Layerwise goal-oriented adaptivity for neural ODEs: An optimal control perspective

Michael Hintermüller[1,3], Michael Hinze[2], Denis Korolev [3]

submitted: January 14, 2026

[1] Humboldt-Universität zu Berlin Unter den Linden 6 10099 Berlin Germany E-Mail: hint@math.hu-berlin.de	[2] Universität Koblenz Universitätsstr. 1 56070 Koblenz Germany E-Mail: hinze@uni-koblenz.de	[3] Weierstrass Institute Anton-Wilhelm-Amo-Str. 39 10117 Berlin Germany E-Mail: michael.hintermueller@wias-berlin.de denis.korolev@wias-berlin.de

No. 3254

Berlin 2026

Layerwise goal-oriented adaptivity for neural ODEs: An optimal control perspective

Michael Hintermüller, Michael Hinze, Denis Korolev

Abstract

In this work, we propose a novel layerwise adaptive construction method for neural network architectures. Our approach is based on a goal–oriented dual-weighted residual technique for the optimal control of neural differential equations. This leads to an ordinary differential equation constrained optimization problem with controls acting as coefficients and a specific loss function. We implement our approach on the basis of a DG(0) Galerkin discretization of the neural ODE, leading to an explicit Euler time marching scheme. For the optimization we use steepest descent. Finally, we apply our method to the construction of neural networks for the classification of data sets, where we present results for a selection of well known examples from the literature.

1 Introduction

In recent years, a new paradigm in mathematical machine learning has come into focus, which considers parameterized neural differential equations as continuum models for residual neural networks [1], [2], [3], [4], [5], [6]. The hidden components of such a network are encoded in a state $\boldsymbol{x}$ $(=\boldsymbol{x}(t))$, which evolves continuously over the "time" horizon[1] $[0, T]$, $T > 0$ and $t \in [0, T]$, according to a neural vector field $F(\boldsymbol{x}, \theta)$, thus transforming the (given) input data $\boldsymbol{x}_{\mathrm{in}}(= \boldsymbol{x}(0))$ into its feature representation $\boldsymbol{x}(T)$. The latter is then used to fit the output (label) data $\boldsymbol{y}$, which are given. This is typically achieved by learning the parameters θ $(= \theta(t))$ occurring in the neural differential equation via training data and the minimization of a suitable objective. Mathematically, this leads to the solution of an optimization (or parameter identification) problem with ordinary differential equation (ODE) constraints [4], [7], [8] reading

$$\begin{aligned} \inf \ & \mathcal{J}(\boldsymbol{x}, \theta) := J(\boldsymbol{x}) + \mathcal{R}(\theta), \quad \text{over} \ \ (\boldsymbol{x}, \theta), \\ \text{subject to:} \ & \dot{\boldsymbol{x}} = F(\boldsymbol{x}, \theta), \quad \boldsymbol{x}(0) = \boldsymbol{x}_{\mathrm{in}}, \end{aligned} \tag{1}$$

where J typically represents the data fit term (thus depending also on $\boldsymbol{y}$) and $\mathcal{R}(\theta)$ is a suitable regularization (*prior*). In this setting, we refer to $\boldsymbol{x}$ as the *state* and θ as the *control* variable, respectively. This allows us to use analytical and computational tools developed within the realm of optimal control of differential equations.

More specifically, here we employ such tools to provide an answer to the question, "How should a neural network be designed so that it satisfactorily solves a given task". For this purpose, we rely on the concept of goal-oriented (mesh) adaptivity [9], [10], [11], [12], [13] in our machine learning context. In fact, given an "ideal" continuous network $F(\,\cdot\,, \theta^{\star})$ that defines our *goal* (i.e., the target value for adaptation) $\mathcal{J}(\boldsymbol{x}^{\star}, \theta^{\star})$, we aim to construct a residual network, characterized by a discrete set of parameters $\theta_{\tau}^{\star}$, that fits this goal well. Accordingly, our adaptivity approach is based on the error bound:

$$\left| \mathcal{J}(\boldsymbol{x}^{\star}, \theta^{\star}) - \mathcal{J}(\boldsymbol{x}_{\tau}^{\star}, \theta_{\tau}^{\star}) \right| \leq \frac{1}{2} \sum_{k=1}^{K} \left(R_{\boldsymbol{p}_{\tau}^{\star}}^{k} \, \omega_{\boldsymbol{x}^{\star}}^{k} + |\rho_{\theta^{\star}, \theta_{\tau}^{\star}}^{k}| + R_{\boldsymbol{x}_{\tau}^{\star}}^{k} \, \omega_{\boldsymbol{p}^{\star}}^{k} \right) + |R|, \tag{2}$$

which contains the residuals of the state equation $R_{\boldsymbol{x}_{\tau}^{\star}}^{k}$ and adjoint equation $R_{\boldsymbol{p}_{\tau}^{\star}}^{k}$ weighted by the corresponding errors in the adjoint and the state $\omega_{\boldsymbol{p}^{\star}}^{k}$ and $\omega_{\boldsymbol{x}^{\star}}^{k}$, respectively, as well as the residual of the gradient equation weighted by the error in the control, i.e., $\rho_{\theta^{\star}, \theta_{\tau}^{\star}}^{k}$. Here, R denotes higher order terms which are neglected in the practical implementation, and $K \in \mathbb{N}$ is the number of time steps. Based on this error bound, we propose a computable indicator that drives our depth-adaptive neural network design. This generates a "time" grid, whose grid points are interpreted as layers of an "ideal" neural network.

---

[1]"Time" is here merely artificial, rather than problem immanent.

The stationarity characterization in the form of the first-order optimality system for (1) and its corresponding discretization form the backbone of our approach. In this regard, it is crucial to ensure the existence of an optimal pair $(\boldsymbol{x}^\star, \theta^\star)$ together with the associated Lagrange multiplier (adjoint state) $\boldsymbol{p}^\star$, with the triplet $(\boldsymbol{x}^\star, \theta^\star, \boldsymbol{p}^\star)$ characterizing (one of) the ideal designs which we aim to meet. In order to achieve such a characterization in our functional-analytic setting, an appropriate regularization (or prior) is required. For this purpose, we stabilize (1) by adding an $H^1(0,T)$ regularization $\mathcal{R}$ to the objective J that enforces smooth temporal variation of the network parameters and helps to show that an ideal design indeed exists. From an optimality perspective, this regularization induces the H^1 topology in which the gradient required for neural network optimization must be computed by solving a boundary value problem. On the discrete level, the latter is solved by combining a discontinuous DG(0) Petrov–Galerkin discretization for the state and adjoint equations with a continuous CG(1) Galerkin discretization for the gradient equation.

To the best of our knowledge, this approach to neural network design is new. Conceptually related approaches include the sensitivity-based insertion of layers into an existing neural network during training [14] and the adaptive successive approximation approach of [15]. In the latter, the optimality system resulting from the associated optimization problem is solved algorithmically with the aid of a fixed-point iteration (method of successive approximation), whereby in each iteration of the algorithm, the state and adjoint state are approximated on the basis of residual error estimators. The solution algorithm is therefore treated adaptively in its sub-steps. Our approach differs in that we derive the adaptivity directly from the optimality system. In this way, we take advantage of the fact that the optimality system mathematically represents a boundary value problem that can be treated adaptively using suitable methods.

The structure and contributions of the paper are as follows: In Section 2, the neural ODE is introduced and cast into its batch form. The corresponding optimal control problem together with a suitable functional analytic framework is introduced in Section 3. The first-order stationarity conditions are derived and analyzed in Section 4. In Section 5, we propose our discretization of the optimality system. A dual-weighted posteriori error bound for the training objective is derived in Section 6, and the numerical realization of our layerwise algorithm is discussed in Section 7. In Section 8, we report on numerical tests for our adaptive approach on binary and multiclass classification problems. The paper ends with our conclusion and an outlook in Section 9.

2 Neural ODEs

The key idea of neural ODEs is to treat the depth of a neural network as a continuous variable, replacing discrete layers with a differential equation parameterized by weights and biases. Over a time interval $I = (0,T)$, with $T > 0$, we consider the following time-dependent weight matrix and bias vector

$$W : I \to \mathbb{R}^{d \times d}, \quad b : I \to \mathbb{R}^d, \tag{3}$$

where $d \in \mathbb{N}$ denotes the dimension of the hidden state. We then collect these quantities into a time-dependent parameter vector $\theta : I \to \mathbb{R}^n$, defined by

$$\theta(t) = \big(\mathrm{vec}\, W(t),\ b(t)\big)^\top, \tag{4}$$

where $n = d^2 + d$, and the operator $\mathrm{vec} : \mathbb{R}^{d \times d} \to \mathbb{R}^{d^2}$ stacks the columns of a matrix into a single vector. Let $f : \mathbb{R}^d \times \mathbb{R}^n \to \mathbb{R}^d$ denote the neural vector field

$$f(v, \theta(t)) := \boldsymbol{\sigma}(W(t)v + b(t)), \quad t \in [0,T], \tag{5}$$

where $\boldsymbol{\sigma} : \mathbb{R}^d \to \mathbb{R}^d$ is the vector-valued activation function

$$\boldsymbol{\sigma}(v) = (\sigma(v_i), \dots, \sigma(v_d))^\top, \tag{6}$$

and $\sigma : \mathbb{R} \to \mathbb{R}$ is a scalar activation function. Given an input $x_0 \in \mathbb{R}^{d_{\mathrm{in}}}$, $d_{\mathrm{in}} \in \mathbb{N}$, the neural ODE seeks a trajectory $x : [0,T] \to \mathbb{R}^d$ satisfying

$$\begin{cases} \dot{x}(t) = f(x(t), \theta(t)) & \text{for} \quad t \in I, \\ x(0) = W_{\mathrm{in}}\, x_0, \end{cases} \tag{7}$$

where $W_{\mathrm{in}} \in \mathbb{R}^{d \times d_{\mathrm{in}}}$ is the input projection matrix mapping the input data into the state space. Further, $\dot{x}$ denotes the time derivative of x with respect to time t.

In supervised learning, a batch of m training examples $\{(x_0^i, y^i)\}_{i=1}^m \subset \mathbb{R}^{d_{\text{in}}} \times \mathbb{R}^{d_{\text{out}}}$, $d_{\text{out}} \in \mathbb{N}$, is typically given. For both generality and practical implementation, it is convenient to express (7) in a batch formulation. For this, we define

$$\boldsymbol{x}_0 = (x_0^1, \ldots, x_0^m)^\top \in \mathbb{R}^{md_{\text{in}}}, \quad \boldsymbol{y} = (y^1, \ldots, y^m)^\top \in \mathbb{R}^{md_{\text{out}}}, \tag{8}$$

with $m \in \mathbb{N}$. The batched neural vector field $F : \mathbb{R}^{md} \times \mathbb{R}^n \to \mathbb{R}^{md}$ acts component-wise as

$$F(\boldsymbol{x}, \theta) := \big(f(x^1, \theta), \ldots, f(x^m, \theta)\big)^\top, \quad \boldsymbol{x} = (x^1, \ldots, x^m)^\top \in \mathbb{R}^{md}, \tag{9}$$

where each trajectory $x^i : [0, T] \to \mathbb{R}^d$ satisfies (7) with $x^i(0) = W_{\text{in}} x_0^i$. The batched projected input is

$$\boldsymbol{x}_{\text{in}} := (W_{\text{in}} x_0^1, \ldots, W_{\text{in}} x_0^m)^\top \in \mathbb{R}^{md}. \tag{10}$$

The i-th block component satisfies $\boldsymbol{x}^i = x^i$, $F^i(\boldsymbol{x}, \theta) = f(x^i, \theta)$ and $\boldsymbol{x}_{\text{in}}^i = W_{\text{in}} x_0^i$. The resulting *full-batch neural ODE* is then given by

$$\begin{cases} \dot{\boldsymbol{x}}(t) = F(\boldsymbol{x}(t), \theta(t)) \quad \text{for} \quad t \in I, \\ \boldsymbol{x}(0) = \boldsymbol{x}_{\text{in}}. \end{cases} \tag{11}$$

The system (11) evolves the input $\boldsymbol{x}_{\text{in}}$ into a batch feature representation $\boldsymbol{x}(T)$. The output map $\boldsymbol{q}_{\text{out}} : \mathbb{R}^{md} \to \mathbb{R}^{md_{\text{out}}}$ transforms these features into predictions $\hat{\boldsymbol{y}} \in \mathbb{R}^{md_{\text{out}}}$, and is defined component-wise as

$$\boldsymbol{q}_{\text{out}}(\boldsymbol{x}(T)) = (\hat{y}^1, \ldots, \hat{y}^m)^\top =: \hat{\boldsymbol{y}}, \tag{12}$$

where $\hat{y}^i := q_{\text{out}}(W_{\text{out}} x^i(T))$ for $i \in \{1, \cdots, m\}$, $q_{\text{out}} : \mathbb{R}^{d_{\text{out}}} \to \mathbb{R}^{d_{\text{out}}}$ is a task-dependent continuous output function, and $W_{\text{out}} \in \mathbb{R}^{d_{\text{out}} \times d}$ is the output weight matrix.

A word on notation, before we continue: In this work, the Euclidean norm and dot product are denoted by $\|\cdot\|$ and $(\cdot, \cdot)$, respectively. For batched vectors $\boldsymbol{x} \in \mathbb{R}^{md}$ as in (8) or (9), we use the batch norm associated with the product space $(\mathbb{R}^d)^m$, which is given by

$$\|\boldsymbol{x}\|^2 = \sum_{i=1}^m \|x^i\|^2. \tag{13}$$

For the ease of notation, we assume that W_{in} and W_{out} are fixed, non-trainable parameters. This assumption does not affect the conclusions of our analysis, and extending the results to trainable W_{in} and W_{out} is straightforward. Nevertheless, where necessary, we provide computational formulas that allow these parameters to be included in the learning process.

3   Optimal control of neural ODEs

The optimal control problem associated with (11) consists in identifying optimal parameters θ based on the training data (8) and a suitable learning objective. Here, we address this problem within the "first optimize-then-discretize"approach, i.e., we first introduce an appropriate functional-analytic setting and derive stationarity conditions, which are then discretized for computational purposes. We use the standard notation for Sobolev–Bochner spaces, such as $L^2(I; \mathbb{R}^{md})$ and $H^1(I; \mathbb{R}^{md})$; see, e.g., [16]. Their respective norms are consistent with (13) and are defined as follows: For $\boldsymbol{x} \in \mathbb{R}^{md}$ as in (9), we have

$$\|\boldsymbol{x}\|_{L^2(I; \mathbb{R}^{md})}^2 = \sum_{i=1}^m \int_0^T \|x^i(t)\|^2 \, dt, \quad \|\boldsymbol{x}\|_{H^1(I; \mathbb{R}^{md})}^2 = \|\dot{\boldsymbol{x}}\|_{L^2(I; \mathbb{R}^{md})}^2 + \|\boldsymbol{x}\|_{L^2(I; \mathbb{R}^{md})}^2.$$

Furthermore, $C([0, T]; \mathbb{R}^d)$ denotes the space of continuous functions on $[0, T]$ with values in $\mathbb{R}^d$, equipped with the supremum norm $\|\boldsymbol{v}\|_\infty := \sup_{t \in [0, T]} \|\boldsymbol{v}(t)\|$. For brevity, we often omit the respective Euclidean space in the definition of function spaces in our proofs, specifically when the appropriate dimension is clear from the context.

We begin by discussing a suitable function space for the ODE parameters (4). Regularization is naturally used to ensure that the quantities in (4) belong to a specific function class. For $\lambda > 0$, the functional

$$\mathcal{R}(\theta) = \frac{\lambda}{2} \int_0^T \left(\|\theta(t)\|^2 + \|\dot{\theta}(t)\|^2 \right) dt \tag{14}$$

ensures that the controls θ belong to $H^1(I)$. Indeed, without the derivative term in (14), one only has $\theta \in L^2(I)$, which admits discontinuities. We note that a regularization technique similar to (14) was used in [4] to improve neural ODE stability within a 'first discretize–then–optimize' setting, where the derivative $\dot{\theta}(t)$ is replaced by its finite-difference approximation.

To guarantee the existence and uniqueness of solutions to (11) in appropriate function spaces, we consider conditions on (4) that can be enforced by choosing a suitable regularization, as in (14).

Proposition 3.1. *Suppose that $\theta \in L^2(I; \mathbb{R}^n)$ and that $\boldsymbol{\sigma} \in C(\mathbb{R}^d)$ is Lipschitz continuous, i.e., there exists $L_\sigma > 0$ such that*

$$\|\boldsymbol{\sigma}(\boldsymbol{s}_2) - \boldsymbol{\sigma}(\boldsymbol{s}_1)\| \leq L_\sigma \|\boldsymbol{s}_2 - \boldsymbol{s}_1\|, \quad \forall \boldsymbol{s}_1, \boldsymbol{s}_2 \in \mathbb{R}^d. \tag{15}$$

Then the problem (11) admits a unique solution $\boldsymbol{x} \in H^1(I; \mathbb{R}^{md})$. Moreover, there exist finite constants $C_1, C_2 > 0$ such that the following stability bound holds:

$$\|\boldsymbol{x}\|_{H^1(I;\mathbb{R}^{md})} \leq C_1 \exp\left[C_2 \|\theta\|_{L^2(I;\mathbb{R}^n)}\right]. \tag{16}$$

Proof. With a slight abuse of previous notation, but for the convenience of ODE theory, we define $f : \mathbb{R}^d \times [0,T] \to \mathbb{R}^d$ and $F : \mathbb{R}^{md} \times [0,T] \to \mathbb{R}^{md}$ as follows

$$f(x,t) := \boldsymbol{\sigma}(W(t)x + b(t)), \quad F(\boldsymbol{x},t) := \left(f(x^1,t), \ldots, f(x^m,t)\right)^\top.$$

Using the Lipschitz condition (15), for $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^{md}$, we estimate

$$\|F(\boldsymbol{v},t) - F(\boldsymbol{w},t)\| = \left[\sum_{i=1}^m \|f(\boldsymbol{v}^i,t) - f(\boldsymbol{w}^i,t)\|^2\right]^{1/2} \leq L_\sigma \|W(t)\| \|\boldsymbol{v} - \boldsymbol{w}\|. \tag{17}$$

Using (17), we get the following estimate

$$\|F(\boldsymbol{v},t)\| \leq \|F(\boldsymbol{v},t) - F(0,t)\| + \|F(0,t)\| \leq L_\sigma \|W(t)\| \|\boldsymbol{v}\| + \|F(0,t)\|.$$

Applying the Lipschitz condition (15) to the second term above yields the following growth condition

$$\|F(\boldsymbol{v},t)\| \leq \sqrt{m}\, c(t) + a(t)\|\boldsymbol{v}\|, \tag{18}$$

which holds for almost every $t \in [0,T]$. The coefficients in (18) are given by

$$a(t) = L_\sigma \|W(t)\|, \quad c(t) = L_\sigma \|b(t)\| + \|\boldsymbol{\sigma}(0)\|. \tag{19}$$

Since $\theta \in L^2(I)$, then $a, c \in L^2(I)$, and by the Cauchy–Schwarz inequality, we have $a, c \in L^1(I)$ as well.

Note that $t \mapsto W(t)x + b(t)$ is not continuous, and $t \mapsto F(\boldsymbol{x},t)$ may fail to be continuous as well[2]. For this reason, we closely follow the approach of [17, Theorem 3.4]. To begin with we define the norm

$$\|\boldsymbol{v}\|_A = \|e^{-A}\boldsymbol{v}\|_\infty, \quad A(t) = \int_0^t a(s)\, ds.$$

Since $A(s) \leq A(t)$ for $0 \leq s \leq t$, we obtain the norm equivalence $e^{-A(T)}\|\boldsymbol{v}\|_\infty \leq \|\boldsymbol{v}\|_A \leq \|\boldsymbol{v}\|_\infty$, which makes $C([0,T])$ a Banach space when equipped with $\|\cdot\|_A$. For $\boldsymbol{v} \in C([0,T])$, we define an integral operator

$$(\mathfrak{T}\boldsymbol{v})(t) = \boldsymbol{x}_{\mathrm{in}} + \int_0^t F(\boldsymbol{v}(s), s)\, ds. \tag{20}$$

---

[2]Therefore, the famous Picard–Lindelöf theorem [17, Theorem 3.1] is not applicable.

From (18), we get $\|F(\boldsymbol{v}(s), s)\| \leq \sqrt{m}\, c(s) + a(s)M$, $M := \|\boldsymbol{v}\|_\infty$, ensuring that the integrand in (20) is integrable and that $\mathfrak{T}$ is well-defined. For $0 \leq t_1 < t_2$, we get

$$\|(\mathfrak{T}\boldsymbol{v})(t_2) - (\mathfrak{T}\boldsymbol{v})(t_1)\| \leq \int_{t_1}^{t_2} \|F(\boldsymbol{v}(s), s)\|\, ds \leq \int_{t_1}^{t_2} \left(\sqrt{m}\, c(s) + a(s)M \right) ds,$$

where the right-hand side tends to 0 as $t_2 \to t_1$ by the continuity of the Lebesgue integral. Hence, $\mathfrak{T}$ maps $C([0, T])$ into itself. We note that $A'(t) = a(t)$ and apply (17) to deduce

$$\begin{aligned}
\|\mathfrak{T}\boldsymbol{v} - \mathfrak{T}\boldsymbol{w}\|_A &\leq \max_{t \in [0,T]} e^{-A(t)} \int_0^t \|F(\boldsymbol{v}(s), s) - F(\boldsymbol{w}(s), s)\|\, ds \leq \max_{t \in [0,T]} e^{-A(t)} \int_0^t a(s) \|\boldsymbol{v}(s) - \boldsymbol{w}(s)\|\, ds \\
&= \max_{t \in [0,T]} e^{-A(t)} \int_0^t A'(s) e^{A(s)} e^{-A(s)} \|\boldsymbol{v}(s) - \boldsymbol{w}(s)\|\, ds \\
&\leq \max_{t \in [0,T]} e^{-A(t)} \left(\int_0^t A'(s) e^{A(s)}\, ds \right) \|\boldsymbol{v} - \boldsymbol{w}\|_A \\
&\leq \max_{t \in [0,T]} e^{-A(t)} \left(e^{A(t)} - 1 \right) \|\boldsymbol{v} - \boldsymbol{w}\|_A \leq \left(1 - e^{-A(T)} \right) \|\boldsymbol{v} - \boldsymbol{w}\|_A,
\end{aligned}$$

Since $1 - e^{-A(T)} < 1$, the Banach fixed-point theorem guarantees a unique $\boldsymbol{x} \in C([0, T])$ such that $\mathfrak{T}\boldsymbol{x} = \boldsymbol{x}$. By the Lebesgue differentiation theorem, we get

$$\dot{\boldsymbol{x}}(t) = \frac{d}{dt}(\mathfrak{T}\boldsymbol{x})(t) = F(\boldsymbol{x}, t),$$

which holds for almost every $t \in [0, T]$. The growth condition (18) and the triangle inequality yield

$$\|\dot{\boldsymbol{x}}\|_{L^2(I)} \leq \left\| \sqrt{m}\, c + a \|\boldsymbol{x}\| \right\|_{L^2(I)} \leq \sqrt{m}\, \|c\|_{L^2(I)} + \|\boldsymbol{x}\|_\infty \|a\|_{L^2(I)}. \tag{21}$$

Therefore, $\dot{\boldsymbol{x}} \in L^2(I)$. Hence, $\boldsymbol{x} \in H^1(I)$ and solves (11), as required.

We proceed with the proof of the stability bound (16), beginning with the observation that (21) implies

$$\|\boldsymbol{x}\|_{H^1(I)} \leq \|\boldsymbol{x}\|_{L^2(I)} + \|\dot{\boldsymbol{x}}\|_{L^2(I)} \leq \left[\sqrt{T} + \|a\|_{L^2(I)} \right] \|\boldsymbol{x}\|_\infty + \sqrt{m}\, \|c\|_{L^2(I)}. \tag{22}$$

Since $a(t) \geq 0$ and $c(t) \geq 0$ for almost every $t \in [0, T]$, the condition (18) yields

$$\|\boldsymbol{x}(t)\| \leq \sqrt{m} \int_0^t c(s)\, ds + \int_0^t a(s) \|\boldsymbol{x}(s)\|\, ds \tag{23}$$

By the continuity of the Lebesgue integral, the first term on the right-hand side of (23) is a continuous, monotonically increasing function of t; applying Grönvall's inequality [18, Proposition 2.1] then yields

$$\|\boldsymbol{x}(t)\| \leq \sqrt{m} \exp\left(\int_0^t a(s)\, ds \right) \int_0^t c(s)\, ds,$$

which holds for all $t \in [0, T]$. Using the Cauchy–Schwarz inequality and the fact that the integrands in the above estimate are non-decreasing, we obtain

$$\|\boldsymbol{x}\|_\infty \leq \sqrt{T m} \exp\left(\sqrt{T} \|a\|_{L^2(I)} \right) \|c\|_{L^2(I)}. \tag{24}$$

Combining (24) with (22) yields

$$\|\boldsymbol{x}\|_{H^1(I)} \leq \sqrt{m} \left[\sqrt{T} \left(\sqrt{T} + \|a\|_{L^2(I)} \right) \exp\left(\sqrt{T} \|a\|_{L^2(I)} \right) + 1 \right] \|c\|_{L^2(I)}.$$

From this bound, it is clear that there exist constants $C_1, C_2 > 0$ depending only on L_σ, m, $\|\boldsymbol{\sigma}(0)\|$ and T such that the stability bound (16) holds. $\qquad\square$

Remark 1. *If $\theta \in H^1(I)$, then by the continuous embedding $H^1(I) \hookrightarrow C([0, T])$ [19], the parameters in (4), and thus the mapping $t \mapsto W(t)x + b(t)$, are continuous. Since $[0, T]$ is compact and $\theta \in C([0, T])$, the parameters in (4) are bounded on $[0, T]$. Hence, $C := \sup_{t \in [0,T]} \|W(t)\|$ is finite and (17) becomes*

$$\|F(\boldsymbol{v}, t) - F(\boldsymbol{w}, t)\| \leq C L_\sigma \|\boldsymbol{v} - \boldsymbol{w}\|.$$

independently of $t \in [0, T]$. Therefore, $F(\cdot, t)$ is Lipschitz continuous in $\boldsymbol{x}$, and the Lipschitz constant in (17) can be chosen independently of t. By the Picard–Lindelöf theorem, the problem (11) then has a unique solution in $C^1([0, T])$.

Proposition 3.1 yields the existence of the so-called *control-to-state map*

$$\mathfrak{S} : L^2(I; \mathbb{R}^n) \to H^1(I; \mathbb{R}^{md}), \quad \theta \mapsto \boldsymbol{x} =: \mathfrak{S}(\theta). \tag{25}$$

This mapping is nonlinear, and therefore its boundedness and continuity must be proved separately. The boundedness of (25) readily follows from the stability bound (16). The next result establishes the continuous dependence of ODE solutions of (11) on the control coefficients (4), i.e., the continuity of (25).

Proposition 3.2. *Suppose that $\boldsymbol{\sigma} \in C(\mathbb{R}^d)$ and is Lipschitz continuous with some Lipschitz constant $L_\sigma > 0$. Then the control-to-state mapping $\mathfrak{S} : L^2(I; \mathbb{R}^n) \to H^1(I; \mathbb{R}^{md})$ is continuous.*

Proof. Let $\theta_1, \theta_2 \in L^2(I)$ be two distinct coefficients given by (4), i.e.,

$$\theta_1(t) = \big(\mathrm{vec}\, W_1(t),\, b_1(t)\big)^\top, \quad \theta_2(t) = \big(\mathrm{vec}\, W_2(t),\, b_2(t)\big)^\top.$$

According to Proposition 3.1, θ_1 and θ_2 produce unique neural ODE solutions $\boldsymbol{x}_1, \boldsymbol{x}_2 \in H^1(I)$ with the same initial value $\boldsymbol{x}_{\mathrm{in}}$. Let $\delta\boldsymbol{x}(t) = \boldsymbol{x}_1(t) - \boldsymbol{x}_2(t)$. Observe that $\delta\boldsymbol{x} \in H^1(I)$ implies $\delta\boldsymbol{x} \in C([0,T])$ by the continuous embedding $H^1(I) \hookrightarrow C([0,T])$. The following inequality holds for almost every $t \in [0,T]$:

$$\|\delta\dot{\boldsymbol{x}}(t)\| \leq \big[\|F(\boldsymbol{x}_1, \theta_1) - F(\boldsymbol{x}_2, \theta_1)\| + \|F(\boldsymbol{x}_2, \theta_1) - F(\boldsymbol{x}_2, \theta_2)\|\big](t). \tag{26}$$

Similar to (17), from the Lipschitz continuity of $\boldsymbol{\sigma}$, we obtain the following bound for almost every $t \in [0,T]$:

$$\big[\|F(\boldsymbol{x}_1, \theta_1) - F(\boldsymbol{x}_2, \theta_1)\|\big](t) \leq a(t)\|\delta\boldsymbol{x}(t)\|, \tag{27}$$

where $a \in L^1(I)$ is given by (19). By integrating inequality (26), using $\delta\boldsymbol{x}(0) = 0$ and (27), we get

$$\|\delta\boldsymbol{x}(t)\| \leq \int_0^t a(s)\|\delta\boldsymbol{x}(s)\|\, ds + \int_0^t \big[\|F(\boldsymbol{x}_2, \theta_1) - F(\boldsymbol{x}_2, \theta_2)\|\big](s)\, ds. \tag{28}$$

By the continuity of the Lebesgue integral, the first term on the right-hand side of (28) is a continuous, monotonically increasing function of t. Additionally, $a(t) \geq 0$ for almost every $t \in [0,T]$. Therefore, Grönvall's inequality [18, Proposition 2.1] applied to (28) yields

$$\|\delta\boldsymbol{x}(t)\| \leq \exp\Big(\int_0^t a(s)\, ds\Big) \int_0^t \big[\|F(\boldsymbol{x}_2, \theta_1) - F(\boldsymbol{x}_2, \theta_2)\|\big](s)\, ds,$$

By the Cauchy–Schwarz inequality and monotonicity of the above integrals, we get

$$\|\delta\boldsymbol{x}(t)\|_{L^2(I)} \leq \sqrt{T}\, C_a \|F(\boldsymbol{x}_2, \theta_1) - F(\boldsymbol{x}_2, \theta_2)\|_{L^2(I)},$$

where $C_a = \exp\big(\int_0^T a(t)\, dt\big) < \infty$. We further use the Lipschitz condition on $\boldsymbol{\sigma}$, Young's inequality and the elementary inequality $\sqrt{z_1 + z_2} \leq \sqrt{z_1} + \sqrt{z_2}$ for $z_1, z_2 \in \mathbb{R}_{\geq 0}$, and obtain

$$\|\delta\boldsymbol{x}(t)\| \leq \sqrt{T}\, C_a\, L_\sigma \Big[\sum_{i=1}^m \int_0^T \big\|\big(W_1(t) - W_2(t)\big)\boldsymbol{x}_2^i(t) + b_1(t) - b_2(t)\big\|^2\, dt\Big]^{1/2}$$

$$\leq \sqrt{2T}\, C_a\, L_\sigma \Big[\int_0^T \big\|W_1(t) - W_2(t)\big\|^2 \big\|\boldsymbol{x}_2(t)\big\|^2 + m\int_0^T \big\|b_1(t) - b_2(t)\big\|^2\, dt\Big]^{1/2}$$

$$\leq \sqrt{2T}\, C_a\, L_\sigma \|\boldsymbol{x}_2\|_\infty \|W_1 - W_2\|_{L^2(I)} + \sqrt{2T m}\, C_a\, L_\sigma \|b_1 - b_2\|_{L^2(I)},$$

where we also used that $\|\boldsymbol{x}_2\|_\infty < \infty$. Since the above bound is uniform in t, it follows that $\|\delta\boldsymbol{x}\|_\infty \to 0$ as $\|\theta_1 - \theta_2\|_{L^2} \to 0$. Therefore, $\|\delta\boldsymbol{x}\|_{L^2(I)} \to 0$ as $\|\theta_1 - \theta_2\|_{L^2} \to 0$ as well.

To complete the proof, it remains to show that $\|\delta\dot{\boldsymbol{x}}\|_{L^2(I)} \to 0$ as $\|\theta_1 - \theta_2\|_{L^2} \to 0$. We start by applying Young's inequality to (26), and obtain for almost every $t \in [0,T]$ the following

$$\|\delta\dot{\boldsymbol{x}}(t)\|^2 \leq 2\big\|\big[F(\boldsymbol{x}_1, \theta_1) - F(\boldsymbol{x}_2, \theta_1)\big](t)\big\|^2 + 2\big\|\big[F(\boldsymbol{x}_2, \theta_1) - F(\boldsymbol{x}_2, \theta_2)\big](t)\big\|^2.$$

Integrating and applying (27) together with the previous estimates, we get

$$\|\delta\dot{\boldsymbol{x}}\|_{L^2(I)} \leq \sqrt{2}\|a\|_{L^2(I)}\|\delta\boldsymbol{x}\|_\infty + 2L_\sigma\Big[\|\boldsymbol{x}_2\|_\infty\|W_1 - W_2\|_{L^2(I)} + \sqrt{m}\,\|b_1 - b_2\|_{L^2(I)}\Big].$$

Since $\|\delta\boldsymbol{x}\|_\infty \to 0$ as $\|\theta_1 - \theta_2\|_{L^2} \to 0$, the proof is complete. $\qquad\square$

After these preparatory results, we are ready to formulate an optimal control problem with neural ODEs. We introduce the following spaces:

$$\mathcal{W} := H^1(I; \mathbb{R}^{md}), \quad \mathcal{Q} := L^2(I; \mathbb{R}^{md}), \quad \mathcal{V} := \mathcal{Q} \times \mathbb{R}^{md}, \quad \mathcal{U} := H^1(I; \mathbb{R}^n). \tag{29}$$

In our problem, we aim to determine the parameters $\theta \in \mathcal{U}$ of (11) from the training data (8) by optimizing the following loss function

$$J : \mathcal{W} \to \mathbb{R}_{\geq 0}, \quad J(\boldsymbol{x}) = l(\boldsymbol{x}(T)), \tag{30}$$

where $l : \mathbb{R}^{md} \to \mathbb{R}$ is a continuous, problem-dependent function. We define the superposition operator

$$N_F : \mathcal{W} \times \mathcal{U} \to \mathcal{Q}, \quad (\boldsymbol{x}, \theta) \mapsto F(\boldsymbol{x}(\cdot), \theta(\cdot)), \tag{31}$$

and consider the optimal control (or parameter identification) problem

$$\begin{aligned} \inf \ &\mathcal{J}(\boldsymbol{x}, \theta) := J(\boldsymbol{x}) + \mathcal{R}(\theta), \ \text{ over } \ (\boldsymbol{x}, \theta) \in \mathcal{W} \times \mathcal{U}, \\ &\text{subject to: } \ e(\boldsymbol{x}, \theta) = 0, \end{aligned} \tag{32}$$

where $\mathcal{J} : \mathcal{W} \times \mathcal{U} \to \mathbb{R}$ includes the regularizer (14), and the constraint is defined by

$$e : \mathcal{W} \times \mathcal{U} \to \mathcal{V}, \quad (\boldsymbol{x}, \theta) \mapsto e(\boldsymbol{x}, \theta) := \big(\dot{\boldsymbol{x}} - N_F(\boldsymbol{x}, \theta), \ \boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{in}} \big)^{\top},$$

where $\dot{\boldsymbol{x}}$ is now understood in the generalized sense. Since $\mathcal{W} \hookrightarrow C([0, T]; \mathbb{R}^{md})$, the initial value $\boldsymbol{x}(0)$ in the above equation is well-defined. Altogether, (32) is an equality-constrained optimization problem, with the neural ODE defining the constraint.

Next we establish the existence of minimizers for the problem (32), i.e., optimal pairs $(\boldsymbol{x}^\star, \theta^\star) \in \mathcal{W} \times \mathcal{U}$ satisfying $\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) \leq \mathcal{J}(\boldsymbol{x}, \theta)$ for all $(\boldsymbol{x}, \theta) \in \mathcal{W} \times \mathcal{U}$.

Proposition 3.3 (Existence of optimal controls). *Suppose that $\theta \in \mathcal{U}$ with $\mathcal{U} := H^1(I; \mathbb{R}^n)$. Then the problem (32) has an optimal solution.*

Proof. We define $F_{\mathrm{ad}} = \{(\boldsymbol{x}, \theta) \in \mathcal{W} \times \mathcal{U} : \boldsymbol{x} = \mathfrak{S}(\theta)\}$. Proposition 3.1 implies that $F_{\mathrm{ad}} \neq \emptyset$. Therefore, $\inf \{\mathcal{J}(\boldsymbol{x}, \theta) : (\boldsymbol{x}, \theta) \in F_{\mathrm{ad}}\} =: j \in \mathbb{R}_{\geq 0}$. By properties of the infimum, we can pick a (infimizing) sequence $(\boldsymbol{x}_k, \theta_k)_{k \in \mathbb{N}} \subset F_{\mathrm{ad}}$, such that $\mathcal{J}(\boldsymbol{x}_k, \theta_k) \to j$ as $k \to \infty$. Furthermore, we have

$$\lim_{\|\theta\|_{\mathcal{U}} \to \infty} \mathcal{J}(\boldsymbol{x}, \theta) \ \geq \ \lim_{\|\theta\|_{\mathcal{U}} \to \infty} \mathcal{R}(\theta) = +\infty,$$

i.e., $\mathcal{J}$ is radially unbounded with respect to θ. It then follows that $(\theta_k)_{k \in \mathbb{N}}$ is bounded in $\mathcal{U}$; otherwise, if $\|\theta_k\|_{\mathcal{U}} \to \infty$ as $k \to \infty$, then necessarily $\mathcal{J}(\boldsymbol{x}_k, \theta_k) \to \infty$, contradicting the notion of an infimizing sequence. The stability bound (16) yields that $(\boldsymbol{x}_k)_{k \in \mathbb{N}}$ is bounded as well. Since $\mathcal{W}$ and $\mathcal{U}$ are Hilbert spaces and thus reflexive, the Banach–Alaoglu theorem implies that the sequence $(\boldsymbol{x}_k, \theta_k)_{k \in \mathbb{N}}$ has a weakly convergent subsequence in $\mathcal{W} \times \mathcal{U}$, which we denote by $(\boldsymbol{x}_{k'}, \theta_{k'})_{k' \in \mathbb{N}}$, with weak limit $(\boldsymbol{x}^\star, \theta^\star) \in \mathcal{W} \times \mathcal{U}$. From the compact embedding $\mathcal{U} \hookrightarrow C([0, T])$, we obtain that $\theta_{k'} \to \theta^\star$ strongly in $C([0, T])$. Since $C([0, T]) \subset L^2(I)$, Proposition 3.2 implies that the control-to-state mapping $\mathfrak{S} : C([0, T]) \to \mathcal{W}$ is continuous, which in turn implies that $\boldsymbol{x}_{k'} \to \boldsymbol{x}^\star$ strongly in $\mathcal{W}$. Since $\boldsymbol{x}^\star = \mathfrak{S}(\theta^\star)$, we deduce that $(\boldsymbol{x}^\star, \theta^\star) \in F_{\mathrm{ad}}$. Since J is continuous and $\mathcal{R}$ is weakly lower semicontinuous (because the norm is convex and continuous), we obtain

$$\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) \leq \liminf_{k' \to \infty} \mathcal{J}(\boldsymbol{x}_{k'}, \theta_{k'}) = \lim_{k' \to \infty} \mathcal{J}(\boldsymbol{x}_{k'}, \theta_{k'}) = \lim_{k' \to \infty} \mathcal{J}(\boldsymbol{x}_k, \theta_k) = j,$$

where the above subsequence of real numbers converges since the entire sequence converges, implying that the $\liminf$ and the limit indeed coincide. Therefore, $\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) = j$, so that $(\boldsymbol{x}^\star, \theta^\star)$ is a minimizer of $\mathcal{J}$. $\qquad\square$

We note that the control-to-state map (25) is generally non-convex, hence minimizers of (32) need not be unique. We also mention that Proposition 3.3 infers the strong continuity of (25) along minimizing sequences arising from H^1-regular controls. However, under weaker assumptions on the controls, such as $\mathcal{U} = L^2(I; \mathbb{R}^n)$, it is somehow difficult to argue the existence of optimal controls via the direct method of calculus of variations.

4 Stationarity conditions

In this section, we derive first-order optimality (or stationarity) conditions for the problem (32). We begin by considering the corresponding constraint qualification, which ensures the existence of a unique Lagrange multiplier (adjoint state) needed to characterize optimal solutions via the derivative of the Lagrangian. For this, we study the properties of the adjoint equation. In fact, the increased regularity of the adjoint variable allows us to reformulate the adjoint equation in a form suitable for our discretization. The function spaces needed in this section are defined in (29).

Let $D_1 N_F(\boldsymbol{x}, \theta) : \mathcal{W} \to \mathcal{Q}$ denote the derivative of (31) with respect to its first argument:

$$D_1 N_F(\boldsymbol{x}, \theta) = \operatorname{diag}\Big(D_1 f(x^1, \theta), \, D_1 f(x^2, \theta), \, \ldots, \, D_1 f(x^m, \theta)\Big).$$

Since the batched vector field (9) acts component-wise across the batch, $D_1 N_F(\boldsymbol{x}, \theta)$ is block-diagonal. Each block stands for the Jacobian of (5) and is given by

$$D_1 f(x^i(t), \theta(t)) = \operatorname{diag}\big(\boldsymbol{\sigma}'(W(t)x^i(t) + b(t))\big) W(t)^\top, \tag{33}$$

where $\operatorname{diag}\big(\boldsymbol{\sigma}'(v)\big) \in \mathbb{R}^{d \times d}$ is the diagonal matrix with $\boldsymbol{\sigma}'(v) \in \mathbb{R}^d$ along the main diagonal. The derivative $D_2 N_F(\boldsymbol{x}, \theta) : \mathcal{U} \to \mathcal{Q}$ of (31) with respect to its second argument is given by

$$D_2 N_F(\boldsymbol{x}, \theta) = \operatorname{diag}\Big(D_2 f(x^1, \theta), \, D_2 f(x^2, \theta), \, \ldots, \, D_2 f(x^m, \theta)\Big).$$

The derivatives $D_2 f(x^i, \theta)$ $(1 \le i \le m)$, which form blocks on the main diagonal of $D_2 N_F(\boldsymbol{x}, \theta)$, are rarely computed analytically, thus rendering automatic differentiation the preferred method in (numerical) practice. The next result shows that $D_1 N_F(\boldsymbol{x}, \theta)$ and $D_2 N_F(\boldsymbol{x}, \theta)$ are in fact Fréchet derivatives and that the constraint qualification for the equality-constrained optimization problem (32) holds. Here, $L(\mathcal{U}, \mathcal{Q})$ denotes the space of linear and continuous operators from $\mathcal{U}$ to $\mathcal{Q}$, etc.

Proposition 4.1 (Constraint qualification). *Suppose that $\boldsymbol{\sigma}' \in L^\infty(\mathbb{R}^d)$ and the selection (29) holds. Then we have the following:*

1 $D_1 N_F(\boldsymbol{x}, \theta) \in L(\mathcal{Q}, \mathcal{Q})$ and $D_2 N_F(\boldsymbol{x}, \theta) \in L(\mathcal{U}, \mathcal{Q})$.

2 The Jacobian $e'(\boldsymbol{x}, \theta) \in L(\mathcal{W} \times \mathcal{U}, \mathcal{V})$ is surjective.

Proof. The linearity of $D_1 N_F(\boldsymbol{x}, \theta)$ straightforwardly follows from the linearity of its non-zero components in (33). Since $D_1 N_F(\boldsymbol{x}, \theta)$ is block-diagonal, we get

$$\|D_1 N_F(\boldsymbol{x}, \theta)\boldsymbol{v}\|_\mathcal{Q} \le \Big[\sum_{i=1}^m \|D_1 f(x^i, \theta)\|_{L^\infty(I)} \|v^i\|_{L^2(I)}\Big]^{1/2} \le C\|\boldsymbol{v}\|_\mathcal{Q}$$

for $\boldsymbol{v} \in \mathcal{Q}$, where the bounding constant is given by $C := \|\boldsymbol{\sigma}'\|_{L^\infty(\mathbb{R}^d)} \|W\|_\infty$. Indeed, such a finite constant $C \ge 0$ exists due to $\boldsymbol{\sigma}' \in L^\infty(\mathbb{R}^d)$, the continuous embedding $\mathcal{U} \hookrightarrow C([0, T])$ and the compactness of $[0, T]$. Hence $D_1 N_F(\boldsymbol{x}, \theta)$ defines a bounded linear map on $\mathcal{Q}$. Note that due to the continuous embedding $\mathcal{W} \hookrightarrow \mathcal{Q}$, $D_1 N_F(\boldsymbol{x}, \theta) \in L(\mathcal{W}, \mathcal{Q})$ as well. For $\xi \in \mathcal{U}$, we obtain the following estimate:

$$\|D_2 N_F(\boldsymbol{x}, \theta)\xi\|_\mathcal{Q} \le \Big[\sum_{i=1}^m \|D_2 f(x^i, \theta)\|_{L^\infty(I)} \|\xi^i\|_{L^2(I)}\Big]^{1/2} \le C\|\xi\|_\mathcal{U}.$$

Without going into the detailed structure of $D_2 f(x^i, \theta)$, it is clear that its components $\frac{\partial f(x^i, \theta)}{\partial b} \in \mathbb{R}^{d \times d}$ and $\frac{\partial f(x^i, \theta)}{\partial \operatorname{vec} W} \in \mathbb{R}^{d \times d^2}$ are bounded due to the boundedness of $\boldsymbol{\sigma}'$ and the boundedness of the continuous functions (4) on $[0, T]$. This yields finiteness of the bounding constant for the linear operator $D_2 N_F(\boldsymbol{x}, \theta)$ in the above estimate, completing the first claim.

Next, we show that the Jacobian $e'(\boldsymbol{x}, \theta) \in L(\mathcal{W} \times \mathcal{U}, \mathcal{V})$ is surjective. It is given by

$$e'(\boldsymbol{x}, \theta) = \begin{pmatrix} \partial_t - D_1 N_F(\boldsymbol{x}, \theta) & -D_2 N_F(\boldsymbol{x}, \theta) \\ \delta_0 & 0 \end{pmatrix},$$

where $\delta_0 \in \mathcal{W}^*$ is the Dirac delta distribution concentrated at 0. Equivalently, we need to show that for arbitrary $q = (q_1, q_0) \in \mathcal{V}$, the equation $e'(\boldsymbol{x}, \theta)\delta\boldsymbol{h} = q$ admits a solution $\delta\boldsymbol{h} = (\delta\boldsymbol{x}, \delta\theta) \in \mathcal{W} \times \mathcal{U}$. This equation can be written as

$$\delta\dot{\boldsymbol{x}} = D_1 N_F(\boldsymbol{x}, \theta)\delta\boldsymbol{x} + g_1, \quad \delta\boldsymbol{x}(0) = q_0, \tag{34}$$

where $g_1 := q_1 + D_2 N_F(\boldsymbol{x}, \theta)\delta\theta$ with $g_1 \in \mathcal{Q}$. Hence, it now suffices to show that, for any given $\delta\theta \in \mathcal{U}$, there exists a unique solution $\delta\boldsymbol{x} \in \mathcal{W}$ to the linear non-autonomous inhomogeneous ODE (34). Since the mapping $t \mapsto D_1 N_F(\boldsymbol{x}(t), \theta(t))$ belongs to $L^\infty(I; \mathbb{R}^{md \times md})$, one can construct a unique solution in $W^{1,\infty}(I; \mathbb{R}^{md})$ for the homogeneous counterpart of (34) (the case $g_1 = 0$) and further use a variation-of-constants approach to obtain a unique solution in $\mathcal{W}$ for (34); see [20, Chapter 5] for the details. $\square$

Concerning Proposition 4.1, we recall that the assumption $\boldsymbol{\sigma}' \in L^\infty(\mathbb{R}^d)$ holds not only for smooth activation functions, but for the ReLU function as well.

For our purposes, specifically for the subsequent Petrov–Galerkin discretization, we require a variational form of (11), which reads: For $\theta \in \mathcal{U}$, find $\boldsymbol{x} \in \mathcal{W}$ such that

$$\mathcal{F}(\boldsymbol{x}, \theta; \varphi) = 0, \quad \text{for all} \quad \varphi \in \mathcal{V}, \tag{35}$$

where the corresponding form $\mathcal{F} : \mathcal{W} \times \mathcal{U} \times \mathcal{V} \to \mathbb{R}$ is given by

$$\mathcal{F}(\boldsymbol{x}, \theta; \varphi) := \int_0^T \big(\dot{\boldsymbol{x}} - N_F(\boldsymbol{x}, \theta), \ \varphi_1\big) \, dt + \big(\boldsymbol{x}(0) - \boldsymbol{x}_{\text{in}}, \varphi_0\big) \tag{36}$$

with the test functions $\varphi = (\varphi_1, \varphi_0) \in \mathcal{V}$.

Remark 2. *We closely follow the notation introduced in [13], where the functionals, as the one given in (36), depend nonlinearly on the arguments before the semicolon and linearly on those after it.*

To characterize minimizers of (32), we derive an optimality system by considering the derivative of the corresponding Lagrangian $\mathcal{L} : \mathcal{W} \times \mathcal{U} \times \mathcal{V} \to \mathbb{R}$, which is given by

$$\mathcal{L}(\boldsymbol{x}, \theta; \boldsymbol{p}) := \mathcal{J}(\boldsymbol{x}, \theta) - \mathcal{F}(\boldsymbol{x}, \theta; \boldsymbol{p}), \tag{37}$$

where we used the variational form (36) of the constraint $e(\boldsymbol{x}, \theta)$. Proposition 3.3 guarantees the existence of an optimal pair $(\boldsymbol{x}^\star, \theta^\star) \in \mathcal{W} \times \mathcal{U}$. In addition, Proposition 4.1 ensures that $e'(\boldsymbol{x}^\star, \theta^\star)$ is surjective; therefore, there exists a unique Lagrange multiplier (adjoint state) $\boldsymbol{p}^\star := (\boldsymbol{z}^\star, \boldsymbol{q}^\star) \in \mathcal{V}$, where $\boldsymbol{z}^\star \in \mathcal{Q}$ and $\boldsymbol{q}^\star \in \mathbb{R}^{md}$, such that the following first-order optimality condition holds:

$$\mathcal{L}'(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, \varphi) = 0, \quad \forall \varphi \in \mathcal{W} \times \mathcal{U} \times \mathcal{V}, \tag{38}$$

or, equivalently, in component form:

$$
\begin{array}{llll}
\text{(adjoint equation)} & D_1\mathcal{J}(\boldsymbol{x}^\star, \theta^\star; \varphi_x) - D_1\mathcal{F}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, \varphi_x) & = 0, & \forall \varphi_x \in \mathcal{W}, \\
\text{(gradient equation)} & D_2\mathcal{J}(\boldsymbol{x}^\star, \theta^\star; \varphi_\theta) - D_2\mathcal{F}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, \varphi_\theta) & = 0, & \forall \varphi_\theta \in \mathcal{U}, \\
\text{(state equation)} & \mathcal{F}(\boldsymbol{x}^\star, \theta^\star; \varphi_p) & = 0, & \forall \varphi_p \in \mathcal{V}.
\end{array}
\tag{39}
$$

We now study the optimality system (39). From the first equation in (39), we derive a variational form of the adjoint equation: find $\boldsymbol{p} = (\boldsymbol{z}, \boldsymbol{q}) \in \mathcal{V}$ such that

$$\int_0^T \big(\dot{\boldsymbol{w}}, \boldsymbol{z}\big) \, dt - \int_0^T \big(D_1 N_F(\boldsymbol{x}, \theta)\boldsymbol{w}, \boldsymbol{z}\big) \, dt + \big(\boldsymbol{w}(0), \boldsymbol{q}\big) = \langle J'(\boldsymbol{x}), \boldsymbol{w}\rangle_{\mathcal{W}^*, \mathcal{W}}, \tag{40}$$

for all $\boldsymbol{w} \in \mathcal{W}$. Due to the chosen loss function (30), which depends only on the evaluation of the state $\boldsymbol{x} \in \mathcal{W}$ at the terminal time T, it holds that

$$\langle J'(\boldsymbol{x}), \boldsymbol{w}\rangle_{\mathcal{W}^*, \mathcal{W}} = \langle l'(\boldsymbol{x}(T)) \, \delta_T, \boldsymbol{w}\rangle_{\mathcal{W}^*, \mathcal{W}} = \big(l'(\boldsymbol{x}(T)), \boldsymbol{w}(T)\big), \tag{41}$$

where $\delta_T \in H^1(I; \mathbb{R}^{md})^*$ is the Dirac delta distribution concentrated at T and $l'(\boldsymbol{x}(T)) \in \mathbb{R}^{md}$. The following holds true for the adjoint variable $\boldsymbol{p} = (\boldsymbol{z}, \boldsymbol{q})$ in the adjoint equation (40), where we omit the $\star$ for better readability.

Proposition 4.2. *Suppose that $\boldsymbol{\sigma}' \in L^\infty(\mathbb{R}^d)$ and the selection (29) holds. Then we have the following:*

1 *The multiplier $\boldsymbol{z}$ satisfies $\boldsymbol{z} \in \mathcal{W}$.*

2 *The multiplier $\boldsymbol{q}$ is equal to $\boldsymbol{z}(0)$ and the adjoint equation (40) can be equivalently written as follows: find $\boldsymbol{z} \in \mathcal{W}$ such that*

$$-\int_0^T (\dot{\boldsymbol{z}}, \varphi_1)\, dt = \int_0^T \big(D_1 N_F(\boldsymbol{x}, \theta)^* \boldsymbol{z}, \varphi_1\big)\, dt, \tag{42}$$
$$(\boldsymbol{z}(T), \varphi_0) = (l'(\boldsymbol{x}(T)), \varphi_0)$$

for all $\varphi = (\varphi_1, \varphi_0) \in \mathcal{V}$.

Proof. We test (40) with $\boldsymbol{w} \in \mathcal{D}(I)$, where $\mathcal{D}(I) := C_c^\infty(I)$ is the space of smooth functions with compact support in I. Due to the compact support, the boundary term in (40) and the derivative (41) vanish, yielding

$$\int_0^T (\dot{\boldsymbol{w}}, \boldsymbol{z})\, dt = -\langle \dot{\boldsymbol{z}}, \boldsymbol{w} \rangle_{\mathcal{D}', \mathcal{D}} = \int_0^T \big(D_1 N_F(\boldsymbol{x}, \theta)^* \boldsymbol{z}, \boldsymbol{w}\big)\, dt, \tag{43}$$

where $\dot{\boldsymbol{z}} \in \mathcal{D}'(I)$ denotes the distributional time derivative of $\boldsymbol{z}$. However, $\dot{\boldsymbol{z}}$ is represented by $-D_1 N_F(\boldsymbol{x}, \theta)^* \boldsymbol{z}$. Since $D_1 N_F(\boldsymbol{x}, \theta) \in L(\mathcal{Q})$ implies $D_1 N_F(\boldsymbol{x}, \theta)^* \in L(\mathcal{Q})$, and $\boldsymbol{z} \in \mathcal{Q}$, we conclude that $\dot{\boldsymbol{z}} \in \mathcal{Q}$, which proves the claim.

For the second claim, we first apply generalized integration by parts to the first term in (40) with $\boldsymbol{w} \in C^\infty([0, T])$, which yields

$$\int_0^T (\boldsymbol{z}, \dot{\boldsymbol{w}})\, dt = \big(\boldsymbol{z}(T), \boldsymbol{w}(T)\big) - \big(\boldsymbol{z}(0), \boldsymbol{w}(0)\big) - \int_0^T (\dot{\boldsymbol{z}}, \boldsymbol{w})\, dt. \tag{44}$$

The traces of $\boldsymbol{z} \in H^1(I)$ in (44) are well-defined due to the continuity of the embedding $H^1(I) \hookrightarrow C([0, T])$, and the last term in (44) is meaningful, as the first claim implies that $\dot{\boldsymbol{z}}$ is also the weak derivative of $\boldsymbol{z}$. Combining (40), (41) and (44), we get

$$-\int_0^T (\dot{\boldsymbol{z}}, \boldsymbol{w})\, dt - \int_0^T \big(D_1 N_F(\boldsymbol{x}, \theta)^* \boldsymbol{z}, \boldsymbol{w}\big)\, dt + \big(\boldsymbol{q} - \boldsymbol{z}(0), \boldsymbol{w}(0)\big) + \big(\boldsymbol{z}(T) - l'(\boldsymbol{x}(T)), \boldsymbol{w}(T)\big) = 0.$$

The fundamental lemma of the calculus of variations yields $\boldsymbol{q} = \boldsymbol{z}(0)$. Finally, using the density of $C^\infty([0, T])$ in $\mathcal{Q}$, we obtain (42). $\qquad\square$

Remark 3. *Inspecting Proposition 4.2 reveals that the increased H^1 regularity of $\boldsymbol{z}$ depends not only on the regularity of θ, but also on the choice of the objective function. In our case, the objective (30) depends only on the terminal state $\boldsymbol{x}(T)$. However, if the objective involves the evaluation of $\boldsymbol{x}(\widetilde{T})$ at $\widetilde{T} < T$, one obtains*

$$\dot{\boldsymbol{z}} = -\big(D_1 N_F(\boldsymbol{x}, \theta)^* \boldsymbol{z} + l'(\boldsymbol{x}(\widetilde{T})) \delta_{\widetilde{T}}\big),$$

where $\delta_{\widetilde{T}} \in H^1(I; \mathbb{R}^{md})^*$ is the Dirac delta distribution concentrated at $\widetilde{T}$. Consequently, we expect that $\boldsymbol{z} \in L^2(I; \mathbb{R}^{md})$ only. We note that such objectives appear naturally in time-series modeling with neural ODEs; see [5] and references therein.

We henceforth identify the adjoint with its trajectory component and write $\boldsymbol{p} := \boldsymbol{z}$ and proceed by examining the structure of the gradient equation in (39). We recall that the Riesz representer of $D_2 \mathcal{L}(\boldsymbol{x}, \theta; \boldsymbol{p}, \cdot) \in \mathcal{U}^*$ is the gradient g of the reduced cost functional $\widehat{\mathcal{J}}(\theta) := \mathcal{J}(\boldsymbol{x}(\theta), \theta)$, where $\boldsymbol{p} = \boldsymbol{p}(\boldsymbol{x}(\theta))$; cf. [21]. That is, the gradient $g \in \mathcal{U}$ is the unique element satisfying

$$\langle g, \varphi_\theta \rangle_{\mathcal{U}} = \int_0^T \lambda \Big[(\theta, \varphi_\theta) + (\dot{\theta}, \dot{\varphi}_\theta)\Big] + \big(D_2 N_F(\boldsymbol{x}, \theta)^* \boldsymbol{p}, \varphi_\theta\big)\, dt, \quad \forall \varphi_\theta \in \mathcal{U}. \tag{45}$$

For convenience, we provide the strong form of (39), where we make use of (42) for the adjoint problem:

$$-\dot{\boldsymbol{p}}^\star = D_1 N_F(\boldsymbol{x}^\star, \theta^\star)^* \boldsymbol{p}^\star, \quad \boldsymbol{p}^\star(T) = l'(\boldsymbol{x}^\star(T)),$$
$$\lambda\big(\theta^\star - \ddot{\theta}^\star\big) = -D_2 N(\boldsymbol{x}^\star, \theta^\star)^* \boldsymbol{p}, \quad \dot{\theta}^\star(0) = \dot{\theta}^\star(T) = 0, \tag{46}$$
$$\dot{\boldsymbol{x}}^\star = N_F(\boldsymbol{x}^\star, \theta^\star), \quad \boldsymbol{x}^\star(0) = \boldsymbol{x}_{\mathsf{in}}.$$

For a given θ, the gradient g in (45) is then obtained from the two-point boundary value problem

$$g - \ddot{g} = \lambda(\theta - \ddot{\theta}) + D_2 N(\boldsymbol{x}, \theta)^* \boldsymbol{p}, \quad \dot{g}(0) = \dot{g}(T) = 0, \tag{47}$$

where $\boldsymbol{x}$ and $\boldsymbol{p}$ are the state and adjoint state trajectories associated to θ. The weak form of the problem (47) is then given in (45). This boundary value problem structure stems from the regularizer (14) inducing the $H^1(I)$ topology in which the gradient must now be computed.

Remark 4. *To include W_{in} and W_{out} as trainable parameters, we need the derivatives $D_3\mathcal{J}(\boldsymbol{x}, \theta, W_{in}, W_{out})$ and $D_4\mathcal{J}(\boldsymbol{x}, \theta, W_{in}, W_{out})$. The former is given by*

$$D_3\mathcal{J}(\boldsymbol{x}, \theta, W_{in}, W_{out}) = \sum_{i=1}^{m} \boldsymbol{p}^i(0)(\boldsymbol{x}_0^i)^\top.$$

$D_4\mathcal{J}(\boldsymbol{x}, \theta, W_{in}, W_{out})$ is independent of the neural vector field in (7) and depends only on the problem-specific loss (30) and the output mapping $\boldsymbol{q}_{\text{out}}$; it can therefore be computed using standard calculus.

5 Petrov-Galerkin discretization

In this section, we consider the discretization of the optimality system (39). We employ a discontinuous Petrov–Galerkin discretization for both the state and the adjoint state, taking advantage of the higher regularity of the adjoint variable, as established in Proposition 4.2. Both the state and adjoint discretizations lead to DG(0) schemes, which can be interpreted as the forward and backward Euler time-stepping schemes, respectively. For the discretization of the gradient equation (45), we use a CG(1) Galerkin approximation, where the piecewise-linear in time parametrization of the control parameters is consistent with the regularization scheme (14).

We start by performing a partition of the time interval $[0, T]$ as follows:

$$0 = t_0 < t_1 < \cdots < t_{K-1} < t_K = T,$$

and denote $I_k := (t_{k-1}, t_k)$, $\tau_k = t_k - t_{k-1}$, for all $k = 1, \cdots, K$, and $\tau := \max \tau_k$. We define the jump of a function v at instant t_k as $[\![v]\!]^k := v(t_k^+) - v(t_k^-)$, where

$$v(t_k^+) := \lim_{s \to 0^+} v(t_k + s), \quad v(t_k^-) := \lim_{s \to 0^+} v(t_k - s).$$

For the discretization of the trial space $\mathcal{W}$, we select the following discrete space:

$$\mathcal{W}_\tau := \{\boldsymbol{w}_\tau \in \mathcal{Q} : \boldsymbol{w}_\tau|_{[t_{k-1}, t_k]} \in \mathcal{P}_r([t_{k-1}, t_k); \mathbb{R}^{md}), \ \forall k = 1, \cdots, K, \ \boldsymbol{w}_\tau(T) \in \mathbb{R}^{md}\},$$

where $\mathcal{P}_r(I_k; \mathbb{R}^{md})$ is the space of polynomials on the interval I_k taking values in $\mathbb{R}^{md}$, with polynomial degree less than or equal to r. Note that $\mathcal{W}_\tau \not\subset \mathcal{W}$; that is, the above discretization is non-conforming for the trial space. However, we have $\mathcal{W}_\tau \subset \mathcal{V}$. For the discretization of the test space $\mathcal{V}$, we choose the following subspace of $\mathcal{V}$:

$$\mathcal{V}_\tau := \{\boldsymbol{v}_\tau \in \mathcal{Q} : \boldsymbol{v}_\tau|_{(t_{k-1}, t_k]} \in \mathcal{P}_r((t_{k-1}, t_k]; \mathbb{R}^{md}), \ \forall k = 1, \cdots, K, \ \boldsymbol{v}_\tau(0) \in \mathbb{R}^{md}\}.$$

Then, we have $\mathcal{V}_\tau \subset \mathcal{V}$; that is, the discretization is conforming in the test space. Note that $\mathcal{W}_\tau$ and $\mathcal{V}_\tau$ both have dimension $((r+1)K + 1)md$. For the discretization of the gradient equation (45) we use the conformal discrete space $\mathcal{U}_\tau \subset \mathcal{U}$, which we specify later in the section.

We define the discrete state equation with the discontinuous Galerkin form $\mathcal{F}_{\text{DG}} : \mathcal{W}_\tau \times \mathcal{U}_\tau \times \mathcal{V}_\tau \to \mathbb{R}$ (see Appendix A for the derivation): for $\theta_\tau \in \mathcal{U}_\tau$, find $\boldsymbol{x}_\tau \in \mathcal{W}_\tau$ such that

$$\mathcal{F}_{\text{DG}}(\boldsymbol{x}_\tau, \theta_\tau; \varphi) := \sum_{k=1}^{K} \int_{I_k} \left(\dot{\boldsymbol{x}}_\tau - N_F(\boldsymbol{x}_\tau, \theta), \varphi_1\right) dt + \left([\![\boldsymbol{x}_\tau]\!]^k, \varphi_1(t_k^-)\right) + \left(\boldsymbol{x}_\tau(0) - \boldsymbol{x}_{\text{in}}, \varphi_0\right) = 0 \tag{48}$$

for all $\varphi = (\varphi_1, \varphi_0) \in \mathcal{V}_\tau$. The formulation (48) yields $((r+1)K + 1)md$ nonlinear algebraic equations in the same number of unknowns. Indeed, the solvability of this system follows from the Lipschitz continuity of $F(\cdot, \theta)$, yielding the discrete control-to-state map

$$\mathfrak{S}_\tau : \mathcal{U}_\tau \to \mathcal{W}_\tau, \quad \theta_\tau \mapsto \boldsymbol{x}_\tau =: \mathfrak{S}_\tau(\theta_\tau), \tag{49}$$

which is continuous due to the continuity of the map $t \mapsto F(\cdot, \theta(t))$, see Remark 1. The existence of a discrete optimal control pair $(\boldsymbol{x}_\tau^\star, \theta_\tau^\star) \in \mathcal{W}_\tau \times \mathcal{U}_\tau$ can then be argued along the lines of Proposition 3.3, which is further simplified by the equivalence of weak and strong convergence in finite-dimensional spaces. By linearizing (48) at $(\boldsymbol{w}_\tau, \theta_\tau)$ as in Proposition 4.1, we obtain $((r+1)K+1)md$ linear algebraic equations, whose solvability yields the surjectivity of the Jacobian arising from the linearization. Then, a discrete stationary point $(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{p}_\tau^\star) \in \mathcal{W}_\tau \times \mathcal{U}_\tau \times \mathcal{V}_\tau$ exists and satisfies the discrete necessary optimality condition

$$\mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{p}_\tau^\star, \varphi) = 0, \quad \forall \varphi \in \mathcal{W}_\tau \times \mathcal{U}_\tau \times \mathcal{V}_\tau, \tag{50}$$

where $\mathcal{L}_{\mathrm{DG}}(\boldsymbol{x}_\tau, \theta_\tau; \boldsymbol{p}_\tau) := \mathcal{J}(\boldsymbol{x}_\tau, \theta_\tau) - \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau, \theta_\tau; \boldsymbol{p}_\tau)$. In component form we then have

$$\begin{aligned}
D_1 \mathcal{J}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \varphi_x) - D_1 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{p}_\tau^\star, \varphi_x) = 0, \quad &\forall \varphi_x \in \mathcal{W}_\tau, \\
D_2 \mathcal{J}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \varphi_\theta) - D_2 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{p}_\tau^\star, \varphi_\theta) = 0, \quad &\forall \varphi_\theta \in \mathcal{U}_\tau, \\
\mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \varphi_p) = 0, \quad &\forall \varphi_p \in \mathcal{V}_\tau.
\end{aligned} \tag{51}$$

The discrete adjoint equation for the optimality system (51) reads: find $\boldsymbol{p}_\tau \in \mathcal{V}_\tau$ such that

$$\sum_{k=1}^K \int_{I_k} \left(-\dot{\boldsymbol{p}}_\tau - D_1 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}, \varphi_x \right) dt - \sum_{k=1}^K \left([\![\boldsymbol{p}_\tau]\!]^{k-1}, \varphi_x(t_{k-1}^+) \right) = 0$$
$$(\boldsymbol{p}_\tau(T), \varphi_x(T)) = \left(l'(\boldsymbol{x}_\tau(T), \varphi_x(T) \right), \tag{52}$$

for all $\varphi_x \in \mathcal{W}_\tau$. The derivation of (52) is provided in Appendix A. We then consider a discretization that yields an explicit Euler scheme forward in time for the discrete state equation in (51) and backward in time for the discrete adjoint equation (52), which we state later in the section. Specifically, we use piecewise-constant in time polynomial functions, corresponding to $r = 0$ in $\mathcal{W}_\tau$ and $\mathcal{V}_\tau$, cf. also [22]. We refer to Appendix B for derivation details of this time-marching interpretation of our DG(0) scheme.

Let us now proceed with the numerical discretization of the gradient equation (45). Let $\mathcal{U}_\tau \subset \mathcal{U}$ denote the finite-dimensional space of piecewise-linear in time finite-element functions with values in $\mathbb{R}^n$:

$$\mathcal{U}_\tau := \left\{ u_\tau \in \mathcal{U} : \ u_\tau|_{[t_{k-1}, t_k]} \in \mathcal{P}_1([t_{k-1}, t_k]; \mathbb{R}^n), \ k = 1, \dots, K \right\} \cong \mathcal{T}_\tau \otimes \mathbb{R}^n,$$

where $\mathcal{T}_\tau \subset H^1(I)$ consists of piecewise-linear, globally continuous $\mathbb{R}$-valued functions, and is spanned by the well-known nodal hat functions $\{\phi^i\}_{i=0}^K$ [23]. Both boundary nodes $i = 0$ and $i = K$ are included due to the Neumann boundary conditions in (47). In view of the above, $g_\tau \in \mathcal{U}_\tau$ and $\theta_\tau \in \mathcal{U}_\tau$ are given by

$$g_\tau(t) = \sum_{k=0}^K \sum_{i=1}^n g_{\tau,i}^k \, \phi^k \otimes e_i \in \mathcal{U}_\tau, \quad \theta_\tau(t) = \sum_{k=0}^K \sum_{i=1}^n \theta_{\tau,i}^k \, \phi^k \otimes e_i.$$

We use g_τ as the discrete trial function and θ_τ as data in (45), and we test (45) with the basis functions $\phi^l \otimes e_j$, $0 \leq l \leq K$, $1 \leq j \leq n$. This way, the left-hand side in (45) is discretized as follows:

$$\langle g_\tau, \phi^l \otimes e_j \rangle_{\mathcal{U}} = \sum_{i=0}^K \sum_{i=1}^n g_{\tau,i}^k \langle \phi^k \otimes e_i, \phi^l \otimes e_j \rangle_{\mathcal{U}} = \sum_{k=0}^K \sum_{i=1}^n g_{\tau,i}^k \langle \phi^k, \phi^l \rangle_{H^1(I)} \langle e_i, e_j \rangle_{\mathbb{R}^n}.$$

For the first summand on the right-hand side of (45), we get

$$\langle \mathcal{R}'(\theta), \phi^l \otimes e_j \rangle_{\mathcal{U}^*, \mathcal{U}} = \lambda \sum_{k=0}^K \sum_{i=1}^n \theta_{\tau,i}^k \langle \phi^k, \phi^l \rangle_{H^1(I)} \langle e_i, e_j \rangle_{\mathbb{R}^n}.$$

Recall that $L^2(I; \mathbb{R}^n) \cong L^2(I) \otimes \mathbb{R}^n$. Our approximation of the second summand on the right-hand side of (45) is then given by

$$\langle D_2 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}_\tau, \phi^l \otimes e_j \rangle_{L^2(I; \mathbb{R}^n)} \approx \sum_{0=1}^K \sum_{i=1}^n z_{\tau,i}^k \, \langle \phi^k, \phi^l \rangle_{L^2(I)} \langle e_i, e_j \rangle_{\mathbb{R}^n},$$

where $z_\tau^k = D_2 F(\boldsymbol{x}_\tau^{k-1}, \theta_\tau^k)^* \widehat{\boldsymbol{p}}_\tau^k \in \mathbb{R}^n$, and $\widehat{\boldsymbol{p}}_\tau$ is the piecewise-linear and continuous reconstruction of $\boldsymbol{p}_\tau$ based on the midpoints m_k of I_k. In fact, on each interval $[m_k, m_{k+1}]$ the reconstruction $\widehat{\boldsymbol{p}}_\tau$ is defined as

$$\widehat{\boldsymbol{p}}_\tau(t) = \frac{m_{k+1} - t}{m_{k+1} - m_k} \boldsymbol{p}_\tau(m_k) + \frac{t - m_k}{m_{k+1} - m_k} \boldsymbol{p}_\tau(m_{k+1}), \quad t \in [m_k, m_{k+1}]. \tag{53}$$

Since $\boldsymbol{p}_\tau(m_k) = \boldsymbol{p}_\tau^k$ and $\boldsymbol{p}_\tau(m_{k+1}) = \boldsymbol{p}_\tau^{k+1}$, it holds $\widehat{\boldsymbol{p}}_\tau^k = \widehat{\boldsymbol{p}}_\tau(t^k) = \frac{\tau_{k+1}}{\tau_k + \tau_{k+1}} \boldsymbol{p}_\tau^k + \frac{\tau_k}{\tau_k + \tau_{k+1}} \boldsymbol{p}_\tau^{k+1}$. Practically, z_τ^k is computed using backward mode automatic differentiation. These steps yield the discrete gradient equation in its algebraic form:

$$\left(\mathbf{B}^\tau \otimes \mathbf{I}_n\right) \mathbf{g}_\tau = \lambda\left(\mathbf{B}^\tau \otimes \mathbf{I}_n\right) \boldsymbol{\Theta}_\tau + \left(\mathbf{M}^\tau \otimes \mathbf{I}_n\right) \mathbf{z}_\tau^{\boldsymbol{p}}, \tag{54}$$

where $\mathbf{B}^\tau = \mathbf{A}^\tau + \mathbf{M}^\tau$, and $\mathbf{A}^\tau, \mathbf{M}^\tau \in \mathbb{R}^{(K+1) \times (K+1)}$ are the stiffness and the mass matrices of $\mathcal{Q}_\tau$-basis functions with the respective entries $\mathbf{A}_{k,l}^\tau = (\dot{\phi}^k, \dot{\phi}^l)_{L^2(I)}$ and $\mathbf{M}_{k,l}^\tau = (\phi^k, \phi^l)_{L^2(I)}$, and $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix. The vectors $\mathbf{g}_\tau, \boldsymbol{\Theta}_\tau, \mathbf{z}_\tau^{\boldsymbol{p}} \in \mathbb{R}^{(K+1)n}$ are given by

$$\mathbf{g}_\tau = (g_\tau^0, \dots, g_\tau^K)^\top, \quad \boldsymbol{\Theta}_\tau = (\theta_\tau^0, \dots, \theta_\tau^K)^\top, \quad \mathbf{z}_\tau^{\boldsymbol{p}} = (\boldsymbol{z}_\tau^0, \dots, \boldsymbol{z}_\tau^K)^\top.$$

Summarizing, the discrete Riesz gradient $\mathbf{g}_\tau$ is computed via the following system:

$$\begin{aligned}
\boldsymbol{x}_\tau^k &= \boldsymbol{x}_\tau^{k-1} + \tau_k F\left(\boldsymbol{x}_\tau^{k-1}, \theta_\tau^{k-1/2}\right), \quad k \in \{1, \cdots, K\}, \\
\boldsymbol{x}_\tau^0 &= \boldsymbol{x}_{\mathrm{in}}, \\
\boldsymbol{p}_\tau^{k-1} &= \boldsymbol{p}_\tau^k + \tau_k D_1 F\left(\boldsymbol{x}_\tau^{k-1}, \theta_\tau^{k-1/2}\right)^* \boldsymbol{p}_\tau^k, \quad k \in \{1, \cdots, K-1\}, \\
\boldsymbol{p}_\tau^K &= l'(\boldsymbol{x}_\tau^K), \\
\left(\mathbf{B}^\tau \otimes \mathbf{I}_n\right) \mathbf{g}_\tau &= \lambda\left(\mathbf{B}^\tau \otimes \mathbf{I}_n\right) \boldsymbol{\Theta}_\tau + \left(\mathbf{M}^\tau \otimes \mathbf{I}_n\right) \mathbf{z}_\tau^{\boldsymbol{p}},
\end{aligned} \tag{55}$$

where $\theta_\tau^{k-1/2} = \theta_\tau(m_k)$. The above system is solved iteratively to find an approximation of the discrete stationary point $(\boldsymbol{x}_\tau^\star, \theta_\tau^\star, \boldsymbol{p}_\tau^\star)$ of (51). Given some initialization of parameters θ_τ, the forward Euler scheme is used to obtain $\boldsymbol{x}_\tau(T)$, which is further used to compute $l'(\boldsymbol{x}_\tau^K)$ to initialize the backward Euler scheme for the adjoint approximation. The latter is then used together with the parameters θ_τ to assemble the right-hand side of the discrete gradient equation (54), which is subsequently solved for $\mathbf{g}_\tau$. The gradient $\mathbf{g}_\tau$ is then used in an optimizer of choice to update θ_τ. This procedure is repeated until the loss value or the validation error reaches a satisfactory threshold, or the overall number of specified iterations is reached.

6 The dual-weighted residual error estimation

In this section, we derive an a-posteriori error bound based on the analysis of the continuous and discrete optimality systems, which will be used for the layerwise-adaptive construction of neural network architectures derived from the discretization of neural ODEs.

First, we note that in view of Proposition 4.2, every solution $(\boldsymbol{x}^\star, \theta^\star, \boldsymbol{p}^\star)$ of the optimality system (39) also solves

$$\mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, \varphi) = 0, \quad \forall \varphi \in \mathcal{V} \times \mathcal{U} \times \mathcal{V}. \tag{56}$$

Component-wise, (56) is given by:

$$\begin{aligned}
\text{(adjoint equation)} \quad & D_1 \mathcal{J}(\boldsymbol{x}^\star, \theta^\star; \varphi_x) - D_1 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, \varphi_x) && = 0, \quad \forall \varphi_x \in \mathcal{V}, \\
\text{(gradient equation)} \quad & D_2 \mathcal{J}(\boldsymbol{x}^\star, \theta^\star; \varphi_\theta) - D_2 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, \varphi_\theta) && = 0, \quad \forall \varphi_\theta \in \mathcal{U}, \\
\text{(state equation)} \quad & \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \varphi_p) && = 0, \quad \forall \varphi_p \in \mathcal{V}.
\end{aligned} \tag{57}$$

Indeed, because $\boldsymbol{x}^*$ and $\boldsymbol{p}^*$ are continuous in time, the jump terms in the above forms vanish. Thanks to this formulation, $\boldsymbol{x}_\tau^* \in \mathcal{W}_\tau$ and $\boldsymbol{p}_\tau^* \in \mathcal{V}_\tau$ become admissible trial functions in the state and the adjoint equation of (57), respectively, even though $\mathcal{W}_\tau \not\subset \mathcal{W}$ and $\mathcal{V}_\tau \not\subset \mathcal{W}$.

We further observe that due to the inclusions $\mathcal{W}_\tau \subset \mathcal{V}$, $\mathcal{U}_\tau \subset \mathcal{U}$ and $\mathcal{V}_\tau \subset \mathcal{V}$, the error $e = (\boldsymbol{x}^\star - \boldsymbol{x}_\tau, \theta^\star - \theta_\tau, \boldsymbol{p}^\star - \boldsymbol{p}_\tau) \in \mathcal{V} \times \mathcal{U} \times \mathcal{V}$ is a feasible test function in (57). Substituting it into (57) yields the Galerkin orthogonality condition

$$\mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, e) = 0. \tag{58}$$

With the continuous optimality system (57) and its discrete counterpart (51) at our disposal, we adapt [13, Theorem 2.2] to the neural ODE setting and provide an a posteriori representation formula for the error in the functional $\mathcal{J}$.

Proposition 6.1. *Suppose that $\boldsymbol{\sigma}''' \in L^\infty(\mathbb{R}^d)$. Let $(\boldsymbol{x}^\star, \theta^\star, \boldsymbol{p}^\star) \in \mathcal{W} \times \mathcal{U} \times \mathcal{W}$ and $(\boldsymbol{x}^\star_\tau, \theta^\star_\tau, \boldsymbol{p}^\star_\tau) \in \mathcal{W}_\tau \times \mathcal{U}_\tau \times \mathcal{V}_\tau$ be solutions of* (57) *and* (51), *respectively. Then*

$$\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) - \mathcal{J}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau) = \frac{1}{2}\rho_{\boldsymbol{x}} + \frac{1}{2}\rho_\theta + \frac{1}{2}\rho_{\boldsymbol{p}} + R. \tag{59}$$

The residuals $\rho_{\boldsymbol{x}}$, ρ_θ and $\rho_{\boldsymbol{p}}$ are given by

$$\rho_{\boldsymbol{x}} = \sum_{k=1}^{K} \int_{I_k} \left(-\dot{\boldsymbol{p}}^\star_\tau - D_1 N_F(\boldsymbol{x}^\star_\tau, \theta^\star_\tau)^* \boldsymbol{p}^\star_\tau, \ \boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_\tau \right) dt - \sum_{k=1}^{K} \left([\![\boldsymbol{p}^\star_\tau]\!]^{k-1}, \ \boldsymbol{x}^\star(t^+_{k-1}) - \tilde{\boldsymbol{x}}_\tau(t^+_{k-1}) \right),$$

$$\rho_\theta = \sum_{k=1}^{K} \int_{I_k} \lambda \left[\left(\theta^\star_\tau, \theta^\star - \tilde{\theta}_\tau \right) + \left(\dot{\theta}^\star_\tau, \dot{\theta}^\star - \dot{\tilde{\theta}}_\tau \right) \right] + \left(D_2 N_F(\boldsymbol{x}^\star_\tau, \theta^\star_\tau)^* \boldsymbol{p}^\star_\tau, \ \theta^\star - \tilde{\theta}_\tau \right) dt, \tag{60}$$

$$\rho_{\boldsymbol{p}} = \sum_{k=1}^{K} \int_{I_k} \left(\dot{\boldsymbol{x}}^\star_\tau - N_F(\boldsymbol{x}^\star_\tau, \theta^\star_\tau), \ \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \right) dt + \sum_{k=1}^{K} \left([\![\boldsymbol{x}^\star_\tau]\!]^k, \ \boldsymbol{p}^\star(t^-_k) - \tilde{\boldsymbol{p}}_\tau(t^-_k) \right),$$

where $\tilde{\boldsymbol{x}}_\tau \in \mathcal{W}_\tau$, $\tilde{\boldsymbol{p}}_\tau \in \mathcal{V}_\tau$ and $\tilde{\theta}_\tau \in \mathcal{U}_\tau$ are arbitrary, and R is the remainder

$$R = \frac{1}{2} \int_0^1 \left(\mathcal{J}'''(\boldsymbol{x}^\star_\tau + s e_{\boldsymbol{x}}, \theta^\star_\tau + s e_\theta; e, e, e) - \mathcal{F}'''(\boldsymbol{x}^\star_\tau + s e_{\boldsymbol{x}}, \theta^\star_\tau + s e_\theta; \boldsymbol{p}^\star_\tau + s e_{\boldsymbol{p}}, e, e, e) \right) s(s-1) \ ds, \tag{61}$$

which is cubic in the error $e = (\boldsymbol{x}^\star - \boldsymbol{x}^\star_\tau, \theta^\star - \theta^\star_\tau, \boldsymbol{p}^\star - \boldsymbol{p}^\star_\tau) \in \mathcal{V} \times \mathcal{U} \times \mathcal{V}$.

Proof. Using the third equation in (57) and the third equation in (51), we obtain

$$\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) - \mathcal{J}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau) = \mathcal{L}_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star) + \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star) - \mathcal{L}_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau) - \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau)$$

$$= \mathcal{L}_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star) - \mathcal{L}_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau)$$

$$= \int_0^1 \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau + s e_{\boldsymbol{x}}, \theta^\star_\tau + s e_\theta; \boldsymbol{p}^\star_\tau + s e_{\boldsymbol{p}}, e) \ ds + \frac{1}{2} \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau, e)$$

$$- \frac{1}{2} \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau, e) - \frac{1}{2} \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, e),$$

where the last term is zero in view of (58), and the first and the last two terms in the above expression form the remainder of the trapezoidal quadrature rule:

$$R = \int_0^1 \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau + s e_{\boldsymbol{x}}, \theta^\star_\tau + s e_\theta; \boldsymbol{p}^\star_\tau + s e_{\boldsymbol{p}}, e) \ ds - \frac{1}{2} \left[\mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau, e) + \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star, \theta^\star; \boldsymbol{p}^\star, e) \right].$$

Therefore, we get the following equality

$$\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) - \mathcal{J}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau) = \frac{1}{2} \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau, \boldsymbol{x}^\star - \boldsymbol{x}^\star_\tau, \theta^\star - \theta^\star_\tau, \boldsymbol{p}^\star - \boldsymbol{p}^\star_\tau) + R$$

$$= \frac{1}{2} \mathcal{L}'_{\mathrm{DG}}(\boldsymbol{x}^\star_\tau, \theta^\star_\tau; \boldsymbol{p}^\star_\tau, \boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_\tau, \theta^\star - \tilde{\theta}_\tau, \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau) + R, \tag{62}$$

where we replaced $(\boldsymbol{x}_\tau^\star, \theta_\tau^\star, \boldsymbol{p}_\tau^\star)$ by an arbitrary $(\tilde{\boldsymbol{x}}_\tau, \tilde{\theta}_\tau, \tilde{\boldsymbol{p}}_\tau) \in \mathcal{W}_\tau \times \mathcal{U}_\tau \times \mathcal{V}_\tau$ in view of (50). Writing (62) in terms of individual components yields

$$\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) - \mathcal{J}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star) = \frac{1}{2}\Big[D_1 \mathcal{J}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_\tau) - D_1 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{p}_\tau^\star, \boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_\tau) \Big]$$
$$+ \frac{1}{2}\Big[D_2 \mathcal{J}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \theta^\star - \tilde{\theta}_\tau) - D_2 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{p}_\tau^\star, \theta^\star - \tilde{\theta}_\tau) \Big]$$
$$+ \frac{1}{2} \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star; \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau) + R = \frac{1}{2}\rho_{\boldsymbol{x}} + \frac{1}{2}\rho_\theta + \frac{1}{2}\rho_{\boldsymbol{p}} + R.$$

Expressing the remainder R using the Peano kernel formula for the trapezoidal quadrature rule yields

$$R = \frac{1}{2} \int_0^1 \mathcal{L}_{\mathrm{DG}}'''(\boldsymbol{x}_\tau^\star + s e_{\boldsymbol{x}}, \theta_\tau^\star + s e_\theta; \boldsymbol{p}_\tau^\star + s e_{\boldsymbol{p}}, e, e, e) s(s-1) \, ds.$$

Expanding the trilinear form $\mathcal{L}_{\mathrm{DG}}'''(\boldsymbol{x}_\tau^\star + s e_{\boldsymbol{x}}, \theta_\tau^\star + s e_\theta; \boldsymbol{p}_\tau^\star + s e_{\boldsymbol{p}}, \cdot, \cdot, \cdot)$ in the above integrand in terms of its components yields the remainder (61). The assumption $\boldsymbol{\sigma}''' \in L^\infty(\mathbb{R}^d)$ then guarantees that the remainder remains bounded. $\qquad\square$

For our purposes, we use the following interval norm

$$\|f\|_{I_k} = \sup_{t \in I_k} \|f(t)\|. \tag{63}$$

Using the error representation formula (59), we proceed by estimating the error in the objective.

Proposition 6.2. *Suppose that* $\boldsymbol{\sigma}''' \in L^\infty(\mathbb{R}^d)$. *Let* $(\boldsymbol{x}^\star, \theta^\star, \boldsymbol{p}^\star) \in \mathcal{W} \times \mathcal{U} \times \mathcal{W}$ *and* $(\boldsymbol{x}_\tau^\star, \theta_\tau^\star, \boldsymbol{p}_\tau^\star) \in \mathcal{W}_\tau \times \mathcal{U}_\tau \times \mathcal{V}_\tau$ *be solutions of* (57) *and* (51), *respectively. Then*

$$|\mathcal{J}(\boldsymbol{x}^\star, \theta^\star) - \mathcal{J}(\boldsymbol{x}_\tau^\star, \theta_\tau^\star)| \leq \frac{1}{2}|\rho_{\boldsymbol{x}}| + \frac{1}{2}|\rho_\theta| + \frac{1}{2}|\rho_{\boldsymbol{p}}| + |R| \tag{64}$$

with the following corresponding estimates

$$|\rho_{\boldsymbol{x}}| \leq \sum_{k=1}^{K} R_{\boldsymbol{p}_\tau^\star}^k \omega_{\boldsymbol{x}^\star}^k, \quad |\rho_\theta| \leq \sum_{k=1}^{K} |\rho_{\theta^\star, \theta_\tau^\star}^k|, \quad |\rho_{\boldsymbol{p}}| \leq \sum_{k=1}^{K} R_{\boldsymbol{x}_\tau^\star}^k \omega_{\boldsymbol{p}^\star}^k, \tag{65}$$

where the residuals are given by

$$R_{\boldsymbol{x}_\tau^\star}^k = \tau_k \|\dot{\boldsymbol{x}}_\tau^\star - N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star)\|_{I_k} + \frac{\tau_k}{\tau_k + \tau_{k+1}}\|[\![\boldsymbol{x}_\tau^\star]\!]^k\| + \mathbb{I}_{\{2 \leq k \leq K-1\}}\frac{\tau_k}{\tau_k + \tau_{k-1}}\|[\![\boldsymbol{x}_\tau^\star]\!]^{k-1}\|,$$

$$\rho_{\theta^\star, \theta_\tau^\star}^k = \int_{I_k} \lambda\Big[(\theta_\tau^\star, \theta^\star - \tilde{\theta}_\tau) + (\dot{\theta}_\tau^\star, \dot{\theta}^\star - \dot{\tilde{\theta}}_\tau)\Big] + (D_2 N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star)^* \boldsymbol{p}_\tau^\star, \theta^\star - \tilde{\theta}_\tau) \, dt, \tag{66}$$

$$R_{\boldsymbol{p}_\tau^\star}^k = \tau_k \|-\dot{\boldsymbol{p}}_\tau^\star - D_1 N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star)^* \boldsymbol{p}_\tau^\star\|_{I_k} + \frac{\tau_k}{\tau_{k-1} + \tau_k}\|[\![\boldsymbol{p}_\tau^\star]\!]^{k-1}\| + \mathbb{I}_{\{1 \leq k \leq K-1\}}\frac{\tau_k}{\tau_k + \tau_{k+1}}\|[\![\boldsymbol{p}_\tau^\star]\!]^k\|,$$

where $\mathbb{I}_{\mathcal{C}}$ *is the indicator function of an integer set* $\mathcal{C}$, *and the weights are given by:*

$$\omega_{\boldsymbol{x}^\star}^k = \|\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_\tau\|_{I_k}, \quad \omega_{\boldsymbol{p}^\star}^k = \|\boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau\|_{I_k}, \tag{67}$$

where $\tilde{\boldsymbol{x}}_\tau \in \mathcal{W}_\tau$, $\tilde{\theta}_\tau \in \mathcal{U}_\tau$ *and* $\tilde{\boldsymbol{p}}_\tau \in \mathcal{V}_\tau$ *are arbitrary.*

Proof. The estimate (64) is straightforward to obtain from the error representation formula (59). We now estimate the terms in (60), and begin with $\rho_{\boldsymbol{x}}$. Using the Cauchy-Schwarz inequality, we get

$$|\rho_{\boldsymbol{x}}| \leq \sum_{k=1}^{K} \tau_k \|-\dot{\boldsymbol{p}}_\tau^\star - D_1 N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star)^* \boldsymbol{p}_\tau^\star\|_{I_k}\|\boldsymbol{x}^\star - \tilde{\boldsymbol{x}}_\tau\|_{I_k} + \sum_{k=1}^{K}\|[\![\boldsymbol{p}_\tau^\star]\!]^{k-1}\|\|\boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+)\|$$

From the continuity of $\boldsymbol{x}^\star$, we get

$$\|\boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+)\| \leq \|\boldsymbol{x}^\star - \boldsymbol{x}_\tau^\star\|_{I_{k-1}}, \quad \|\boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+)\| \leq \|\boldsymbol{x}^\star - \boldsymbol{x}_\tau^\star\|_{I_k}.$$

Utilizing the convex combination

$$\left\| \frac{\tau_{k-1}}{\tau_{k-1} + \tau_k} \left(\boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+) \right) + \frac{\tau_k}{\tau_{k-1} + \tau_k} \left(\boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+) \right) \right\|$$

$$\leq \frac{\tau_{k-1}}{\tau_{k-1} + \tau_k} \left\| \left(\boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+) \right) \right\| + \frac{\tau_k}{\tau_{k-1} + \tau_k} \left\| \left(\boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+) \right) \right\|$$

$$\leq \frac{\tau_{k-1}}{\tau_{k-1} + \tau_k} \| \boldsymbol{x}^\star - \boldsymbol{x}_\tau^\star \|_{I_{k-1}} + \frac{\tau_k}{\tau_{k-1} + \tau_k} \| \boldsymbol{x}^\star - \boldsymbol{x}_\tau^\star \|_{I_k},$$

and rearranging indices, we obtain the following estimate

$$\sum_{k=1}^{K} \| [\![\boldsymbol{p}_\tau^\star]\!]^{k-1} \| \| \boldsymbol{x}^\star(t_{k-1}^+) - \boldsymbol{x}_\tau^\star(t_{k-1}^+) \| \leq \sum_{k=1}^{K-1} \left(\frac{\tau_k}{\tau_{k-1} + \tau_k} \| [\![\boldsymbol{p}_\tau^\star]\!]^{k-1} \| \right.$$

$$\left. + \frac{\tau_k}{\tau_k + \tau_{k+1}} \| [\![\boldsymbol{p}_\tau^\star]\!]^{k} \| \right) \| \boldsymbol{x}^\star - \boldsymbol{x}_\tau^\star \|_{I_k} + \frac{\tau_K}{\tau_{K-1} + \tau_K} \| [\![\boldsymbol{p}_\tau^\star]\!]^{K-1} \| \| \boldsymbol{x}^\star - \boldsymbol{x}_\tau \|_{I_K},$$

where $\tau_0 = 0$. Therefore,

$$|\rho_{\boldsymbol{x}}| \leq \sum_{k=1}^{K-1} \left(\tau_k \, \| -\dot{\boldsymbol{p}}_\tau^\star - D_1 N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star)^* \boldsymbol{p}_\tau^\star \|_{I_k} + \frac{\tau_k}{\tau_{k-1} + \tau_k} \| [\![\boldsymbol{p}_\tau^\star]\!]^{k-1} \| + \frac{\tau_k}{\tau_k + \tau_{k+1}} \| [\![\boldsymbol{p}_\tau^\star]\!]^{k} \| \right) \| \boldsymbol{x}^\star - \boldsymbol{x}_\tau \|_{I_k}$$

$$+ \left(\tau_K \, \| -\dot{\boldsymbol{p}}_\tau^\star - D_1 N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star)^* \boldsymbol{p}_\tau^\star \|_{I_K} + \frac{\tau_K}{\tau_{K-1} + \tau_K} \| [\![\boldsymbol{p}_\tau^\star]\!]^{K-1} \| \right) \| \boldsymbol{x}^\star - \boldsymbol{x}_\tau \|_{I_K} =: \sum_{k=1}^{K} R_{\boldsymbol{p}_\tau^\star}^{k} \, \omega_{\boldsymbol{x}^\star}^{k}.$$

This yields the first term in (65). We proceed by estimating $\rho_{\boldsymbol{p}}$, and apply the Cauchy-Schwarz inequality

$$|\rho_{\boldsymbol{p}}| \leq \sum_{k=1}^{K} \tau_k \| \dot{\boldsymbol{x}}_\tau^\star - N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star) \|_{I_k} \| \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \|_{I_k} + \sum_{k=1}^{K} \| [\![\boldsymbol{x}_\tau^\star]\!]^{k} \| \| \boldsymbol{p}^\star(t_k^-) - \tilde{\boldsymbol{p}}_\tau(t_k^-) \|$$

From the continuity of $\boldsymbol{p}^\star$, which follows from Proposition 4.2, we get

$$\| \boldsymbol{p}^\star(t_k^-) - \tilde{\boldsymbol{p}}_\tau(t_k^-) \| \leq \| \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \|_{I_k}, \quad \| \boldsymbol{p}^\star(t_k^-) - \tilde{\boldsymbol{p}}_\tau(t_k^-) \| \leq \| \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \|_{I_{k+1}}.$$

This yields the following bound

$$\| \boldsymbol{p}^\star(t_k^-) - \tilde{\boldsymbol{p}}_\tau(t_k^-) \| \leq \frac{\tau_k}{\tau_k + \tau_{k+1}} \| \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \|_{I_k} + \frac{\tau_{k+1}}{\tau_k + \tau_{k+1}} \| \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \|_{I_{k+1}},$$

which we use to obtain the estimate of the jump term

$$\sum_{k=1}^{K} \| [\![\boldsymbol{x}_\tau^\star]\!]^{k} \| \| \boldsymbol{p}^\star(t_k^-) - \tilde{\boldsymbol{p}}_\tau(t_k^-) \| \leq \frac{\tau_1}{\tau_1 + \tau_2} \| [\![\boldsymbol{x}_\tau^\star]\!]^{1} \| \| \boldsymbol{p}^\star - \boldsymbol{p}_\tau \|_{I_1} + \sum_{k=2}^{K} \left(\frac{\tau_k}{\tau_k + \tau_{k+1}} \| [\![\boldsymbol{x}_\tau]\!]^{k} \| \right.$$

$$\left. + \frac{\tau_k}{\tau_k + \tau_{k-1}} \| [\![\boldsymbol{x}_\tau^\star]\!]^{k-1} \| \right) \| \boldsymbol{p}^\star - \boldsymbol{p}_\tau \|_{I_k}.$$

Therefore, we get

$$|\rho_{\boldsymbol{p}}| \leq \left(\tau_1 \| \dot{\boldsymbol{x}}_\tau^\star - N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star) \|_{I_1} + \frac{\tau_1}{\tau_1 + \tau_2} \| [\![\boldsymbol{x}_\tau^\star]\!]^{1} \| \right) \| \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \|_{I_1} + \sum_{k=2}^{K} \left(\tau_k \| \dot{\boldsymbol{x}}_\tau^\star - N_F(\boldsymbol{x}_\tau^\star, \theta_\tau^\star) \|_{I_k} \right.$$

$$\left. + \frac{\tau_k}{\tau_k + \tau_{k+1}} \| [\![\boldsymbol{x}_\tau^\star]\!]^{k} \| + \frac{\tau_k}{\tau_k + \tau_{k-1}} \| [\![\boldsymbol{x}_\tau^\star]\!]^{k-1} \| \right) \| \boldsymbol{p}^\star - \tilde{\boldsymbol{p}}_\tau \|_{I_k} = \sum_{k=1}^{K} R_{\boldsymbol{x}_\tau^\star}^{k} \, \omega_{\boldsymbol{p}^\star}^{k}.$$

This yields the last term in the estimator (65). The estimate of ρ_θ is obtained by directly distributing the residual in (60) over the time intervals. This completes the proof. $\square$

We note that Proposition 6.2 requires sufficiently regular activation functions. For example, we use the smooth hyperbolic tangent in our numerical examples. Less regular activation functions can, in principle, be approximated (or smoothed) to meet the required assumptions; see, e.g., [24].

7 Layerwise adaptive algorithm

In this section, we present a depth-adaptive algorithm that relies on the error estimate (64) for the objective functional, and provide the details for its computer implementation.

By neglecting the remainder in (64), we obtain the following error indicator

$$\triangle = \frac{1}{2} \sum_{k=1}^{K} \left(R_{\boldsymbol{p}_\tau^\star}^k \omega_{\boldsymbol{x}^\star}^k + \rho_\theta^k + R_{\boldsymbol{x}_\tau^\star}^k \omega_{\boldsymbol{p}^\star}^k \right) =: \frac{1}{2} \sum_{k=1}^{K} \triangle_k, \tag{68}$$

where the global indicator is decomposed into its local contributions $\triangle_k$ on the intervals $I_k \subset [0,T]$. The residuals $R_{\boldsymbol{p}_\tau^\star}^k$ and $R_{\boldsymbol{x}_\tau^\star}^k$ require the discrete solution $(\boldsymbol{x}_\tau^\star, \theta_\tau^\star, \boldsymbol{p}_\tau^\star)$ of (51), and we replace it with its approximation $(\boldsymbol{x}_\tau, \theta_\tau, \boldsymbol{p}_\tau)$ obtained from (55). With our choice of discrete spaces, we get

$$R_{\boldsymbol{x}_\tau^\star}^k \approx \tau_k \|F(\boldsymbol{x}_\tau, \theta_\tau)\|_{I_k} + \frac{\tau_k}{\tau_k + \tau_{k+1}} \|\boldsymbol{x}_\tau^k - \boldsymbol{x}_\tau^{k-1}\| + \mathbb{I}_{\{2 \le k \le K-1\}} \frac{\tau_k}{\tau_k + \tau_{k-1}} \|\boldsymbol{x}_\tau^{k-1} - \boldsymbol{x}_\tau^{k-2}\| =: R_{\boldsymbol{x}_\tau}^k,$$

$$R_{\boldsymbol{p}_\tau^\star}^k \approx \tau_k \|D_1 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}_\tau\|_{I_k} + \frac{\tau_k}{\tau_{k-1} + \tau_k} \|\boldsymbol{p}_\tau^k - \boldsymbol{p}_\tau^{k-1}\| + \mathbb{I}_{\{1 \le k \le K-1\}} \frac{\tau_k}{\tau_k + \tau_{k+1}} \|\boldsymbol{p}_\tau^{k+1} - \boldsymbol{p}_\tau^k\| =: R_{\boldsymbol{p}_\tau}^k,$$

where the supremum norm (63) on I_k is approximated by uniformly sampling points in I_k and replacing the supremum with the maximum of the evaluated norms at those samples. The weights (67) and the residual $\rho_{\theta^\star, \theta_\tau^\star}^k$ require the triplet $(\boldsymbol{x}^\star, \theta^\star, \boldsymbol{p}^\star)$, which is not available and must be replaced for computing purposes by a suitable reconstruction obtained from $(\boldsymbol{x}_\tau, \theta_\tau, \boldsymbol{p}_\tau)$. We recall that $\boldsymbol{x}_\tau(t) = \boldsymbol{x}_\tau^{k-1}$ for $t \in [t_{k-1}, t_k)$ and $\boldsymbol{p}_\tau(t) = \boldsymbol{p}_\tau^k$ for $t \in (t_{k-1}, t_k]$. Then, we replace $\boldsymbol{x}^\star$ and $\boldsymbol{p}^\star$ on I_k by using the piecewise-linear reconstructions from $\boldsymbol{x}_\tau$ and $\boldsymbol{p}_\tau$, which are given by

$$\boldsymbol{x}^\star|_{I_k} \approx \boldsymbol{x}_\tau^{k-1} + \frac{t - t_{k-1}}{\tau_k} \left(\boldsymbol{x}_\tau^k - \boldsymbol{x}_\tau^{k-1} \right), \quad \boldsymbol{p}^\star|_{I_k} \approx \boldsymbol{p}_\tau^{k-1} + \frac{t - t_{k-1}}{\tau_k} \left(\boldsymbol{p}_\tau^k - \boldsymbol{p}_\tau^{k-1} \right).$$

The weights (67) are then approximated as follows:

$$\omega_{\boldsymbol{x}^\star}^k = \|\boldsymbol{x}^\star - \boldsymbol{x}_\tau\|_{I_k} \approx \sup_{t \in I_k} \frac{t - t_{k-1}}{\tau_k} \|\boldsymbol{x}_\tau^k - \boldsymbol{x}_\tau^{k-1}\| = \|\boldsymbol{x}_\tau^k - \boldsymbol{x}_\tau^{k-1}\| =: \omega_{\boldsymbol{x}_\tau}^k,$$

$$\omega_{\boldsymbol{p}^\star}^k = \|\boldsymbol{p}_\tau - \boldsymbol{p}\|_{I_k} \approx \sup_{t \in I_k} \left(1 - \frac{t - t_{k-1}}{\tau_k} \right) \|\boldsymbol{p}_\tau^k - \boldsymbol{p}_\tau^{k-1}\| = \|\boldsymbol{p}_\tau^k - \boldsymbol{p}_\tau^{k-1}\| =: \omega_{\boldsymbol{p}_\tau}^k.$$

We replace $\theta^\star$ with the globally continuous piecewise-quadratic reconstruction $\vartheta_\tau \in \mathcal{U}$, which is defined on each interval $[t_{k-1}, t_k]$ by

$$\vartheta_\tau|_{[t_{k-1}, t_k]}(t) = A_k s^2 + B_k s + C_k, \quad s = \frac{t - t_{k-1}}{\tau_k} \in [0, 1].$$

The coefficients $A_k, B_k, C_k \in \mathbb{R}^n$ are determined by the conditions $\vartheta_\tau(t_{k-1}) = \theta_\tau^{k-1}$, $\vartheta_\tau(t_k) = \theta_\tau^k$ and $\dot{\vartheta}_\tau(t_{k-1}) = S_{k-1}$, where S_{k-1} is the left-sided slope, which is given by $S_0 = 0$ and $S_k = \frac{\theta_\tau^k - \theta_\tau^{k-1}}{\tau_k}$. This gives explicitly $A_k = \theta_\tau^k - \theta_\tau^{k-1} - B_k$, $B_k = S_{k-1} \tau_k$, and $C_k = \theta_\tau^{k-1}$.

Fig. 1 shows piecewise linear reconstructions from the (piecewise constant) state approximation $\boldsymbol{x}_\tau$ and adjoint state approximation $\boldsymbol{p}_\tau$, as well as a piecewise quadratic reconstruction from the (piecewise linear) θ_τ, taken from our Swiss Roll numerical example (see the next section). We now discuss computing $\rho_{\theta^\star, \theta_\tau^\star}^k$ in (66), where $\theta^\star$ is replaced by ϑ_τ and $\theta_\tau^\star$ is replaced by θ_τ to obtain $\rho_{\vartheta_\tau, \theta_\tau}^k$, which is given by

$$\rho_{\vartheta_\tau, \theta_\tau}^k = \int_{I_k} \lambda \Big[(\theta_\tau, \vartheta_\tau - \theta_\tau) + (\dot{\theta}_\tau, \dot{\vartheta}_\tau - \dot{\theta}_\tau) \Big] + \left(D_2 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}_\tau, \vartheta_\tau - \theta_\tau \right) dt. \tag{69}$$

The integral containing θ_τ and ϑ_τ is computed as follows:

$$\int_{I_k} (\theta_\tau, \vartheta_\tau) dt = \int_0^1 (\theta_\tau^{k-1}(1-s) + \theta_\tau^k s, A_k s^2 + B_k s + C_k) \tau_k ds = \tau_k(\theta_\tau^{k-1}, \alpha^k) + \tau_k(\theta_\tau^k, \beta^k),$$

where the coefficients $\alpha^k \in \mathbb{R}^n$ and $\beta^k \in \mathbb{R}^n$ are given by

$$\alpha^k = -\frac{A_k}{4} + \frac{B_k - C_k}{3} + \frac{B_k - C_k}{2} + C_k, \quad \beta^k = \frac{A_k}{4} + \frac{B_k}{3} + \frac{C_k}{2}.$$
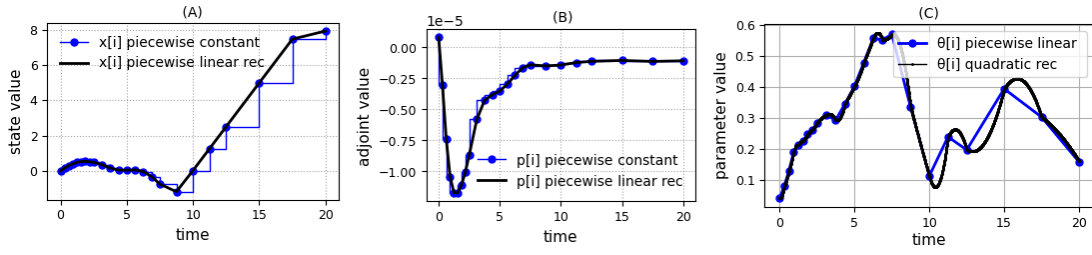
Figure 1: Discrete components of the state, control and the adjoint, and their respective reconstructions. (**A**): $x_{\tau,j}^i$ (j-th component of the i-th equation) of the discrete state $\boldsymbol{x}_\tau$ and its piecewise linear reconstruction. (**B**): component $p_{\tau,j}^i$ of the discrete adjoint $\boldsymbol{p}_\tau$ and its piecewise linear reconstruction. (**C**): component θ^i of the discrete control θ_τ and its piecewise quadratic reconstruction

For the derivatives $\dot{\theta}_\tau$ and $\dot{\vartheta}_\tau$, the respective integral is given by

$$\int_{I_k} \left(\dot{\theta}_\tau, \dot{\vartheta}_\tau\right) dt = \int_0^1 \left(\frac{\theta_\tau^k - \theta_\tau^{k-1}}{\tau_k}, \frac{2A_k s + B_k}{\tau_k}\right) \tau_k \ ds = \frac{1}{\tau_k}\left(\theta_\tau^k - \theta_\tau^{k-1}, \theta_\tau^k - \theta_\tau^{k-1}\right).$$

We also compute the integrals

$$\int_{I_k} (\theta_\tau, \theta_\tau) \, dt = \begin{pmatrix}\theta_\tau^{k-1}\\ \theta_\tau^k\end{pmatrix}^\top M_k^\tau \begin{pmatrix}\theta_\tau^{k-1}\\ \theta_\tau^k\end{pmatrix}, \quad M_k^\tau = \begin{pmatrix}\frac{\tau_k}{3} & \frac{\tau_k}{6}\\ \frac{\tau_k}{6} & \frac{\tau_k}{3}\end{pmatrix}, \quad \int_{I_k} \left(\dot{\theta}_\tau, \dot{\theta}_\tau\right) dt = \frac{1}{\tau_k}\|\theta_\tau^k - \theta_\tau^{k-1}\|^2.$$

where M_k^τ is the local mass matrix. The second term in (69) is evaluated similarly by using the structure of $D_2 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}_\tau$ and appropriate quadratures. Therefore, $\rho_{\vartheta_\tau, \theta_\tau}^k$ can now be evaluated. Finally, we define

$$\eta_k := R_{\boldsymbol{p}_\tau}^k \, \omega_{\boldsymbol{x}_\tau}^k + \rho_{\vartheta_\tau, \theta_\tau}^k + R_{\boldsymbol{x}_\tau}^k \, \omega_{\boldsymbol{p}_\tau}^k, \tag{70}$$

which provides a computable approximation of $\triangle_k$.

Algorithm 1 summarizes our adaptive approach. Starting from a coarse temporal mesh with initialized neural ODE parameters at each node, the algorithm iteratively refines the mesh while optimizing neural network parameters to minimize the objective. At each refinement step, a new node is inserted into the interval with the largest contribution to the indicator (68), and a new neural network layer is assigned to it. The indicator (70) is used to guide the refinement process. In the layer insertion procedure, new layers must be initialized to avoid disrupting prior training progress. For this, we set θ_τ^{new} as the average of $\theta_\tau^{k^*-1}$ and $\theta_\tau^{k^*}$, which performs well in our examples due to the regularization (14).

Algorithm 1 (Layerwise Adaptive Neural ODE Training)**.**
Require: *Coarse grid* $\{t_k\}_{k=0}^K$, *tolerance* $\text{tol} > 0$, *maximum iterations* $it_{\max}$, *refinement frequency* it_{up}
Ensure: *Adaptively refined grid* $\{t_k\}$, *trained parameters* $\{\theta_\tau^k\}$
1: *Initialize parameters* $\{\theta_\tau^k\}$ *at each node* t_k
2: $it \leftarrow 0$
3: **while** $it \leq it_{\max}$ **and** $J > \text{tol}$ **do**
4:     *Compute gradient* $\mathbf{g}_\tau$ *by solving* (55)
5:     *Update parameters* $\{\theta_\tau^k\}$ *using* $\mathbf{g}_\tau$
6:     **if** $it \bmod it_{\text{up}} = 0$ **then**
7:         *Compute indicators* η_k *for all intervals* I_k
8:         $k^* \leftarrow \arg\max_k \eta_k$
9:         $t_{\text{new}} \leftarrow (t_{k^*-1} + t_{k^*})/2$
10:        *Insert new layer with parameter* θ_τ^{new} *at* t_{new}
11:        *Update grid* $\{t_k\} \leftarrow \{t_0, \ldots, t_{k^*-1}, t_{\text{new}}, t_{k^*}, \ldots, t_K\}$
12:        $K \leftarrow K + 1$
13:     **end if**
14:     $it \leftarrow it + 1$
15: **end while**

For optimization, we use the Adam algorithm [25] in our numerical experiments. Since Adam accumulates first and second moments using exponential moving averages of past gradients, layer insertion requires resetting
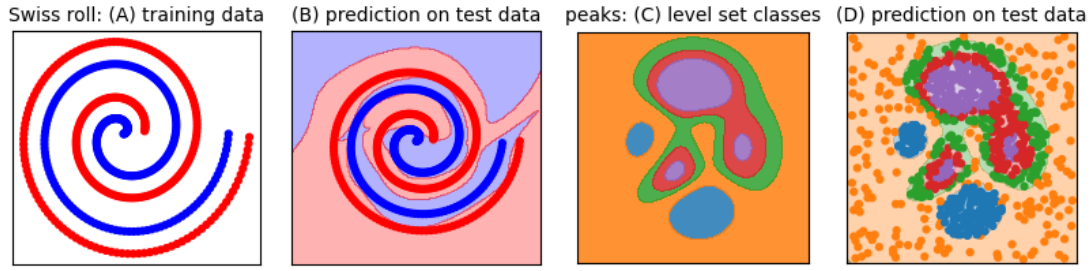
Figure 2: (**A**): Swiss roll training data with two colors indicating two different classes. (**B**): neural network prediction on the Swiss roll test data. (**C**): Peaks function level set classes. (**D**): neural network prediction on the Peaks test data.

these statistics, potentially reducing the benefit of momentum-based updates. Further exploring optimization strategies for adaptive training is left for future work. We only note here that second-order optimization or quasi-Newton methods as in [26] may offer an attractive alternative within our framework setting, especially given the small network sizes in early training stages; alternatively we also point to multilevel optimization techniques as in [27], [28].

8   Numerical examples

In this section, we present two numerical examples: one for binary classification and another one for multiclass classification; cf. also [4].

8.1   Binary classification

We generate a two-class dataset, known as the Swiss roll, consisting of two concentric spirals in $\mathbb{R}^2$:

$$f_{\text{blue}}(r, \varphi) = r(\cos(\varphi), \sin(\varphi))^\top, \quad f_{\text{red}}(r, \varphi) = (r + 0.2)(\cos(\varphi), \sin(\varphi))^\top,$$

for $r \in [0, 1]$ and $\varphi \in [0, 4\pi]$ at 513 points each. Every other point along these spiral curves is removed from the data set and used for the validation set, which is also used as the test set. The resulting training and test sets each contain $m = 513$ points, and the corresponding labels $y \in \{0, 1\}$, indicating the spirals our data points x^i are sampled from, see Fig. 2(A). For this binary classification task, the neural ODE yields a score (12), where $\hat{y}^i \in (0, 1)$ and $q_{\text{out}}(s) = (1 + e^{-s})^{-1}$. The empirical binary cross-entropy function

$$l(\boldsymbol{x}(T)) := \frac{1}{m} \sum_{i=1}^{m} \left[y^i \log \hat{y}^i + (1 - y^i) \log(1 - \hat{y}^i) \right]$$

is used to define the objective (30). For comparison, we consider three training approaches. First, we start with a one-layer network and iteratively refine it via Algorithm 1, producing an adaptive grid and a K-layer network. For the second approach, we insert layers at random intervals. Third, we train a K-layer residual network on a uniform time grid (non-adaptive), where K matches the depth from the adaptive approach. In all experiments, the network width is $d = 4$, training stops when the loss (30) falls below the tolerance tol $= 0.025$ or after $it_{\text{max}} = 3000$ iterations, the terminal time is $T = 20$, the activation function is $\sigma(x) = \tanh(x)$, the regularization is $\lambda = 10^{-3}$, the learning rate is $lr = 5 \cdot 10^{-3}$, and the layer insertion frequency for both adaptive and random insertion approaches is $it_{\text{up}} = 50$. We run 5 experiments with different random seeds and report the accuracy (fraction of correctly classified points on the validation set) and the number of iterations to reach the specified tolerance in Table 1. Compared to random layer insertion, adaptive learning via Algorithm 1 converges faster and produces smaller networks, while random insertion never reaches the tolerance, terminating at $it_{\text{max}} = 3000$ in all cases. Accuracy is also consistently higher with the adaptive approach. We observe that the fixed-grid training is highly sensitive to initialization, with the iteration count significantly varying across experiments, whereas adaptive training yields a more consistent number of iterations. Figure 2(B) shows a successful classification on the test set using one of our network realizations: one observes quite smooth decision boundary, which comes from the regularization (14). Fig. 3 exemplarily shows the produced adaptive

Table 1: Binary classification: comparison of adaptive, random layer insertion, and non-adaptive training for five neural network initializations. Accuracy and iteration count are reported for each experiment. The resulting neural network depths for the adaptive approach are $K = 52$ (Exp 1), $K = 40$ (Exp 2), $K = 51$ (Exp 3), $K = 39$ (Exp 4), and $K = 39$ (Exp 5).

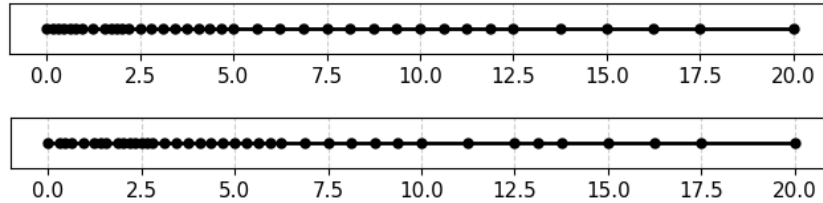| Method	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Adaptive	0.99 || 2447	0.99 || 1801	0.99 || 2397	0.99 || 1795	0.99 || 1794
Random	0.74 || 3000	0.94 || 3000	0.77 || 3000	0.91 || 3000	0.83 || 3000
Non-adaptive	0.98 || 317	0.99 || 1978	0.98 || 1254	0.99 || 1568	0.99 || 370

Figure 3: Binary classification: examples of time grids illustrating the new layer insertion positions in the interval $T = [0, 20]$ produced by the adaptive algorithm (Algorithm 1).

grids for two experiments (Exp 4 and Exp 2): the consistent trend is that the algorithm allocates more points at the beginning of the time interval and far fewer points near the end.

8.2 Multiclass classification

For the following experiment, we use the test problem for multiclass classification proposed in [4]. It is based on the Peaks function

$$f_{\mathrm{ps}}(x) = 3(1 - x_1)^2 \exp(-(x_1^2) - (x_2 + 1)^2)10(\frac{x_1}{5} - x_1^3 - x_2^5) \exp(-x_1^2 - x_2^2) - \frac{1}{3} \exp(-(x_1 + 1)^2 - x_2^2),$$

where $x \in [-3, 3]^2$. The function is discretized on a regular 256×256 grid, and the points are divided into 5 classes based on their function values. Specifically, we define $\mathcal{C}_i = \{x^i : c_{i-1} \leq f_{\mathrm{ps}}(x^i) < c_i\}$ for $i \in \{1, \dots, 4\}$, and $\mathcal{C}_5 = \{x^i : f_{\mathrm{ps}}(x^i) \geq c_4\}$. Here, $c_0 = \min f_{\mathrm{ps}}$ and $\{c_1, c_2, c_3, c_4\} = \{-2.2, 0.55, 1.75, 3.2\}$; see Fig. 2(C) for the resulting classes. We use a one-hot encoding $y^i \in \mathbb{R}^{d_{\mathrm{out}}}$ to mark the corresponding class of each point. For this multiclass classification task with $d_{\mathrm{out}} = 5$ classes, we use the empirical multiclass cross-entropy loss together with the softmax output map, defined by

$$l(\boldsymbol{x}(T)) := -\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{d_{\mathrm{out}}} y_j^i \log \hat{y}_j^i, \quad q_{\mathrm{out}}(s^j) = \frac{\exp(s^j)}{\sum_{k=1}^{d_{\mathrm{out}}} \exp(s^k)}, \quad j \in \{1, \dots, d_{\mathrm{out}}\}.$$

We randomly sample 1000 points from each class and use 80% of the samples for training, with the remaining points used for validation. Fig. 2(D) illustrates one of the successful classifications on the test set.

As in the previous example, we run 5 experiments with different neural network initializations for 3 approaches: adaptive layer insertion, random layer insertion, and training with a fixed architecture of K layers, where K is the network depth produced by the adaptive procedure. For all the experiments, the network width is $d = 20$, and training stops when the loss (30) drops below $\mathrm{tol} = 0.05$ or the iteration count exceeds $it_{\max} = 2500$. The terminal time is $T = 10$, the regularization parameter is $\lambda = 10^{-3}$, the activation function is $\tanh(x)$, the learning rate is $lr = 10^{-3}$, and the update frequency for adaptive training is $it_{\mathrm{up}} = 75$. The obtained accuracy and iteration count for each of these 5 experiments are displayed in Table 2. Similar to the binary classification problem, Table 2 shows faster convergence in all five experiments, compared to random layer insertion. The accuracy of both methods, however, is comparable, which can be explained by the substantially larger network width compared to the binary classification example. Nevertheless, it appears that the time grids produced by the adaptive approach yield better stability properties, which help reduce the iteration count. We also observe that the non-adaptive approach requires $40 - 50\%$ fewer iterations to reach the same tolerance

Table 2: Multiclass classification: comparison of adaptive, random layer insertion, and non-adaptive training for five neural network initializations. Accuracy and iteration count are reported for each experiment. The resulting neural network depths for the adaptive approach are $K = 24$ (Exp 1), $K = 24$ (Exp 2), $K = 27$ (Exp 3), $K = 26$ (Exp 4), and $K = 24$ (Exp 5).

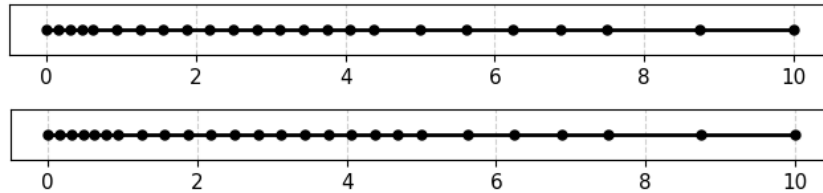| Method	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5
Adaptive	0.96 ‖ 1623	0.97 ‖ 1622	0.98 ‖ 1865	0.97 ‖ 1793	0.97 ‖ 1641
Random	0.98 ‖ 2094	0.98 ‖ 2500	0.98 ‖ 2500	0.96 ‖ 2500	0.97 ‖ 1698
Non-adaptive	0.98 ‖ 1000	0.98 ‖ 899	0.98 ‖ 1021	0.98 ‖ 775	0.98 ‖ 871

Figure 4: Multiclass classification: examples of time grids illustrating the new layer insertion positions in the interval $T = [0, 10]$ produced by the adaptive algorithm (Algorithm 1).

level as our adaptive approach, but this comes at the cost of training a large network from the start. In Fig. 4, we exemplarily show the adaptive grids produced by the algorithm for several experiments (Exp 1 and Exp 4) with the dataset. Similar to the previous binary classification example, we observe that the algorithm tends to allocate more points at the beginning of the time interval and fewer points near the end.

9   Conclusion and outlook

The identification of an "optimal" neural network topology is a very important, yet very challenging task in machine learning. In this work we have presented a time-continuous viewpoint on this problem, by introducing neural ODEs and employing an adaptive (time) meshing technique. Our dual-weighted goal-oriented approach aims at optimally resolving a pre-defined target quantity and it appears to work very well in practice when compared to more conventional approaches. This indeed encourages to develop and refine the method further.

To this end, one may identify several avenues of research: (i) In our approach, we inserted layers of a pre-defined "constant" size. This may be made more flexible by admitting variable layer-sizes. Borrowing once again terminology from adaptive finite element methods, the number of neurons per layer may be related to a spatial discretization of a (within a layer) continuous setting for x. One may then attempt to carry over space-time adaptation concepts from adaptive finite element discretizations. This may also include more complex target quantities in order to stabilize the adaptation process. (ii) The produced hierarchy of meshes induces a hierarchy of objective functions and constraint realizations associated with the continuous neural ODE constrained optimization problem. Upon the identification of suitable transfer operators within this hierarchy, one may develop (stochastic) multilevel minimization algorithms for further computational speed-up. (iii) We close by mentioning another possible direction which is connected to carrying our approach over to more complex network structures as, for instance, in convolutional neural networks (CNNs), where specific sub-blocks of a network are dedicated to specific tasks. Incorporating the latter in an error estimator / indicator based technique would be beneficial when trying to keep the overall number of networks parameters as small as possible. Clearly, one has to overcome challenges such as the non-locality of convolutional layers, pooling layers, local connectivity etc.

A   Discontinuous Galerkin formulation of neural ODE

We derive the variational formulation (48) which is suitable for our discontinuous Galerkin approach. Starting from the form (36), we split it over the time intervals as follows:

$$\mathcal{F}(\boldsymbol{x}, \theta; \varphi) := \sum_{k=1}^{K} \int_{I_k} \left(\dot{\boldsymbol{x}} - N_F(\boldsymbol{x}, \theta), \ \varphi_1 \right) dt + \left(\boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{in}}, \varphi_0 \right). \tag{71}$$

We integrate by parts:

$$\int_{I_k} \left(\dot{\boldsymbol{x}} - N_F(\boldsymbol{x}, \theta), \varphi_1 \right) dt = \int_{I_k} -\left(\boldsymbol{x}, \dot{\varphi}_1 \right) - \left(N_F(\boldsymbol{x}, \theta), \varphi_1 \right) dt + \left(\boldsymbol{x}(t_k^-), \varphi_1(t_k^-) \right) - \left(\boldsymbol{x}(t_{k-1}^+), \varphi_1(t_{k-1}^+) \right).$$

We perform a downwind approximation by replacing $\varphi_1(t_k^-)$ with $\varphi_1(t_k^+)$, and integrate by parts once more to obtain

$$\int_{I_k} \left(\dot{\boldsymbol{x}} - N_F(\boldsymbol{x}, \theta), \varphi_1 \right) dt + \left(\boldsymbol{x}(t_k^+), \varphi_1(t_k^-) \right) - \left(\boldsymbol{x}(t_k^-), \varphi_1(t_k^-) \right).$$

Recalling the definition of the jump, we define the following form:

$$\mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}, \theta; \varphi) := \sum_{k=1}^{K} \int_{I_k} \left(\dot{\boldsymbol{x}} - N_F(\boldsymbol{x}, \theta), \varphi_1 \right) dt + \left([\![\boldsymbol{x}]\!]^k, \varphi_1(t_k^-) \right) + \left(\boldsymbol{x}(0) - \boldsymbol{x}_{\mathrm{in}}, \varphi_0 \right),$$

which is the form required in (48).

The variational formulation of the adjoint equation which is suitable for our discontinuous Galerkin approach is derived as follows. First, we compute the derivative $D_1 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau, \theta_\tau; \boldsymbol{p}_\tau, \varphi_x)$ in the direction of $\varphi_x \in \mathcal{W}_\tau$, and obtain

$$\begin{aligned} D_1 \mathcal{F}_{\mathrm{DG}}(\boldsymbol{x}_\tau, \theta_\tau; \boldsymbol{p}_\tau, \varphi_x) = \sum_{k=1}^{K} \int_{I_k} \left(\boldsymbol{p}_\tau, \dot{\varphi}_x \right) - \left(D_1 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}_\tau, \varphi_x \right) dt \\ + \sum_{k=1}^{K} \left([\![\varphi_x]\!]^k, \boldsymbol{p}_\tau(t_k^-) \right) + \left(\varphi_x(0), \boldsymbol{p}_\tau(0) \right). \end{aligned} \tag{72}$$

Integrating the first term in (72) by parts, we get

$$\int_{I_k} \left(\boldsymbol{p}_\tau, \dot{\varphi}_x \right) dt = \left(\boldsymbol{p}_\tau(t_k^-), \varphi_x(t_k^-) \right) - \left(\boldsymbol{p}_\tau(t_{k-1}^+), \varphi_x(t_{k-1}^+) \right) - \int_{I_k} \left(\dot{\boldsymbol{p}}_\tau, \varphi_x \right) dt.$$

According to our jump notation, we have $[\![\varphi_x]\!]^k = \varphi_x(t_k^+) - \varphi_x(t_k^-)$. Therefore, we get

$$\int_{I_k} \left(\boldsymbol{p}_\tau, \dot{\varphi}_x \right) dt + \left([\![\varphi_x]\!]^k, \boldsymbol{p}_\tau(t_k^-) \right) = \int_{I_k} \left(-\dot{\boldsymbol{p}}_\tau, \varphi_x \right) dt + \left(\boldsymbol{p}_\tau(t_k^-), \varphi_x(t_k^+) \right) - \left(\boldsymbol{p}_\tau(t_{k-1}^+), \varphi_x(t_{k-1}^+) \right). \tag{73}$$

Observe further that

$$\sum_{k=1}^{K} \left(\boldsymbol{p}_\tau(t_k^-), \varphi_x(t_k^+) \right) = \sum_{k=1}^{K} \left(\boldsymbol{p}_\tau(t_{k-1}^-), \varphi_x(t_{k-1}^+) \right) - \left(\boldsymbol{p}_\tau(0), \varphi_x(0) \right) + \left(\boldsymbol{p}_\tau(T), \varphi_x(T) \right). \tag{74}$$

Using the identity (73) together with (72) and (74) yields the first equation in (52). Computing $D_2 \mathcal{J}(\boldsymbol{x}_\tau, \theta_\tau; \varphi_\theta)$ as in (41) gives the second equation in (52).

B   Time-marching interpretation of the DG(0) approximation scheme.

Here, we derive the forward Euler interpretation of the discrete state equation and the backward Euler interpretation of the discrete adjoint equation in (51). For $r = 0$ in $\mathcal{W}_\tau$, the ansatz $\boldsymbol{x}_\tau \in \mathcal{W}_\tau$ is given by

$$\boldsymbol{x}_\tau(t) = \sum_{k=1}^{K} \boldsymbol{x}_\tau^{k-1} \phi^{k-1}(t) + \boldsymbol{x}_\tau^K \chi^K(t),$$

where $\boldsymbol{x}_\tau^k \in \mathbb{R}^{md}$ for all $k = 0, ..., K$, and

$$\phi^{k-1}(t) = \begin{cases} 1, & t \in [t_{k-1}, t_k), \\ 0, & \text{elsewhere,} \end{cases} \quad , \quad \chi^K(t) = \begin{cases} 1, & t = t_K, \\ 0, & \text{elsewhere.} \end{cases}$$

To describe the test functions from $\mathcal{V}_\tau$ with $r = 0$, we use $\boldsymbol{v}_\tau^0 \chi^0(t)$, and $\boldsymbol{v}_\tau^k \xi^k(t)$ for $k \in \{1, .., K\}$, where $\boldsymbol{v}_\tau^k \in \mathbb{R}^{md}$ for each $k \in \{0, .., K\}$, and

$$\xi^k(t) = \begin{cases} 1, & t \in (t_{k-1}, t_k], \\ 0, & \text{elsewhere,} \end{cases} \quad , \quad \chi^0(t) = \begin{cases} 1, & t = 0, \\ 0, & \text{elsewhere.} \end{cases}$$

Since the trial functions are constant in time on I_k, we get $\dot{\boldsymbol{x}}_\tau = 0$, $\boldsymbol{x}_\tau(t_k^+) = \boldsymbol{x}_\tau^k$ and $\boldsymbol{x}_\tau(t_k^-) = \boldsymbol{x}_\tau^{k-1}$. Thus, from the discontinuous formulation (48), we get

$$\mathcal{F}_{\text{DG}}(\boldsymbol{x}_\tau, \theta_\tau; \boldsymbol{v}_\tau) = \sum_{k=1}^K \int_{I_k} -\big(N_F(\boldsymbol{x}_\tau, \theta_\tau), \boldsymbol{v}_\tau^k\big) \, dt + \big(\boldsymbol{x}_\tau^k - \boldsymbol{x}_\tau^{k-1}, \boldsymbol{v}_\tau^k\big) + \big(\boldsymbol{x}_\tau^0 - \boldsymbol{x}_{\text{in}}, \boldsymbol{v}_\tau^0\big), \tag{75}$$

which yields the following expression after rearrangement:

$$\boldsymbol{x}_\tau^k = \boldsymbol{x}_\tau^{k-1} + \int_{I_k} \big(N_F(\boldsymbol{x}_\tau, \theta_\tau), \boldsymbol{v}_\tau^k\big) \, dt, \quad \boldsymbol{x}_\tau^0 = \boldsymbol{x}_{\text{in}}. \tag{76}$$

Discretizing the right-hand side using the midpoint quadrature rule and with the standard canonical basis in $\mathbb{R}^{md}$, we obtain an explicit ResNet-type time-stepping scheme:

$$\boldsymbol{x}_\tau^0 = \boldsymbol{x}_{\text{in}}, \quad \boldsymbol{x}_\tau^k = \boldsymbol{x}_\tau^{k-1} + \tau_k F(\boldsymbol{x}_\tau^{k-1}, \theta_\tau(t_{k-1/2})), \tag{77}$$

where $k \in \{1, \cdots, K\}$ and $t_{k-1/2} := \frac{t_{k-1} + t_k}{2}$ is the midpoint of the interval I_k.

For the discretization of the adjoint equation (52), we use the ansatz $\boldsymbol{p}_\tau \in \mathcal{V}_\tau$, which is given by

$$\boldsymbol{p}_\tau(t) = \boldsymbol{p}_\tau^0 \chi^K(t) + \sum_{k=1}^K \boldsymbol{p}_\tau^k \xi^k(t),$$

and apply the test functions from $\mathcal{W}_\tau$. For the latter, we have $\boldsymbol{w}_\tau^{k-1} \phi^{k-1}(t)$ for $k \in \{1, \cdots, K\}$, and $\boldsymbol{w}_\tau^K \chi^K(t)$, where $\boldsymbol{w}_\tau^k \in \mathbb{R}^{md}$ for all $k \in \{0, \cdots, K\}$. For the trial functions from $\mathcal{V}_\tau$, we get $\dot{\boldsymbol{p}}_\tau = 0$, $\boldsymbol{p}_\tau(t_{k-1}^+) = \boldsymbol{p}_\tau^k$ and $\boldsymbol{p}_\tau(t_{k-1}^-) = \boldsymbol{p}_\tau^{k-1}$ in (52), which yields

$$\sum_{k=1}^K \int_{I_k} \big(-D_1 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}_\tau, \boldsymbol{w}_\tau^k\big) \, dt - \sum_{k=1}^K \big(\boldsymbol{p}_\tau^k - \boldsymbol{p}_\tau^{k-1}, \boldsymbol{w}_\tau^{k-1}\big) = 0$$
$$\big(\boldsymbol{p}_\tau(T), \boldsymbol{w}_\tau^K\big) = \big(l'(\boldsymbol{x}_\tau(T)), \boldsymbol{w}_\tau^K\big). \tag{78}$$

Rearranging, we get

$$\big(\boldsymbol{p}_\tau^{k-1}, \boldsymbol{w}_\tau^{k-1}\big) = \big(\boldsymbol{p}_\tau^k, \boldsymbol{w}_\tau^{k-1}\big) + \int_{I_k} \big(D_1 N_F(\boldsymbol{x}_\tau, \theta_\tau)^* \boldsymbol{p}_\tau, \boldsymbol{w}_\tau^k\big) \, dt,$$
$$\big(\boldsymbol{p}_\tau^K, \boldsymbol{w}_\tau^K\big) = \big(l'(\boldsymbol{x}_\tau^K), \boldsymbol{w}_\tau^K\big).$$

By testing with the standard canonical basis in $\mathbb{R}^{md}$, and approximating the integral with the midpoint quadrature rule, we get the time-marching scheme

$$\boldsymbol{p}_\tau^{k-1} = \boldsymbol{p}_\tau^k + \tau_k D_1 F\big(\boldsymbol{x}_\tau^{k-1}, \theta_\tau(t_{k-1/2})\big)^* \boldsymbol{p}_\tau^k, \quad k \in \{1, \cdots, K\},$$
$$\boldsymbol{p}_\tau^K = l'(\boldsymbol{x}_\tau^K). \tag{79}$$

The right-hand side, involving the product of the Jacobian and $\boldsymbol{p}_\tau^k$ in (79), is typically produced via backward-mode automatic differentiation without explicitly forming the Jacobian; see, e.g., [1].

References

[1] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations", *NeurIPS*, vol. 31, 2018.

[2] E. Dupont, A. Doucet, and Y. W. Teh, "Augmented neural odes", *NeurIPS*, vol. 32, 2019.

[3] C. Finlay, J.-H. Jacobsen, L. Nurbekyan, and A. Oberman, "How to train your neural ode: The world of jacobian and kinetic regularization", in *ICML*, 2020, pp. 3154–3164.

[4] E. Haber and L. Ruthotto, "Stable architectures for deep neural networks", *Inverse problems*, vol. 34, no. 1, p. 014 004, 2017.

[5] P. Kidger, "On neural differential equations", *arXiv preprint arXiv:2202.02435*, 2022.

[6] T. Zhang et al., "Anodev2: A coupled neural ode framework", *NeurIPS*, vol. 32, 2019.

[7] M. Benning, E. Celledoni, M. J. Ehrhardt, B. Owren, and C.-B. Schönlieb, "Deep learning as optimal control problems: Models and numerical methods", *Journal of Computational Dynamics*, vol. 6, no. 2, pp. 171–198, 2019.

[8] D. Ruiz-Balet and E. Zuazua, "Neural ode control for classification, approximation, and transport", *SIAM Review*, vol. 65, no. 3, pp. 735–773, 2023.

[9] R. Becker and R. Rannacher, "A general concept of adaptivity in finite element methods with applications to problems in fluid and structural mechanics", in *Grid Generation and Adaptive Algorithms*, Springer, 1999, pp. 51–75.

[10] M. Hintermüller and R. H. W. Hoppe, "Goal-oriented adaptivity in control constrained optimal control of partial differential equations", *SIAM J. Control Optim.*, vol. 47, no. 4, pp. 1721–1743, 2008, ISSN: 0363-0129,1095-7138. DOI: 10.1137/070683891. [Online]. Available: https://doi.org/10.1137/070683891.

[11] M. Hintermüller and R. H. W. Hoppe, "Goal-oriented adaptivity in pointwise state constrained optimal control of partial differential equations", *SIAM J. Control Optim.*, vol. 48, no. 8, pp. 5468–5487, 2010, ISSN: 0363-0129,1095-7138. DOI: 10.1137/090761823. [Online]. Available: https://doi.org/10.1137/090761823.

[12] M. Hintermüller and R. H. W. Hoppe, "Goal oriented mesh adaptivity for mixed control-state constrained elliptic optimal control problems", in *Applied and numerical partial differential equations*, ser. Comput. Methods Appl. Sci. Vol. 15, Springer, New York, 2010, pp. 97–111, ISBN: 978-90-481-3238-6. DOI: 10.1007/978-90-481-3239-3_8. [Online]. Available: https://doi.org/10.1007/978-90-481-3239-3_8.

[13] K. Kraft and S. Larsson, "The dual weighted residuals approach to optimal control of ordinary differential equations", *BIT Numerical Mathematics*, vol. 50, no. 3, pp. 587–607, 2010.

[14] E. Herberg, R. Herzog, F. Köhne, L. Kreis, and A. Schiela, "Sensitivity-based layer insertion for residual and feedforward neural networks", *arXiv preprint arXiv:2311.15995*, 2023.

[15] J. Aghili and O. Mula, "An optimal control framework for adaptive neural odes", *Advances in Computational Mathematics*, vol. 50, no. 3, p. 52, 2024.

[16] H. Amann and J. Escher, *Analysis. III*. Birkhäuser Verlag, Basel, 2009, pp. xii+468, Translated from the 2001 German original by Silvio Levy and Matthew Cargo, ISBN: 978-3-7643-7479-2; 3-7643-7479-2.

[17] D. O'Regan, *Existence theory for nonlinear ordinary differential equations*. Springer Science & Business Media, 1997, vol. 398.

[18] E. Emmrich, *Discrete versions of Gronwall's lemma and their application to the numerical analysis of parabolic problems*. Techn. Univ., 1999.

[19] R. A. Adams and J. J. F. Fournier, *Sobolev spaces* (Pure and Applied Mathematics (Amsterdam)), Second. Elsevier/Academic Press, Amsterdam, 2003, vol. 140, pp. xiv+305, ISBN: 0-12-044143-8.

[20] A. Pazy, *Semigroups of linear operators and applications to partial differential equations*. Springer Science & Business Media, 2012, vol. 44.

[21] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE constraints*. Springer Science & Business Media, 2008, vol. 23.

[22] J. Munoz-Matute, V. M. Calo, D. Pardo, E. Alberdi, and K. G. van der Zee, "Explicit-in-time goal-oriented adaptivity", *Computer Methods in Applied Mechanics and Engineering*, vol. 347, pp. 176–200, 2019.

[23] D. Braess, *Finite elements*, Third. Cambridge University Press, Cambridge, 2007, pp. xviii+365, Theory, fast solvers, and applications in elasticity theory, Translated from the German by Larry L. Schumaker, ISBN: 978-0-521-70518-9; 0-521-70518-5. DOI: 10 . 1017 / CBO9780511618635. [Online]. Available: https://doi.org/10.1017/CBO9780511618635.

[24] G. Dong, M. Hintermüller, and K. Papafitsoros, "A descent algorithm for the optimal control of relu neural network informed pdes based on approximate directional derivatives", *SIAM Journal on Optimization*, vol. 34, no. 3, pp. 2314–2349, 2024.

[25] D. P. Kingma, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

[26] E. Newman, L. Ruthotto, J. Hart, and B. van Bloemen Waanders, "Train like a (var) pro: Efficient training of neural networks with variable projection", *SIAM Journal on Mathematics of Data Science*, vol. 3, no. 4, pp. 1041–1066, 2021.

[27] R. Baraldi, M. Hintermüller, and Q. Wang, *A multilevel proximal trust-region method for nonsmooth optimization with applications*, 2025. arXiv: 2512 . 00538 [math.OC]. [Online]. Available: https ://arxiv.org/abs/2512.00538.

[28] H. Calandra, S. Gratton, E. Riccietti, and X. Vasseur, "On a multilevel Levenberg-Marquardt method for the training of artificial neural networks and its application to the solution of partial differential equations", *Optim. Methods Softw.*, vol. 37, no. 1, pp. 361–386, 2022, ISSN: 1055-6788,1029-4937. DOI: 10.1080/10556788.2020.1775828. [Online]. Available: https://doi.org/10.1080/10556788. 2020.1775828.