))||y(s) - x(s) - \epsilon\xi(s)|ds \leq \int_0^t L_{\bar{R}}(1+\bar{R})\left(1+|\theta(s)|^2\right)|y(s) - x(s) - \epsilon\xi(s)|\, ds. \tag{A8}$$

Hence, by combining (A5), (A6), (A7) and (A8), the thesis follows from Gronwall Lemma. □

Proposition A.7 (Properties of the resolvent map). *Let us assume that the controlled dynamics $\mathscr{F}$ satisfies Assumptions 1–2. Given an admissible control $\theta \in L^2([0,T], \mathbb{R}^m)$ and a trajectory $t \mapsto x(t) = \Phi^\theta_{(0,t)}(x)$ with $x \in B_R(0)$, for every $\tau \in [0,T]$ the resolvent map $\mathscr{R}^\theta_{(\tau,\cdot)}(x) : [0,T] \to \mathbb{R}^{d\times d}$ is the curve $s \mapsto \mathscr{R}^\theta_{(\tau,s)}(x_0)$ that solves*

$$\begin{cases} \frac{d}{ds}\mathscr{R}^\theta_{(\tau,s)}(x) = \nabla_x \mathscr{F}\left(s, \Phi^\theta_{(0,s)}(x), \theta(s)\right) \cdot \mathscr{R}^\theta_{(\tau,s)}(x) \quad \text{for a.e. } s \in [0,T], \\ \mathscr{R}^\theta_{(\tau,\tau)}(x) = \mathrm{Id}. \end{cases} \tag{A9}$$

Then for every $\tau, s \in [0,T]$, there exists a constant C_1 depending on $T, R, \|\theta\|_{L^2}$ such that

$$\left|\mathscr{R}^\theta_{(\tau,s)}(x)\right| := \sup_{v\neq 0} \frac{\left|\mathscr{R}^\theta_{(\tau,s)}(x)\cdot v\right|}{|v|} \leq C_1. \tag{A10}$$

Moreover, for every $x, y \in B_R(0)$, there exists a constant C_2 depending on $T, R, \|\theta\|_{L^2}$ such that

$$\left|\mathscr{R}^\theta_{(\tau,s)}(x) - \mathscr{R}^\theta_{(\tau,s)}(y)\right| := \sup_{v\neq 0} \frac{\left|\mathscr{R}^\theta_{(\tau,s)}(x)\cdot v - \mathscr{R}^\theta_{(\tau,s)}(y)\cdot v\right|}{|v|} \leq C_2|x-y|. \tag{A11}$$

Finally, if θ_1, θ_2 satisfy $\|\theta_1\|, \|\theta_2\| \leq \rho$, then there exists a constant C_3 depending on T, R, ρ such that

$$\left|\mathscr{R}^{\theta_1}_{(\tau,s)}(x) - \mathscr{R}^{\theta_2}_{(\tau,s)}(x)\right| := \sup_{v\neq 0} \frac{\left|\mathscr{R}^{\theta_1}_{(\tau,s)}(x)\cdot v - \mathscr{R}^{\theta_2}_{(\tau,s)}(x)\cdot v\right|}{|v|} \leq C_3\|\theta_1 - \theta_2\|_{L^2}. \tag{A12}$$

Proof. We first prove the boundedness of the resolvent map. Let us fix $v \in \mathbb{R}^d$ with $v \neq 0$, and let us define $\xi(s) := \mathscr{R}^\theta_{(\tau,s)}(x)\cdot v$ for every $s \in [0,T]$. Then, in virtue of Assumption 2−(*vi*), we have:

$$|\xi(s)| \leq |\xi(\tau)| + \int_\tau^s \left|\nabla_x \mathscr{F}(\sigma, \Phi^\theta_{(0,\sigma)}(x), \theta(\sigma))\right| |\xi(\sigma)|\, d\sigma \leq |v| + L_{\bar{R}} \int_0^t \left(1+\theta(\sigma)^2\right)|\xi(\sigma)|\, d\sigma,$$

and, by Gronwall's Lemma, we deduce (A10). Similarly as before, given $x, y \in B_R(0)$ and $v \neq 0$, let us define $\xi^x(s) := \mathscr{R}^\theta_{(\tau,s)}(x)\cdot v$ and $\xi^y(s) := \mathscr{R}^\theta_{(\tau,s)}(y)\cdot v$ for every $s \in [0,T]$. Then, we have that

$$\begin{aligned} |\xi^x(s) - \xi^y(s)| &\leq \int_\tau^s \left|\nabla_x \mathscr{F}\left(\sigma, \Phi^\theta_{(0,\sigma)}(x), \theta(\sigma)\right)\xi^x(\sigma) - \nabla_x \mathscr{F}\left(\sigma, \Phi^\theta_{(0,\sigma)}(y), \theta(\sigma)\right)\xi^y(\sigma)\right| d\sigma \\ &\leq \int_\tau^s \left|\nabla_x \mathscr{F}\left(\sigma, \Phi^\theta_{(0,\sigma)}(x), \theta(\sigma)\right) - \nabla_x \mathscr{F}\left(\sigma, \Phi^\theta_{(0,\sigma)}(y), \theta(\sigma)\right)\right| |\xi^y(\sigma)|\, d\sigma \\ &\quad + \int_\tau^s \left|\nabla_x \mathscr{F}\left(\sigma, \Phi^\theta_{(0,\sigma)}(x), \theta(\sigma)\right)\right| |\xi^x(\sigma) - \xi^y(\sigma)|\, d\sigma \\ &\leq C_1|v| \int_\tau^s L_{\bar{R}}\left(1+\theta(\sigma)^2\right)\left|\Phi^\theta_{(0,\sigma)}(x) - \Phi^\theta_{(0,\sigma)}(y)\right| d\sigma \\ &\quad + \int_\tau^s L_{\bar{R}}\left(1+\theta(\sigma)^2\right)|\xi^x(\sigma) - \xi^y(\sigma)|\, d\sigma, \end{aligned}$$

where we used (A10) and Assumption 2-(*iv*). Hence, combining Lemma A.2 with Gronwall's Lemma, we deduce (A11). Finally, we prove the dependence of the resolvent map on different controls $\theta_1, \theta_2 \in L^2([0,T];\mathbb{R}^m)$. Given $x \in B_R(0)$ and $v \neq 0$, let us define $\xi^{\theta_1}(s) := \mathscr{R}^{\theta_1}_{(\tau,s)}(x)\cdot v$ and $\xi^{\theta_2}(s) := \mathscr{R}^{\theta_2}_{(\tau,s)}(x)\cdot v$ for

https://doi.org/10.1017/S0956792524000032 Published online by Cambridge University Press

every $s \in [0, T]$. Then, we compute

$$\begin{aligned}
\left|\xi^{\theta_1}(s) - \xi^{\theta_2}(s)\right| &\leq \int_\tau^s \left|\nabla_{x'}\mathscr{F}\left(\sigma, \Phi^{\theta_1}_{(0,\sigma)}(x), \theta_1(\sigma)\right)\xi^{\theta_1}(\sigma) - \nabla_{x'}\mathscr{F}\left(\sigma, \Phi^{\theta_2}_{(0,\sigma)}(x), \theta_2(\sigma)\right)\xi^{\theta_2}(\sigma)\right| d\sigma \\
&\leq \int_\tau^s \left|\nabla_{x'}\mathscr{F}\left(\sigma, \Phi^{\theta_1}_{(0,\sigma)}(x), \theta_1(\sigma)\right) - \nabla_{x'}\mathscr{F}\left(\sigma, \Phi^{\theta_2}_{(0,\sigma)}(x), \theta_2(\sigma)\right)\right| |\xi^{\theta_1}(\sigma)| \, d\sigma \\
&\quad + \int_\tau^s \left|\nabla_{x'}\mathscr{F}\left(\sigma, \Phi^{\theta_2}_{(0,\sigma)}(y), \theta(\sigma)\right)\right| \left|\xi^{\theta_1}(\sigma) - \xi^{\theta_2}(\sigma)\right| \, d\sigma \\
&\leq C_1|v| \int_\tau^s L_{\bar{R}}\left(1 + \theta_1(\sigma)^2\right) \left|\Phi^{\theta_1}_{(0,\sigma)}(x) - \Phi^{\theta_2}_{(0,\sigma)}(x)\right| \, d\sigma \\
&\quad + C_1|v| \int_\tau^s L_{\bar{R}}(1 + |\theta_1(\sigma)| + |\theta_2(\sigma)|)|\theta_1(\sigma) - \theta_2(\sigma)| \, d\sigma \\
&\quad + \int_\tau^s L_{\bar{R}}\left(1 + \theta(\sigma)^2\right) |\xi^{\theta_1}(\sigma) - \xi^{\theta_2}(\sigma)| \, d\sigma,
\end{aligned}$$

where we used Assumption 2−(*iv*)-(*vi*). □

Cite this article: Cipriani C., Fornasier M. and Scagliotti A. From NeurODEs to AutoencODEs: A mean-field control framework for width-varying neural networks. *European Journal of Applied Mathematics*, https://doi.org/10.1017/S0956792524000032