SEPARABLE APPROXIMATIONS OF OPTIMAL VALUE FUNCTIONS AND THEIR REPRESENTATION BY NEURAL NETWORKS. *

MARIO SPERL†, LUCA SALUZZI ‡, DANTE KALISE §, AND LARS GRÜNE†

Abstract. The use of separable approximations is proposed to mitigate the curse of dimensionality related to the approximation of high-dimensional value functions in optimal control. The separable approximation exploits intrinsic decaying sensitivity properties of the system, where the influence of a state variable on another diminishes as their spatial, temporal, or graph-based distance grows. This property allows the efficient representation of global functions as a sum of localized contributions. A theoretical framework for constructing separable approximations in the context of optimal control is proposed by leveraging decaying sensitivity in both discrete and continuous time. Results extend prior work on decay properties of solutions to Lyapunov and Riccati equations, offering new insights into polynomial and exponential decay regimes. Connections to neural networks are explored, demonstrating how separable structures enable scalable representations of high-dimensional value functions while preserving computational efficiency.

Key words. Separable approximations, Decaying sensitivity, Neural Networks, Optimal control

MSC codes. 49L20, 68T07, 93C41

1. Introduction.

High-dimensional systems are ubiquitous in science and engineering, with applications ranging from physics and biology, to social networks and machine learning. A recurring challenge in these domains is the efficient representation and computation of functions defined over high-dimensional spaces. It is often the case that the computational complexity associated to function representation grows exponentially with the space dimension, a phenomenon referred to as the *curse of dimensionality*.

One promising approach to mitigating the curse of dimensionality lies in exploiting separability, that is, the ability to approximate a global function as a sum of local contributions involving only a subset of variables. Separability properties can be intrinsic to the problem under study, as in systems where interactions decay with distance or graph connectivity, such as in physical systems governed by locality principles or networked systems with sparse interactions. At a computational level, separability can be enforced by a suitable approximation ansatz, including separation of variables, ANOVA [26], and most recent tools such as tensor decompositions [6, 25]. Theoretical and computational frameworks for separable approximations offer significant reductions in complexity while retaining essential structural features of the underlying functions.

In this work, we focus on the construction and analysis of separable approximations for functions that exhibit a decaying sensitivity property, where the influence of one variable on another diminishes as their distance (spatial, temporal, or graph-based) increases.

*Submitted to the editors February 11, 2025.

Funding: This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - project number 463912816.

†Mathematical Institute, University of Bayreuth, Germany, (mario.sperl@uni-bayreuth.de, and lars.gruene@uni-bayreuth.de).

‡Department of Mathematics and Computer Science, University of Ferrara, Italy, (luca.saluzzi@unife.it)

§Department of Mathematics, Imperial College London, United Kingdom, (d.kalise-balza@imperial.ac.uk)

The concept of decaying sensitivity is intrinsically linked to the off-diagonal decay property, a topic extensively explored in the field of numerical linear algebra. A foundational theoretical framework for understanding this behavior is introduced in [4], where decay bounds are derived for the entries of general matrix functions. This analysis is further advanced in [15], which examines decay properties in the context of optimal control, specifically for solutions to Lyapunov equations. Additionally, related investigations are presented in [22], where the sparsity structure of solutions to Riccati equations is analyzed, providing complementary insights into the decay and sparsity phenomena in matrix equations. Our particular focus of application are optimal control problems, for which such decaying sensitivity properties in time and space have been intensively studied in the past years, see, e.g., [13, 24, 31] for decaying sensitivity in time and [14, 27, 30, 34] for decaying sensitivity in space for problems governed by discrete-time dynamics, ODEs and PDEs. See also [29] for a related property for static nonlinear optimization problems. The key prerequisite for the occurrence of decaying sensitivity in space is that the optimal control problem consists of subsystems that are interconnected by a graph structure. For instance in multi-agent control systems or robotic swarms, each agent's state and control variables are influenced predominantly by neighboring agents, leading to a natural decaying sensitivity structure. In [9], it was shown that low-rank approximations to linear-quadratic value functions explicitly depend on the connectivity of the dynamics. By leveraging this decaying sensitivity property, we develop computationally efficient methods for approximating optimal value functions through separable structures.

1.1. Contribution. The main purpose of this paper is to show that decaying sensitivity in space enables the construction of accurate separable approximations. While our result is formulated for arbitrary functions and thus provides a general framework for separable approximations, we focus on the approximation of value functions arising in the context of dynamic programming or reinforcement learning algorithms. Preliminary versions of the results in this paper appeared in the conference paper [33], however, the estimates provided in this paper significantly improve upon the results in [33]. In particular, we also address the case in which the sensitivity does not decay exponentially but only polynomially. Moreover, we comment on the relation of our result to known approximation-theoretic methods, such as *anchored decompositions*.

Furthermore, we establish connections between the existence of separable approximations and neural networks, illustrating how structured architectures can benefit from separability to achieve scalable representations of high-dimensional functions. Through theoretical results and numerical experiments, we demonstrate the potential of this approach in reducing computational complexity while maintaining high approximation accuracy.

1.2. Outline. After defining the setting of this paper in Section 2, our main results are formulated and proved in Section 3. Here, we also explain the relation of our construction to the *anchored decomposition*. In Section 4 we recall results on the existence of exponentially decaying sensitivity for discrete-time linear quadratic optimal control problems and provide a decaying sensitivity result for a class of continuous-time linear-quadratic optimal control problems. In Section 5 we explain why the existence of separable approximations reduces the complexity of representing high-dimensional functions with neural networks and in Section 6 we provide numerical examples that illustrate these findings.

1.3. Notation. Throughout this paper, we adopt the following notation. For a vector $x \in \mathbb{R}^n$, we denote its one-dimensional components as x_j for $1 \leq j \leq n$. The entries of a given matrix A, which may include block-entries, are denoted by $A[i,j]$. The identity matrix in $\mathbb{R}^{n\times n}$ is denoted by I_n, and the null matrix of the same dimension is represented by 0_n. Additionally, unless otherwise specified, the notation $\|\cdot\|$ refers to the Euclidean 2-norm. The comparison space $\mathcal{L}$ consists of all continuous and strictly decreasing functions $\gamma\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that satisfy the condition $\lim_{\tau\to\infty}\gamma(\tau) = 0$. For a set $\Omega \subset \mathbb{R}^n$ and a function $f\colon \Omega \to \mathbb{R}^m$, we set $\|f\|_{\infty,\Omega} \coloneqq \sup_{x\in\Omega}\|f(x)\|$. If f is continuously differentiable, we further define

$$\|f\|_{\mathcal{C}^1(\Omega)} \coloneqq \|f\|_{\infty,\Omega} + \|\nabla f\|_{\infty,\Omega}.$$

2. General setting for decaying sensitivity. In this section, we introduce sensitivity for graph-based problems through a general framework. Though our primary results in Section 4 address decaying sensitivity in optimal control problems, the underlying concepts have broader applicability. This general formulation is valuable because the existence of separable approximations – a key finding related to decaying sensitivity – extends beyond optimal control to other graph-based problems.

Setting 2.1. Let us consider a system of $s \in \mathbb{N}$ agents with states $z_1, \ldots, z_s$ with $z_j \in \mathbb{R}^{n_j}$ for $j = 1, \ldots, s$, which together constitute a vector in the state space $\mathbb{R}^n$, where $n = \sum_{j=1}^{s} n_j$. Denote by $x = (z_1, \ldots, z_s) \in \mathbb{R}^n$ the combined state of all agents. The interactions among the agents are represented by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of s vertices $\mathcal{V} = \{1, \ldots, s\}$, corresponding to the agents, with an edge $(i,j) \in \mathcal{E}$ indicating that agent z_i exerts some influence on z_j. For $i, j \in \{1, \ldots, s\}$, define $\mathrm{dist}_{\mathcal{G}}(i,j)$ to be the length of the shortest path from node i to node j in $\mathcal{G}$ (possibly ∞ if no path exists and 0 if $i = j$). We are interested in computing a quantity of interest given by a function $V\colon \Omega \subset \mathbb{R}^n \to \mathbb{R}$.

Note that we use a directed graph $\mathcal{G}$ to describe the interaction among agents. However, in the special case of symmetric influence, these interactions can be represented using an undirected graph, which can be viewed as a particular case of a directed graph. While the discussion in Section 4 and the presented examples focus on the symmetric case, leading to undirected graphs, we adopt the more general formulation with directed graphs, as the main results in Section 3 hold for both cases. We emphasize that the edges in our graph are defined as tuples of nodes without assigned weights and the length of a path is determined by the number of its edges. An interesting extension would be to consider weighted graphs, where the weight of an edge represents the strength of the coupling between the nodes and the graph distance between two nodes is defined as the sum of the weights along a path. A detailed analysis of this setting is, however, deferred to future research.

In the following, we formalize the concept of a decaying sensitivity between agents with an increasing graph distance. To this end, for each $j \in \{1, \ldots, s\}$ define the mapping

$$g_j\colon \mathbb{R}^n \to \mathbb{R}, \qquad x = (z_1, \ldots, z_s) \mapsto V(x) - V(z_1, \ldots, z_{j-1}, 0, z_{j+1}, \ldots, z_s). \tag{2.1}$$

Each such mapping g_j allows us to quantify the influence that the value of z_j has on the evaluation of V at some point x in the state space compared to the anchor 0. While the mappings g_j above could be defined using a general anchor in place of 0 in the second term of the difference, we choose to compare with 0 to avoid notational overload. We are now interested in how much the value of such a function g_j at some point in the

state space is affected by a change of the values z_i with $i \in \{1, \dots, s\}, i \neq j$. We characterize this effect by examining the Lipschitz constant of the mapping g_j with respect to z_i. We call g_j Lipschitz with respect to z_i at a point $x = (z_1, \dots, z_s) \in \Omega$ with Lipschitz constant $L > 0$ if for all $\tilde{x} = (z_1, \dots, z_{i-1}, \tilde{z}_i, z_{i+1}, \dots, z_s) \in \Omega$ it holds that

$$|g_j(x) - g_j(\tilde{x})| \leq L \|z_i - \tilde{z}_i\|.$$

DEFINITION 2.2. *We say that V has the γ-decaying sensitivity property with respect to $\mathcal{G}$ on Ω for $\gamma \in \mathcal{L}$ if for all $i, j \in \{1, \dots, s\}$ with $i \neq j$ and for all $x \in \Omega$ the mapping g_j as defined in* (2.1) *is Lipschitz w.r.t. z_i at the point $x = (z_1, \dots, z_s)$ with Lipschitz constant*

$$L_{i,j} \coloneqq \gamma(\mathrm{dist}_{\mathcal{G}}(i, j)) \, \|z_j\| \,. \tag{2.2}$$

If the considered graph $\mathcal{G}$ and the set Ω are clear from the context, we refer to this property as γ-decaying sensitivity. Intuitively, Definition 2.2 implies that the influence an agent z_i has on z_j, as measured by the mapping g_j, decreases as the graph distance between i and j increases. Note that in the situation of Definition 2.2 we may have $\mathrm{dist}_{\mathcal{G}}(i, j) = \infty$ for some $i, j \in \{1, \dots, s\}$ if there is no path from node i to node j in $\mathcal{G}$. In this case, (2.2) is to be understood as $L_{i,j} = 0$, thus stating that g_j is independent of z_i.

3. Separable approximation through decaying sensitivity. In this section, we first clarify the relationship between the decaying sensitivity property introduced in Definition 2.2 and the Hessian of the function V, under the assumption that V is twice continuously differentiable. We then revisit the approximation method for V proposed in [33] and apply Definition 2.2 to establish an enhanced error bound that is independent of the number of agents s for settings exhibiting either exponential or polynomial decay in sensitivity. Finally, we highlight a connection between our construction and the anchored decomposition, a well-established concept in approximation theory. Throughout this section, we work within the framework established in Setting 2.1.

3.1. Relation between decaying sensitivity and the Hessian. Assuming twice differentiability of the function V, we can deduce γ-decaying sensitivity from a corresponding property of its Hessian, as established in the following result.

Proposition 3.1. Let $\Omega \subset \mathbb{R}^n$ be convex and assume that V is $\mathcal{C}^2(\Omega)$. If there exist $\gamma \in \mathcal{L}$ such that for all $i, j \in \{1, \dots, s\}$ with $i \neq j$ and for all $x = (z_1, \dots, z_s) \in \Omega$ we have

$$\left\| \frac{\partial^2}{\partial z_i \partial z_j} V(x) \right\| \leq \gamma(\mathrm{dist}_{\mathcal{G}}(i, j)),$$

then V has γ-decaying sensitivity on Ω.

Proof. Fix some $i, j \in \{1, \dots, s\}$, $i \neq j$. Let $z_{-j} = (z_1, \dots, z_{j-1}, z_{j+1}, \dots, z_s) \in \mathbb{R}^{n-n_j}$. By assumption, the mapping

$$z_j \mapsto \frac{\partial}{\partial z_i} V(z_1, \dots, z_s)$$

is continuously differentiable. Thus, for all $z_j \in \mathbb{R}^{n_j}$ with $(z_1, \dots, z_s) \in \Omega$ there exists some $\xi \in [0,1]$ such that

$$\left\| \frac{\partial}{\partial z_i} g_j(z_1, \dots, z_s) \right\| = \left\| \frac{\partial}{\partial z_i} V(x) - \frac{\partial}{\partial z_i} V(z_1, \dots, z_{j-1}, 0, z_{j+1}, \dots, z_s) \right\|$$
$$= \left\| \frac{\partial^2}{\partial z_i \partial z_j} V(z_1, \dots, z_{j-1}, \xi z_j, z_{j+1}, \dots, z_s) z_j \right\|.$$

By assumption and the convexity of Ω we obtain

$$\left\| \frac{\partial}{\partial z_i} g_j(z_1, \dots, z_s) \right\| \leq \gamma(\mathrm{dist}_{\mathcal{G}}(i,j)) \left\| z_j \right\|.$$

Thus, $L_{i,j} \coloneqq \gamma(\mathrm{dist}_{\mathcal{G}}(i,j)) \left\| z_j \right\|$ is a Lipschitz constant of g_j with respect to z_i at the point $x = (z_1, \dots, z_s)$. Then, applying Definition 2.2, the result follows. ∎

Quadratic functions are of particular interest in optimal control, as they represent solutions to the Linear Quadratic Regulator (LQR) problem, which will be examined in greater detail in Section 4. For quadratic functions of the form $V(x) = x^T P x$ for some matrix $P \in \mathbb{R}^{n \times n}$ we can characterize the decaying sensitivity of V via a decay in the block structure of P. Given an exponential decay, this links Definition 2.2 to the concept of spatially exponential decaying matrices. In the following, for $P \in \mathbb{R}^{n \times n}$ we denote with $P[i,j]$ the block-matrix of P consisting of the columns corresponding to z_j and the rows corresponding to z_i.

Definition 3.2 (Definition 1 in [34]). *A matrix $P \in \mathbb{R}^{n \times n}$ is called (C, ρ) spatially exponential decaying (SED) for $C > 0$ and $\rho \in (0,1)$ if for all $i, j \in \{1, \dots, s\}$*

$$\|P[i,j]\| \leq C \rho^{\mathrm{dist}_{\mathcal{G}}(i,j)}.$$

Corollary 3.3. *Let $V(x) = x^T P x$ with $P \in \mathbb{R}^{n \times n}$. Assume that there exists $\gamma \in \mathcal{L}$ such that for all $i, j \in \{1, \dots, s\}$ it holds that*

$$\|P[i,j]\| \leq \gamma(\mathrm{dist}_{\mathcal{G}}(i,j)). \tag{3.1}$$

Then V admits a γ-decaying sensitivity property. In particular, if P is (C, ρ) SED for some $C > 0$ and $\rho \in (0,1)$, then V admits an exponential decaying sensitivity given by $\gamma(\cdot) = C\rho^{(\cdot)}$.

Proof. Since P is the Hessian of V, the claim follows directly from Proposition 3.1. ∎

3.2. Construction of a Separable Approximation. In the following, we examine the primary advantage of the decaying sensitivity property, which is that the global behavior of V can be approximated by considering its behavior on local subsets of the graph, leading to a separable approximation of the function V.

Definition 3.4. *Let $n, d \in \mathbb{N}$ with $d \leq n$ and $\Omega \subset \mathbb{R}^n$. A function $F \colon \mathbb{R}^n \to \mathbb{R}$ is called d-separable if there exist $k \in \mathbb{N}$ and functions $F_j \colon \mathbb{R}^{d_j} \to \mathbb{R}$ with $d_j \leq d$ for all $1 \leq j \leq k$ such that for all $x \in \Omega$*

$$F(x) = \sum_{j=1}^{k} F_j(y_j),$$

where for all $j \in \{1, \dots, k\}$ it holds that $y_j = (x_{\kappa_{j,1}}, \dots, x_{\kappa_{j,d_j}})$ for some $\kappa_{j,i} \in \{1, \dots, n\}$, $1 \leq i \leq d_j$.

We consider the following construction for approximating V by a separable function, first presented in [33]. Although the construction therein was developed in the context of an optimal value function V, it directly extends to Setting 2.1, that is, to a general function V defined on substates with an associated graph $\mathcal{G}$. The approximation depends on a parameter l, which determines the size of local neighborhoods in the graph $\mathcal{G}$ used for the separable representation. For any fixed $l \in \mathbb{N}$ it can be expressed as follows:

$$V(x) \approx V(0) + \Psi_l(x) = V(0) + \sum_{j=1}^{s} \Psi_l^j(z_{j,l}), \quad x \in \Omega, \tag{3.2}$$

where $z_{j,l}$ represents the vector containing the states of subsystem z_j along with all subsystems z_i with $i \geq j$ and $\mathrm{dist}_{\mathcal{G}}(i,j) \leq l$. Let $d_{j,l} \in \mathbb{N}$ denote the dimension associated with $z_{j,l}$, i.e., $d_{j,l}$ is the sum of the dimensions of the subsystems that have an index larger than j and a graph distance of at most l from z_j. Furthermore, let $x_{j,l} \in \mathbb{R}^n$ denote the lifting of $z_{j,l}$ into $\mathbb{R}^n$ by inserting the value 0 for all subsystems not included in $z_{j,l}$. The component functions Ψ_l^j in (3.2) are defined as

$$\Psi_l^j(z_{j,l}) \coloneqq V(x_{j,l}) - V(\Lambda_j x_{j,l}), \tag{3.3}$$

where $\Lambda_j \colon \mathbb{R}^n \to \mathbb{R}^n$ sets the values of subsystem j to 0, i.e., for $x = (z_1, \ldots, z_s)$ it is

$$\Lambda_j x = (z_1, \ldots, z_{j-1}, 0, z_{j+1}, \ldots, z_s).$$

Figure 1 illustrates a graph composed of five subsystems, where the graph distance is given by $\mathrm{dist}_{\mathcal{G}}(i,j) = |i-j|$. According to our notation, in this example we have $z_{1,2} = (z_1, z_2, z_3)$, $x_{1,2} = (z_1, z_2, z_3, 0, 0)$, and $\Psi_2^1(z_{1,2}) = V(z_1, z_2, z_3, 0, 0) - V(0, z_2, z_3, 0, 0)$. Note that (3.2) is based on a successive reduction of substates through the functions Ψ_l^j; see Figure 2 for an illustration. Specifically, for $x = (z_1, \ldots, z_s) \in \Omega$, we begin with the approximation

$$V(x) \approx \Psi_l^1(z_{1,l}) + V(0, z_2, \ldots, z_s).$$

In the next step, we approximately have

$$V(0, z_2, \ldots, z_s) \approx \Psi_l^2(z_{2,l}) + V(0, 0, z_3, \ldots, z_s).$$

This procedure is continued iteratively until all substates are taken into account. The remaining term corresponds to the value at the origin, which then yields (3.2). Note that in general the resulting approximation will depend on the numbering of the subsystems z_j.

Remark 3.5 (Role of the parameter l). The expression in (3.2) constitutes a d-separable approximation of V with $d \coloneqq \max_{j \in \{1,\ldots,s\}} d_{j,l}$. The parameter l plays a crucial role in this construction. On one hand, it determines the dimension of the subspaces associated with the vectors $z_{j,l}$: smaller values of l reduce the number of substates included in $z_{j,l}$, thereby decreasing $d_{j,l}$ and consequently lowering the order of separability. This reduction is advantageous, as it results in lower-dimensional functions in the separable approximation of V. On the other hand, larger values of l account for longer graph distances, which generally lead to better approximation quality, see Lemma 3.6 in the following. In conclusion, l provides a trade-off between

sparsity and approximation quality. Moreover, note that while the presented construction is valid for all $l \in \mathbb{N}$, choosing $l \geq s$ yields the same graph neighborhoods as $l = s - 1$, since the graph distance between any two nodes in a graph with s nodes is at most $s - 1$.

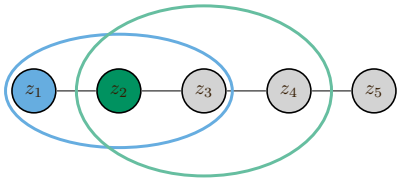

Figure 1: Graph neighborhoods $z_{1,2}$ (left ellipse) and $z_{2,2}$ (right ellipse) in a sequential graph.

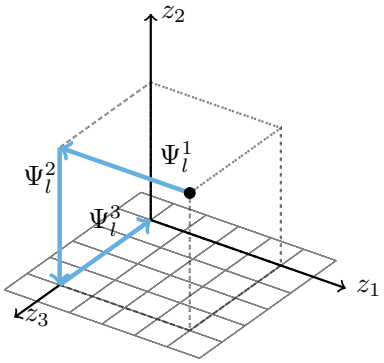

Figure 2: Stepwise isolation of the substates z_j through the functions Ψ_l^j illustrated for $s = 3$.

We now establish an error bound for the approximation in (3.2) based on the decaying sensitivity property introduced in Definition 2.2. In contrast to Theorem 4 in [33], this error bound is designed to be independent of both the number of agents s and the dimension n. However, as a trade-off, the result is restricted to $\mathcal{L}_2$ balls, instead of $\mathcal{L}_\infty$ balls, as presented in [33, Theorem 4]. This modification is achieved through the revised concept of decaying sensitivity outlined in Definition 2.2, as well as by leveraging the interplay between the type of decay and the growth of the graph as the dimension increases.

LEMMA 3.6. *Assume that V has the γ-decaying sensitivity property. Then, for any $l \in \mathbb{N}$ and $x \in \Omega$, we have*

$$|V(x) - \Psi_l(x) - V(0)| \leq \|x\|_2^2 \|D_l\|_2,$$

where $D_l \in \mathbb{R}^{s \times s}$ is given by

$$D_l[i,j] := \begin{cases} \gamma(\mathrm{dist}_{\mathcal{G}}(i,j)), & \text{if } \mathrm{dist}_{\mathcal{G}}(i,j) > l, \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

Proof. Initially, we observe that for any $i, j \in \{1, \ldots, s\}$ we have that either $\mathrm{dist}_{\mathcal{G}}(i,j) \leq s-1$ or $\mathrm{dist}_{\mathcal{G}}(i,j) = \infty$. In the latter case it follows that $V(x) - V(\Lambda_j x)$ is independent of z_i. Thus, by considering graph-neighborhoods of the size $s-1$ we can express $V(x)$ as $V(x) = V(0) + \sum_{j=1}^{s} \Psi_{s-1}^j(z_{j,s-1})$. This shows the claim for

$l \geq s-1$. For the case $l \leq s-2$ we use a telescoping sum to write

$$\begin{aligned} &\left|V(x) - \sum_{j=1}^{s} \Psi_l^j(z_{j,l}) - V(0)\right| = \left|\sum_{j=1}^{s} \Psi_{s-1}^j(z_{j,s-1}) - \sum_{j=1}^{s} \Psi_l^j(z_{j,l})\right| \\ &\leq \sum_{j=1}^{s} \sum_{k=l}^{s-2} \left|\Psi_{k+1}^j(z_{j,k+1}) - \Psi_k^j(z_{j,k})\right|. \end{aligned} \tag{3.5}$$

Now, fix some $j \in \{1,\dots,s\}$ and $\tilde{l} \in \{l,\dots,s-2\}$. Let $\tau \in \mathbb{N}$ and $i_1,\dots,i_\tau \in \{1,\dots,s\}$ such that $\{i \in \{1,\dots,s\} \mid \operatorname{dist}_{\mathcal{G}}(j,i) = \tilde{l}\} = \{i_1,\dots,i_\tau\}$. Then, by the definitions of Ψ_l^j in (3.3) and g_j in (2.1) we have

$$\begin{aligned} &\Psi_{\tilde{l}}^j(z_{j,\tilde{l}}) - \Psi_{\tilde{l}+1}^j(z_{j,\tilde{l}+1}) = V(x_{j,\tilde{l}}) - V(\Lambda_j x_{j,\tilde{l}}) - V(x_{j,\tilde{l}+1}) + V(\Lambda_j x_{j,\tilde{l}+1}) \\ =&\sum_{p=1}^{\tau} g_j(x_{j,\tilde{l}+1}^{(p)}) - g_j(x_{j,\tilde{l}}^{(p)}), \end{aligned}$$

where for each $1 \leq p \leq \tau$ the vectors $x_{j,\tilde{l}+1}^{(p)}$ and $x_{j,\tilde{l}}^{(p)}$ are obtained by setting all entries corresponding to $z_{i_1},\dots,z_{i_p}$ to 0 in the vectors $x_{j,\tilde{l}+1}$ and $x_{j,\tilde{l}}$, respectively. By successively applying the γ-decaying sensitivity property as defined in Definition 2.2, we obtain

$$\left|\Psi_{\tilde{l}+1}^j(z_{j,\tilde{l}+1}) - \Psi_{\tilde{l}}^j(z_{j,\tilde{l}})\right| \leq \gamma(\tilde{l}+1) \sum_{p=1}^{\tau} \left\|z_{i_p}\right\| \|z_j\|.$$

From this, using (3.5) we can derive

$$\begin{aligned} &\left|V(x) - \sum_{j=1}^{s} \Psi_l^j(z_{j,l}) - V(0)\right| \leq \sum_{j=1}^{s} \sum_{k=l}^{s-2} \sum_{i:\operatorname{dist}_{\mathcal{G}}(i,j)=k+1} \|z_i\| \, \gamma(k+1) \, \|z_j\| \\ =&\sum_{j=1}^{s} \sum_{i:\operatorname{dist}_{\mathcal{G}}(i,j)>l} \|z_i\| \, \gamma(\operatorname{dist}_{\mathcal{G}}(i,j)) \, \|z_j\| = \sum_{j=1}^{s} \|z_j\| \sum_{i=1}^{s} D_l[i,j] \, \|z_i\| \\ =&\begin{bmatrix} \|z_1\| \\ \vdots \\ \|z_s\| \end{bmatrix}^T D_l \begin{bmatrix} \|z_1\| \\ \vdots \\ \|z_s\| \end{bmatrix} \leq \left\| \begin{bmatrix} \|z_1\| \\ \vdots \\ \|z_s\| \end{bmatrix} \right\|^2 \|D_l\|_2 = \|x\|_2^2 \|D_l\|_2, \end{aligned}$$

which shows the claim. ∎

In particular, Lemma 3.6 shows that increasing the graph distance l improves the approximation quality. Specifically, for $l \geq s-1$, we have $D_l = 0$, which implies $\Psi_l + V(0) = V$. However, in this case, the local state $z_{1,l}$ already contains all substates connected to z_1, resulting in a high order of separability. In view of our objective to approximate V using low-dimensional functions, it is therefore not desirable to choose $l \geq s-1$. Instead, l should be selected as a trade-off between approximation accuracy and numerical efficiency, cf. Remark 3.5.

In the following, we specify the error bound derived from Lemma 3.6 for both exponential and polynomial decay scenarios. For this, it is essential to quantify the growth of neighborhoods within the underlying graph. We define a function $r\colon \mathbb{N} \to \mathbb{N}$ as a growth bound for the graph $\mathcal{G}$ if it satisfies the condition:

$$\forall l \in \{1,\dots,s\}, j \in \{1,\dots,s\} : \ |\{i \in \{1,\dots,s\} \mid \operatorname{dist}_{\mathcal{G}}(j,i) = l\}| \leq r(l). \tag{3.6}$$

Note that if r is a subexponential function, condition (3.6) coincides with Assumption 3.4 in [30], which is used in Corollary 3.5 therein to show the exponential accuracy of a truncated feedback controller for time-discrete linear quadratic optimal control problems. Further, we require the following lemma concerning induced matrix norms.

LEMMA 3.7 (Corollary 2.3.2. in [11]). *Let $A \in \mathbb{R}^{s\times s}$. Then it holds that*

$$\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}.$$

THEOREM 3.8 (Exponential Decay). *Assume that V possesses the γ-decaying sensitivity property with $\gamma(l) = C\rho^l$ for some $C > 0$ and $\rho \in (0,1)$, and that there exists a growth bound $r\colon \mathbb{N} \to \mathbb{N}$ for the underlying graph, satisfying $r(k) \leq \widehat{C}\mu^k$, $k \in \mathbb{N}$, for some $\widehat{C} > 0$ and $\mu \in [1, \rho^{-1})$. Then for all $l \in \mathbb{N}$ and all $x \in \Omega$ it holds*

$$|V(x) - \Psi_l(x) - V(0)| \leq \widetilde{C}\delta^{l+1}\|x\|_2^2,$$

where $\widetilde{C} > 0$ and $\delta \in (\rho\mu, 1)$ are independent of s and n.

Proof. Let $l \in \mathbb{N}$, D_l as in (3.4), $i \in \{1,\dots,s\}$, and $\delta \in (\rho\mu, 1)$. Then it holds

$$\sum_{j=1}^{s} |D_l[i,j]| = \sum_{j\in\{1,\dots,s\}:\mathrm{dist}_{\mathcal{G}}(i,j)>l} \gamma(\mathrm{dist}_{\mathcal{G}}(i,j)) \leq \sum_{k=l+1}^{\infty} \sum_{j:\mathrm{dist}_{\mathcal{G}}(i,j)=k} \gamma(k)$$
$$\leq C \sum_{k=l+1}^{\infty} r(k)\rho^k \leq C \sup_{k\in\mathbb{N}} \left\{ r(k) \left(\frac{\rho}{\delta}\right)^k \right\} \frac{\delta}{1-\delta}\delta^l.$$

Thus, we obtain

$$\|D_l\|_\infty = \max_{i\in[s]} \sum_{j\in[s]} |D_l[i,j]| \leq C \sup_{k\in\mathbb{N}} \left\{ r(k) \left(\frac{\rho}{\delta}\right)^k \right\} \frac{\delta^{l+1}}{1-\delta}$$

$$\leq C\widehat{C} \sup_{k\in\mathbb{N}} \left\{ \mu^k \left(\frac{\rho}{\delta}\right)^k \right\} \frac{\delta^{l+1}}{1-\delta} \leq C\widehat{C}\frac{\delta^{l+1}}{1-\delta}\delta^{l+1} =: \widetilde{C}\delta^{l+1}.$$

Analogously, we obtain the same bound for $\|D_l\|_1$. Finally, using Lemma 3.6 and Lemma 3.7 yields

$$|V(x) - \Psi_l(x) - V(0)| \leq \|x\|_2^2 \|D_l\|_2 \leq \sqrt{\|D_l\|_1 \|D_l\|_\infty} \leq \widetilde{C}\delta^{l+1}.$$

This shows the claim. ∎

Note that the growth bound required in Theorem 3.8 is met if the number of agents with a graph distance of l grows at most polynomial in l. If the growth rate is exponential, its base must be less than the reciprocal of the exponential decay base. Similarly to Theorem 3.8, one can establish a bound on the approximation error assuming a *polynomial* decay in sensitivity.

THEOREM 3.9 (Polynomial Decay). *Assume that V possesses the γ-decaying sensitivity property with $\gamma(l) = C(l+1)^{-\alpha}$ for some $C > 0$ and $\alpha > 1$. Furthermore, assume that there exists a growth bound $r\colon \mathbb{N} \to \mathbb{N}$ for the underlying graph and some $\beta < \alpha - 1$ such that $r \in \mathcal{O}(l^\beta)$ for $l \to \infty$. Then for all $l \in \mathbb{N}$ it holds that for all*

$x \in \Omega$

$$|V(x) - \Psi_l(x) - V(0)| \leq \widetilde{C}\|x\|_2^2 \sum_{k=l+2}^{\infty} k^{-\alpha+\beta},$$

where $\widetilde{C} > 0$ *is independent of* s *and* n.

Proof. Analogously to the proof of Theorem 3.8, we obtain

$$\|D_l\|_\infty \leq \sum_{k=l+1}^{\infty} r(k)C(k+1)^{-\alpha} \leq C \sup_{k\in\mathbb{N}} \left\{r(k)(k+1)^{-\beta}\right\} \sum_{k=l+2}^{\infty} k^{-\alpha+\beta}.$$

Again, combining Lemma 3.6 and Lemma 3.7 yields the claim with $\widetilde{C} := C \sup_{k\in\mathbb{N}} \left\{r(k)(k+1)^{-\beta}\right\}$. ∎

Remark 3.10. Theorems 3.8 and 3.9 are stated for a fixed dimension n and a fixed number of subsystems s. However, the resulting approximations are dimension-independent, as the approximation quality does not degrade with increasing s or n. Consequently, if we consider a family of problems with increasing dimension and the assumptions are satisfied uniformly with respect to the dimension, i.e., each member of the family admits the same γ-decaying sensitivity and the same growth-bound r, then the resulting approximations are also uniform in the dimension.

Remark 3.11 (Anchored decomposition). The separable approximation is strongly related to the notion of anchored decomposition [20,28]. For a given function $f : \mathbb{R}^n \to \mathbb{R}$ and an anchor $c \in \mathbb{R}^n$, the anchored decomposition of the function f with respect to the anchor c is defined as

$$f(x) = \sum_{u\subseteq[1,\dots,d]} f_{u,c}(x_u), \tag{3.7}$$

where $f_{u,c}$ satisfies the following properties

$$f_{\varnothing,c} = f(c), \quad f_{u,c}(x_u) = f((x_u; c)) - \sum_{v\subsetneq u} f_{v,c}(x_v). \tag{3.8}$$

Here the notation $(x_u; c) \in \mathbb{R}^d$ denotes the vector obtained by inserting the anchor c_i into every coordinate $i \notin u$ and the free variable x_i into every coordinate $i \in u$. Equivalently, if

$$y_i \;=\; \begin{cases} x_i, & i \in u, \\ c_i, & i \notin u, \end{cases} \qquad i = 1, \dots, d,$$

then

$$(x_u; c) \;=\; (y_1, \dots, y_d), \qquad f\big((x_u; c)\big) \;=\; f(y_1, \dots, y_d).$$

The key idea behind anchored decomposition is to express a high-dimensional function $f(z_1, \dots, z_s)$ as a sum of terms that depend on a progressively larger number of variables, starting from simpler, lower-dimensional components. This decomposition allows simplifying the computation by focusing on low-dimensional interactions (e.g., pairwise or three-variable interactions) when possible, making high-dimensional problems more tractable. The d-separable decomposition shares some terms of the anchored decomposition and neglects those which include indices far away according to the graph distance.

For example, let us consider the term $f_{[i,j,k],c}$, which accounts for three elements in the set $\{1,\dots,s\}$ and assume $d_{\mathcal{G}}(z_i,z_k) \gg 1$. We also introduce the notation $f_{k,c}(z_{i_1},\dots,z_{i_k})$, which is the evaluation of the function f with the variables not in the list fixed at the anchor c. To maintain consistency with the construction outlined in Section 3.2, we fix the anchor point c at the origin and henceforth omit its explicit dependence by writing $f_{[i,j,k]}$ and $f_k(x_{i_1},\dots,x_{i_k})$. By property (3.8), we have

$$f_{[i,j,k]} = f_3(z_i,z_j,z_k) - f_{[i,j]} - f_{[j,k]} - f_{[i,k]} - f_{[i]} - f_{[j]} - f_{[k]} - f_{\varnothing}. \tag{3.9}$$

Again, by definition,

$$f_{[i_1,i_2]} = f_2(z_{i_1},z_{i_2}) - f_{[i_1]} - f_{[i_2]} - f_{\varnothing}, \quad i_1,i_2 \in \{1,\dots,s\},$$

$$f_{[i_1]} = f_1(z_{i_1}) - f_{\varnothing}, \quad i_1 \in \{1,\dots,s\}.$$

Substituting these into (3.9) and canceling terms, we obtain

$$f_{[i,j,k]} = f_3(z_i,z_j,z_k) - f_2(z_i,z_j) - f_2(z_j,z_k) - f_2(z_i,z_k) + f_1(z_i) + f_1(z_j) + f_1(z_k) - f_{\varnothing}.$$

This expression can be further rewritten in terms of the functions $\{g_j\}_j$ introduced in (2.1), as follows

$$f_{[i,j,k]} = g_k(z_i,z_j,z_k) - g_k(z_j,z_k) - g_i(z_i,z_k) + g_i(z_i),$$

where, as before, any variables not explicitly included in the list of entries are fixed at the anchor point. Now, assume that the function f satisfies the γ-decaying sensitivity property introduced in Definition 2.2. Then, the term $f_{[i,j,k]}$ can be estimated using the Lipschitz continuity of the functions g_k and g_i

$$|f_{[i,j,k]}| \le 2\gamma(d_{\mathcal{G}}(z_i,z_k))\|z_k\|\|z_i\|.$$

We note that the decay of the element $f_{[i,j,k]}$ depends solely on the reciprocal distance and positions of the nodes z_i and z_k, rather than on the node z_j.

In summary, anchored decomposition and separable decomposition share a common goal of simplifying high-dimensional functions by focusing on lower-dimensional interactions. We note that each separable component Ψ_l^j can be interpreted as a localized aggregation of the anchored decomposition terms with anchor $c = 0$. Specifically, Ψ_l^j corresponds to the sum of all functions $f_{u,0}$ for which the index set u contains j and any $i \in u$ satisfies $i \ge j$ and $\mathrm{dist}_{\mathcal{G}}(i,j) \le l$. This yields the identity

$$\Psi_l^j(z_{j,l}) = \sum_{\substack{u \subseteq \{j,j+1,\dots,s\} \\ j \in u,\ \max_{i\in u} \mathrm{dist}_{\mathcal{G}}(i,j) \le l}} f_{u,0}(z_u),$$

which makes explicit the structural relationship between our separable representation (3.2) and the anchored decomposition. While anchored decomposition systematically includes all terms based on variable subsets, separable decomposition selectively neglects terms involving distant variables, leveraging graph-based distances. This distinction allows separable decomposition to achieve greater computational efficiency by emphasizing localized interactions, making it particularly suited for functions with decaying sensitivity over distance.

4. Sensitivity decay in linear-quadratic optimal control. In this section, we show that the decaying sensitivity property appears naturally in the infinite-horizon Linear Quadratic Regulator (LQR). In Subsection 4.1 we recall results from [33] (relying on [30]) for discrete-time problems while in Subsection 4.2 we prove a novel result for continuous-time problems, though under stronger conditions than in the discrete-time case.

Based on Setting 2.1, we define the state $x \in \mathbb{R}^n$ as being composed of $s \leq n$ substates $x = (z_1, \dots, z_s)$, where each substate z_j may have an associated control $u_j \in \mathbb{R}^{m_j}$ for $j \in \{1, \dots, s\}$. Whenever a substate does not have an associated control, we set $m_j = 0$. The collection of all controls is denoted by $u = (u_1, \dots, u_s) \in \mathbb{R}^m$, with $m = \sum_{j=1}^{s} m_j$. We denote by $\mathcal{U}_d = \ell^\infty(\mathbb{N}, \mathbb{R}^m)$ the space of all bounded control sequences in the discrete-time case, and by $\mathcal{U}_c = L^\infty([0, \infty), \mathbb{R}^m)$ the space of all essentially bounded control sequences in the continuous-time case.

Given matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $Q \in \mathbb{R}^{n\times n}$, $Q \succeq 0_n$, and $R \in \mathbb{R}^{m\times m}$, $R \succ 0_m$, we consider an infinite-horizon optimal control problem formulated either in discrete time

$$\min_{u\in\mathcal{U}_d} J(x_0, u) = \sum_{k=0}^{\infty} \ell(x(k), u(k)), \quad \text{s.t. } x(k+1) = f(x(k), u(k)),\ x(0) = x_0, \tag{4.1}$$

or in continuous time

$$\min_{u\in\mathcal{U}_c} J(x_0, u) = \int_0^{\infty} \ell(x(t), u(t))\, dt \quad \text{s.t. } \dot{x}(t) = f(x(t), u(t)),\ x(0) = x_0, \tag{4.2}$$

where $\ell(x, u) = x^\top Q x + u^\top R u$, $f(x, u) = Ax + Bu$, and $x_0 \in \mathbb{R}^n$ is an initial value. In both cases, we are concerned with computing the optimal value function

$$\begin{aligned} V &\colon \mathbb{R}^n \to \mathbb{R}, \\ &x_0 \mapsto \inf_{u\in\mathcal{U}} J(x_0, u), \end{aligned}$$

where $\mathcal{U} = \mathcal{U}_d$ in the discrete-time case and $\mathcal{U} = \mathcal{U}_c$ in the continuous-time case. In the LQR framework, it is well-known that the optimal value function V is quadratic, taking the form $V(x) = x^\top P x$, where P satisfies the Discrete Algebraic Riccati Equation for discrete-time systems, and the Continuous Algebraic Riccati Equation for continuous-time systems. As described in Setting 2.1, the graph $\mathcal{G}$ associated with such a problem is determined by the subsystems and their mutual influences. For the set of vertices $\{1, \dots, s\}$, the graph $\mathcal{G}$ has an edge (j, i) if

$$\begin{aligned} &A[i,j] \neq 0 \quad \text{or} \quad \Big(m_j > 0 \text{ and } B[i,j] \neq 0\Big) \quad \text{or} \quad Q[i,j] \neq 0 \quad \text{or} \\ &\Big(m_i > 0, m_j > 0 \text{ and } R[i,j] \neq 0\Big), \end{aligned} \tag{4.3}$$

where $A[i,j] \in \mathbb{R}^{n_i \times n_j}$ denotes the submatrix of A corresponding to the rows of z_i and the columns of z_j, $B[i,j] \in \mathbb{R}^{n_i \times m_j}$ denotes the submatrix of B corresponding to the rows of z_i and the columns of u_j, and analogously for $Q[i,j]$ and $R[i,j]$. In the case of interactions via control inputs, the conditions $m_j > 0$ and $m_i > 0, m_j > 0$ are imposed in (4.3) prior to the conditions on B and R, respectively, to accommodate subsystems that do not possess control inputs. Such subsystems can only influence other subsystems through the matrices A or Q.

Note that (4.3) establishes a connection from z_j to z_i if either z_j or u_j influences the dynamics of z_i, if there is an interaction in the state cost between z_j and z_i, represented by Q, or if there is an interaction in the control cost between u_j and u_i, captured by R. While the influence in the costs given by Q and R is always symmetric, leading to bidirectional connections in the graph, this symmetry does not generally hold for the dynamics described by A and B. Consequently, $\mathcal{G}$ may still be represented as a directed graph. Yet, the results from [30], on which we rely in Subsection 4.1, use an undirected graph and the results in Subsection 4.2 require symmetry of A. Thus, in the remainder of this section we only consider the case that $\mathcal{G}$ is undirected, which is easily accomplished by adding all edges (j, i) to $\mathcal{G}$ for which (i, j) is contained in $\mathcal{G}$.

As highlighted in Corollary 3.3, the decaying sensitivity of the optimal value function is associated with the SED property of the Riccati solution. In the first subsection, we address the discrete-time setting, where we revisit results previously derived in [33]. In the second subsection, we transition to the continuous-time setting and, under appropriate assumptions on the problem data, demonstrate an off-diagonal decay property of the Riccati solution. We introduce some notions that will be used in the following.

DEFINITION 4.1. *The pair (A, B) is called stabilizable if there exists a feedback matrix $K \in \mathbb{R}^{m\times n}$ such that $A - BK$ is stable.*

DEFINITION 4.2. *The pair (A, C) is called detectable if the pair $(A^\top, C^\top)$ is stabilizable.*

4.1. Discrete-Time Linear Quadratic Regulator. In this section, we examine the decaying sensitivity property in discrete-time linear quadratic systems. To this purpose, we revisit the results on exponential decay in the feedback matrix established in [30] and the decay in the Riccati solution as presented in [33]. Let us introduce the notion of stability in the discrete setting.

DEFINITION 4.3 (Stability in discrete time). *A matrix $C \in \mathbb{R}^{n\times n}$ is called stable in discrete time (or Schur), if all its eigenvalues lie in the unit circle, i.e. $|\lambda_i| < 1\ \forall\ \lambda_i \in \sigma(C)$, where $\sigma(C)$ is the spectrum of the matrix C.*

Note that for each $\mu > \max_{\lambda_i\in\sigma(C)} |\lambda_i|$ there is $\kappa > 0$ such that

$$\|C^k\| \leq \kappa\mu^k \tag{4.4}$$

holds all $k \in \mathbb{N}$. We therefore consider a family of infinite-horizon discrete optimal control problems expressed in the form (4.1), for which the following conditions hold uniformly, i.e., with the same constants $L, \eta, \kappa > 0$ and $\mu \in (0, 1)$ for all members of the family:

i) $\|A\|, \|B\|, \|Q\|, \|R\| \leq L$,
ii) (A, B) is uniformly stabilizable, i.e., Definition 4.1 holds with $\|K\| \leq L$ and $C = A - BK$ satisfies (4.4),
iii) $Q \succeq 0_n$ and $(A, Q^{\frac{1}{2}})$ is uniformly detectable, i.e., Definition 4.2 holds with $\|K\| \leq L$ and $C = (A - KQ^{\frac{1}{2}})^T$ satisfies (4.4),
iv) $R \succeq \eta I_m$.

The solution of the discrete-time infinite horizon optimal control problem relies on the solution of the Discrete Algebraic Riccati Equation (DARE):

$$P = A^T PA - A^T PB \left(R + B^T PB\right)^{-1} B^T PA + Q. \tag{4.5}$$

It has been proved in [30] that the feedback matrix

$$K = (R + B^\top P B)^{-1} B^T P A$$

exhibits a spatially exponential decaying (SED) property, *i.e.*

$$\|K[i,j]\| \leq C_K \rho_K^{\mathrm{dist}_{\mathcal{G}}(i,j)},$$

where the constants C_K and ρ_K are defined in Theorem 3.3 of [30]. Starting from this result, in [33] we extended the SED property to the matrix P, the solution of the DARE, as stated in the following result.

Proposition 4.4. Consider a family of LQR problems of the form (4.1) satisfying conditions i)–iv) from above, and assume that for $i, j \in \{1, \dots, s\}$ we have $dist_{\mathcal{G}}(i,j) = |i-j|$. Then, the solution P of the DARE (4.5) satisfies

$$\|P[i,j]\| \leq C_P \rho_P^{|i-j|}$$

for constants C_P and ρ_P independent of s.

Note that Proposition 4.4 is stated for the case $\mathrm{dist}_{\mathcal{G}}(i,j) = |i-j|$, but its proof extends naturally to any graph where $|\{j \mid \mathrm{dist}_{\mathcal{G}}(i,j) \leq 1\}|$ remains bounded independently of the dimension for all $i \in \{1, \dots, s\}$. For example, if $B = Q = R = I_n$ and A is a p-banded matrix for some $p \in \mathbb{N}$, it follows that $|\{j \mid \mathrm{dist}_{\mathcal{G}}(i,j) \leq 1\}| \leq 2p+1$ for any dimension and any node i.

4.2. Continuous-Time Linear Quadratic Regulator. In this section, we show that a similar property also holds for continuous-time LQR problems of the form (4.2). Since this is not the main focus of this paper, we do not reproduce the reasoning from [30, 33] but rather present a different proof, which is significantly shorter and of interest on its own, but which also requires stricter assumptions.

As previously mentioned, the optimal value function takes the form $V(x) = x^\top P x$, where $P \in \mathbb{R}^{n \times n}$ satisfies the Algebraic Riccati Equation (ARE)

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0. \tag{4.6}$$

To ensure the existence and uniqueness of a positive semi-definite solution to the ARE, we assume that the system is stabilizable and detectable according to Definition 4.1 and Definition 4.2, respectively, where the notion of stability in the continuous setting is provided in the following definition.

Definition 4.5 (Stability in continuous time). *A matrix $C \in \mathbb{R}^{n \times n}$ is called stable in continuous time (or Hurwitz), if all its eigenvalues lie in the open left half complex plane, i.e. $Re(\lambda_i) < 0 \ \forall \ \lambda_i \in \sigma(C)$, where $Re(\cdot)$ stands for the real part.*

Theorem 4.6. *Given a stabilizable pair (A, B) and a detectable pair $(A, Q^{1/2})$, the ARE equation* (4.6) *admits a unique stabilizing symmetric positive semidefinite solution P.*

For a detailed discussion of this result, we refer to [21]. Throughout this section, the assumptions of Theorem 4.6 will be maintained to ensure the well-posedness of the problem. We now turn to a specific case where the solution to the ARE can be found in closed form.

Proposition 4.7. Given a symmetric matrix A, and matrices $B = I_n$, $Q = cI_n$ and $R = \gamma I_n$, with $c, \gamma > 0$, the unique positive semi-definite solution of the ARE (4.6) is given by:

$$P = \gamma \left(\sqrt{A^2 + \frac{c}{\gamma} I_n} + A \right). \tag{4.7}$$

Proof. First, observe that $A^2 + \frac{c}{\gamma} I_n \succ 0$, ensuring that the matrix square root in the expression for P is well defined. Furthermore, since $\sqrt{\lambda^2 + \frac{c}{\gamma}} > |\lambda|$ and $\gamma > 0$, the matrix in (4.7) is positive definite. Substituting expression (4.7) into the ARE (4.6) confirms that the expression for P satisfies the equation. ∎

We now introduce the concept of functions of diagonalizable matrices, which will be instrumental in the subsequent analysis. For further details and alternative definitions, the reader is referred to [16].

Definition 4.8 (Function of a diagonalizable matrix). *Let $A \in \mathbb{R}^{n \times n}$ be a diagonalizable matrix. Then there exists an invertible matrix $X \in \mathbb{C}^{n \times n}$ and a diagonal matrix $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n) \in \mathbb{C}^{n \times n}$, containing the eigenvalues of A, such that $A = X \Lambda X^{-1}$. Let $f(z)$ be a scalar-valued function (real or complex) that is defined at each eigenvalue λ_i of A. The function of the matrix A, denoted $f(A)$, is then defined by*

$$f(A) = X f(\Lambda) X^{-1},$$

where

$$f(\Lambda) = \operatorname{diag}(f(\lambda_1), f(\lambda_2), \ldots, f(\lambda_n)).$$

Then, the expression (4.7) can be reformulated as $P = f(A)$, where

$$f(x) = \gamma \left(\sqrt{x^2 + \frac{c}{\gamma}} + x \right), \quad x \in \mathbb{C}. \tag{4.8}$$

We note that if A is a banded matrix with spectrum in $[a, b]$, the scaled and shifted matrix

$$\widetilde{A} = \psi(A) := \frac{2}{b-a} A - \frac{a+b}{b-a} I_n$$

remains banded with the same bandwidth, and its spectrum is contained in $[-1, 1]$. Moreover, $f(A) = f(\psi^{-1}(\widetilde{A}))$. Accordingly, we focus on bounding the off-diagonal entries of the matrix function $F(\widetilde{A})$, where for $x \in \mathbb{C}$

$$F(x) := f(\psi^{-1}(x)) = \gamma \left(\sqrt{\left(\frac{b-a}{2} x + \frac{a+b}{2} \right)^2 + \frac{c}{\gamma}} + \frac{b-a}{2} x + \frac{a+b}{2} \right), \tag{4.9}$$

and $\widetilde{A}$ is banded with spectrum in $[-1, 1]$. The following useful result is adapted from [5, Theorem 2.2].

Theorem 4.9. *Let $g : \mathbb{C} \to \mathbb{C}$ be a function analytical in the interior of an ellipse $\mathcal{E}_\chi$ with foci at ± 1 and sum of the half axes $\chi > 1$, and continuous on $\mathcal{E}_\chi$. Additionally, let $A \in \mathbb{C}^{n \times n}$ be symmetric, p-banded, and such that $[-1, 1]$ contains its spectrum, and let $\rho := \chi^{-\frac{1}{p}} \in (0, 1)$. Then, for all pairs of indices (i, j) such that $i \neq j$, we have that*

$$|g(A)[i, j]| \leq K \rho^{|i-j|},$$

where

$$K = \frac{2\chi M(\chi)}{\chi - 1}, \quad M(\chi) = \max_{z \in \mathcal{E}_\chi} |g(z)|.$$

We now formally state the result on the off-diagonal decay of the solution of the CARE equation (4.6), utilizing the previously mentioned theorem and the closed-form solution (4.8).

Proposition 4.10. Let $A \in \mathbb{R}^{n \times n}$ be a p-banded symmetric matrix with spectrum in $[a, b]$, and let f be the function in (4.8) that solves the Riccati equation (4.6) with $B = I_n$, $Q = cI_n$, and $R = \gamma I_n$. Then for all pairs of indices (i, j) such that $i \neq j$, we have that

$$|P[i,j]| = |f(A)[i,j]| \leq K\rho^{|i-j|}, \quad i \neq j, \tag{4.10}$$

with

$$\rho = \chi^{-\frac{1}{p}}, \qquad \chi = \sqrt{\theta} + \sqrt{\theta + 1}, \qquad \theta = \frac{2}{\gamma(b-a)^2}\left(\sqrt{(a^2\gamma + c)(b^2\gamma + c)} + ab\gamma + c\right),$$

$$K = \frac{2\chi M_F(\chi)}{\chi - 1}, \qquad M_F(\chi) := \max_{z \in \mathcal{E}_\chi} |F(z)| = F\left(\sqrt{\theta + 1}\right), \qquad F(x) = f(\psi^{-1}(x)),$$

and $\mathcal{E}_\chi$ is the ellipse with foci at ± 1 and sum of the half axes $\chi > 1$.

Proof. We have $f(A) = f(\psi^{-1}(\widetilde{A})) = F(\widetilde{A})$ and the function $F(x)$ (given in (4.9)) satisfies the conditions of Theorem 4.9. In particular, it is analytical in any ellipse with foci in -1 and 1 that do not include, in its interior part, the points

$$x_{1/2} = -\frac{b+a}{b-a} \pm i\, 2\frac{\sqrt{c}}{\sqrt{\gamma}(b-a)}.$$

Thus, we consider the ellipse $\mathcal{E}_\chi$ with foci in -1 and 1 passing through the points $x_{1/2}$. The sum of the half axes of $\mathcal{E}_\chi$ is given by

$$\chi = \sqrt{\theta} + \sqrt{\theta + 1}, \qquad \theta = \frac{2}{\gamma(b-a)^2}\left(\sqrt{(a^2\gamma + c)(b^2\gamma + c)} + ab\gamma + c\right).$$

Applying Theorem 4.9 yields the desired result. It can be verified that the maximum of $|F(z)|$ on the ellipse occurs at $z = \sqrt{\theta + 1}$. ∎

Example 4.11 (Heat equation). We consider the optimal control problem for the heat equation with homogeneous Neumann boundary conditions, given by:

$$\partial_t y(t,x) = \sigma \partial_{xx} y(t,x) + u(t,x),$$

with $x \in [0,1]$ and $t \in (0, +\infty)$. The corresponding cost functional is defined as:

$$\tilde{J}(y_0, u) = \int_0^\infty \int_0^1 (|y(t,x)|^2 + \tilde{\gamma}|u(t,x)|^2) dx\, dt\,.$$

By discretizing the PDE using finite difference schemes with a spatial stepsize $\Delta x = 5 \cdot 10^{-3}$, we approximate the system by the following system of ODEs:

$$\dot{y}(s) = Ay(s) + Bu(s),$$

where A is a tridiagonal matrix corresponding to the discretized Laplacian with Neumann boundary conditions. The discretized cost functional reads

$$J(y_0, u) = \int_0^\infty \Delta x \, y(t)^\top y(t) + \gamma \, u(t)^\top u(t) \, dt \,,$$

with $\gamma = \tilde{\gamma}\Delta x$ and $n = 200$, the number of discretized spatial points. We set $\tilde{\gamma} = 0.01$ and consider two different values for the diffusion coefficient σ, namely 10^{-3} and 10^{-2}.

In Figure 3, we present a comparison between the decay of the absolute value of the first column of the solution to the CARE (4.6) and the upper bound derived in Proposition 4.10 for diffusion coefficients $\sigma = 10^{-3}$ (left panel) and $\sigma = 10^{-2}$ (right panel). In both cases, the bound correctly captures the decay behavior, though a shift is observed. This discrepancy may be attributed to the constant K in (4.10), which could potentially be refined through sharper estimates. Additionally, it is evident that increasing the diffusion coefficient results in a slower decay. Indeed, a higher diffusion leads to a faster communication between nodes and, consequently, to a slower decay of sensitivity.

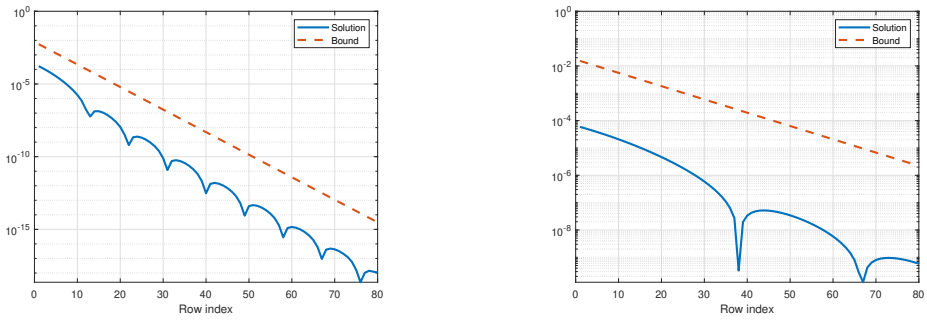

Figure 3: Decay of the first column's absolute value of the solution of the CARE (4.6) and the corresponding upper bound (4.10) with $\sigma = 10^{-3}$ (left) and $\sigma = 10^{-2}$ (right).

5. Efficient Neural Network Representation for Separable Structures. In this section, we examine how neural networks can efficiently represent functions exhibiting decaying sensitivity. While neural networks are universal approximators, they typically require an exponential growth in neurons and parameters as input dimensionality increases. However, we demonstrate that for functions V possessing a separable approximation, this curse of dimensionality can be mitigated. Under appropriate conditions, such functions can be represented with neural networks requiring a number of parameters which only grows polynomially. This result is consistent with existing literature showing that certain structural properties enable polynomial parameter scaling with dimension. For example, a similar result holds for compositional functions (cf. [17]) and in particular, for the sub-class of separable functions. The approximation of separable functions with neural networks has been discussed in [12, 32] in the context of (control) Lyapunov functions. While the approach and the neural network construction in the following theorem is similar, there are two major differences compared to these works. The first one is that [12, 32] considers strictly separable functions, meaning that each component x_i of the state x appears in exactly one subsystem. This is not the case in the framework of this paper, since

we achieve separability by considering *overlapping* graph neighborhoods, whence we cannot directly apply the results in [12, 32] towards our setting. On the other hand, in [12, 32] the authors used a linear transformation as the first hidden layer for the computation of suitable substates to achieve separability. This is not necessary in our case, since for any given length l of the graph-neighborhoods, the vectors $z_{j,l}$ are uniquely determined by the graph $\mathcal{G}$.

In the following, a neural network refers to a feedforward network with a one-dimensional output and a linear activation function in its output neuron. We associate each neural network with its corresponding function, denoted by $W(x;\theta)\colon \mathbb{R}^n \to \mathbb{R}$, where $x \in \mathbb{R}^n$ represents the input vector and θ denotes the trainable parameters.

THEOREM 5.1. *Let $R > 0$ and consider the setting of Theorem 3.8 with $\Omega := [-R, R]^n$. Assume that V is continuously differentiable on Ω with $\|V\|_{\mathcal{C}^1(\Omega)} \leq C_V$ for some $C_V > 0$. Let $\sigma \in \mathcal{C}^\infty(\mathbb{R}, \mathbb{R})$ be not a polynomial. Then for any $\varepsilon > 0$ there exists a feedforward neural network $W(x;\theta)$ with activation function σ and a corresponding parameter vector θ such that for any $x \in \Omega$*

$$|W(x;\theta) - V(x)| \leq \varepsilon(1 + \|x\|_2^2).$$

The number of neurons $N(W)$ in the network $W(x;\theta)$ and its number of trainable parameters $P(W)$ are bounded by

$$N(W) \leq s^{d+1}C^d + n + 1, \quad P(W) \leq s^{d+1}C^d(d + s + 1) + 1,$$

respectively, where $C > 0$ and $d > 0$ are independent of s and n, but are determined by ε, R, $\widetilde{C}$, δ, and $d_{j,l}$, where $\widetilde{C}$ and δ are defined as in Theorem 3.8, and $d_{j,l}$ denotes the dimension of the subvectors $z_{j,l}$ introduced in Subsection 3.2, with l itself depending on ε, $\widetilde{C}$, and δ.

Proof. Let $l := \left\lceil \ln\left(\epsilon\widetilde{C}^{-1}\right)\ln^{-1}(\delta) - 1 \right\rceil \in \mathbb{N}$, where $\lceil\cdot\rceil$ stands for the ceiling function. Let Ψ_l be as defined in (3.2) and (3.3). Then, by Theorem 3.8, we obtain that for all $x \in \Omega$

$$|V(x) - \Psi_l(x)| \leq \widetilde{C}\delta^{l+1}\|x\|_2^2 \leq \epsilon\|x\|_2^2.$$

Thus, it is sufficient to show the existence of a neural network that approximates Ψ_l on Ω with error bounded by ϵ. We begin by constructing subnetworks for approximating Ψ_l^j, $1 \leq j \leq s$. Recall that Ψ_l^j depends on $z_{j,l} \in \mathbb{R}^{d_{j,l}}$. Since σ is non-polynomial, by Theorem 2.1 in [23] and its extension to cubes with length R in [17], there exists a constant $\mu_{d_{j,l}}$ such that any neural network $W_j(\cdot;\theta_j)\colon \mathbb{R}^{d_{j,l}} \to \mathbb{R}$ consisting of one hidden layer with $M \in \mathbb{N}$ neurons and activation function σ satisfies

$$\inf_{\theta_j} |W_j(z_{j,l};\theta_j) - \Psi_l^j(z_{j,l})|_{\infty,[-R,R]^{d_{j,l}}} \leq \mu_{d_j,l}\,\widetilde{R}\,M^{-\frac{1}{d_{j,l}}}\,\|\Psi_l^j\|_{C^1([-R,R]^{d_{j,l}})},$$

where $\widetilde{R} := \max\{1, R\}$. Since, by assumption, V has a C^1-norm bounded by C_V on $[-R, R]^n$, it follows from (3.3) that Ψ_l^j has C^1-norm bounded by $2C_V$ on $[-R, R]^{d_{j,l}}$, leading to

$$\inf_{\theta_j} |W_j(z_{j,l};\theta_j) - \Psi_l^j(z_{j,l})|_{\infty,[-R,R]^{d_{j,l}}} \leq 2C_V\,\mu_{d_j,l}\,\widetilde{R}\,M^{-\frac{1}{d_{j,l}}}. \tag{5.1}$$

We now construct the neural network $W(x;\theta)$ as concatenation of the s subnetworks $W_1, \ldots, W_s$. The network $W(x;\theta)$ takes an n-dimensional input vector x, with each

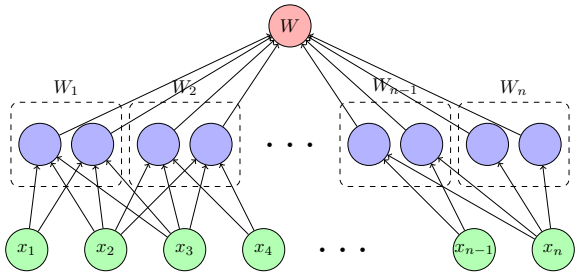

Figure 4: Illustration of the separable structured neural network (S-NN) described in the proof of Theorem 5.1 for the case of $s = n$ one-dimensional subsystems with $l = 2$ and $\mathrm{dist}_{\mathcal{G}}(i, j) = |i - j|$.

subnetwork W_j then independently processing its corresponding input $z_{j,l}$, connected to the relevant components of x. The outputs of the subnetworks are aggregated via summation to compute $W(x;\theta)$. Since each W_j outputs a linear combination of its M hidden-layer neurons, these combinations can be unified into a single linear layer connecting the sM neurons of the subnetworks to the output neuron of $W(x;\theta)$. By (5.1), the network $W(x;\theta)$ then satisfies

$$\begin{aligned}\inf_{\theta} |W(x;\theta) - \Psi_l(x)|_{\infty,\Omega} \leq& \sum_{j=1}^{s} \inf_{\theta_j} |W_j(z_{j,l};\theta_j) - \Psi_l^j(z_{j,l})|_{\infty,[-R,R]^{d_{j,l}}} \\ \leq& 2sC_V\mu_d\, \widetilde{R}\, M^{-\frac{1}{d}},\end{aligned}$$

where $d \coloneqq \max_j d_{j,l}$. Solving for M yields $M \geq s^d(2C_V\mu_d\widetilde{R})^d \frac{1}{\varepsilon^d}$. To establish the existence of a parameter θ that achieves the desired error bound ε, we choose

$$M = s^d \left\lceil 2C_V\mu_d\widetilde{R}\varepsilon^{-1} \right\rceil^d =: s^d C^d$$

hidden neurons for each network W_j. Hence, the resulting neural network W has at most $sM = s^{d+1}C^d$ hidden neurons, n input neurons, and one output neuron. Since each subnetwork W_j contains $d_{j,l}M$ weights and M bias terms, and the final linear layer connecting the sM neurons to the output neuron involves sM weights and one bias term, the total number of parameters is

$$\sum_{j=1}^{s}(d_{j,l}M + M) + sM + 1 \leq s^{d+1}C^d(d + s + 1) + 1.$$

This completes the proof. ∎

Besides the theoretical value of the result above, it also provides a systematic approach for designing a neural network with a separable structure, referred to as a Separable-structured Neural Network (S-NN). Figure 4 illustrates the architecture of the neural network W as constructed in the proof of Theorem 5.1. This example considers the case of one-dimensional subsystems arranged in a sequential graph, where the graph distance is defined as $\mathrm{dist}_{\mathcal{G}}(i, j) = |i - j|$, with $s = n$ and $z_j = x_j$ for $1 \leq j \leq n$. The selected graph-distance parameter is $l = 2$, which determines the

connectivity of the subnetworks. Specifically, the subnetwork W_1 is connected to the input nodes x_1, x_2, and x_3, while W_2 is connected to x_2, x_3, and x_4. This connectivity pattern continues sequentially, ensuring that subnetwork W_j is connected to input nodes $x_j, \ldots, x_{j+l}$. The final subnetwork W_n only processes the last subsystem given by x_n.

Theorem 5.1 provides an upper bound on the size of a neural network required to represent V under the assumption of exponential decay, as stated in Theorem 3.8 A similar result also holds under the assumption of polynomial decay, as considered in Theorem 3.9, with the difference that the constants C and d in Theorem 5.1 are increased. For conciseness, the corresponding result is omitted.

Remark 5.2. Theorem 5.1 states that, under the assumption of exponentially decaying sensitivity, neural networks can approximate V with a number of neurons and parameters that grows polynomially with respect to the number of subsystems s, where the exponent is d. The maximum dimension of a graph-neighborhood d depends on l, which is determined by $\varepsilon, \widetilde{C}$, and δ, but independent of s and n. Thus, for a family of problems with increasing dimension and uniform decaying sensitivity, where, for any fixed l, the maximum neighborhood dimension d remains constant, Theorem 5.1 describes a situation where the curse of dimensionality is avoided. This applies, for example, to a sequential graph with one-dimensional subsystems, where $d = l + 1$ for any number of subsystems. However, it is important to note that Theorem 5.1 does not address the required growth of the network in relation to the accuracy ε.

6. Numerical Experiments: Neural Network Approximations for High-Dimensional Problems. In this section, we provide a detailed analysis of the capability of neural networks to approximate solutions for high-dimensional problems with decaying sensitivity, as stated in Theorem 5.1. The first numerical experiment focuses on a LQR example, where the data of the optimal control problem exhibits a banded structure. In the second example, we consider a more general setting, in which the function is no longer quadratic but continues to display sensitivity decay. The numerical simulations reported in this paper are performed on an Nvidia GeForce RTX 3070. The codes are written in Python 3.9 with TensorFlow 2.10 and are available at https://github.com/MarioSperl/SNN-DecayingSensitivity.

6.1. Linear quadratic optimal control problems. In this subsection, we first introduce a training algorithm designed to enable a neural network to approximate the optimal value function of linear quadratic optimal control problems exhibiting exponentially decaying sensitivity. Subsequently, we highlight the benefits of the separable network architecture depicted in Figure 4 compared to a conventional fully connected network. Finally, we present a numerical experiment that investigates the impact of varying the problem dimension, illustrating how the number of neurons and parameters scales with increasing dimensions in linear quadratic problems.

Consider a continuous-time linear quadratic optimal control problem as described in (4.2). The system matrix $A \in \mathbb{R}^{n\times n}$ is randomly generated with a bandwidth of 1, *i.e.* the matrix is tridiagonal, and entries uniformly distributed in $(0, 1)$, using the Python command `numpy.random`. Additionally, the matrices B, Q and R are all set to the identity matrix I_n. This induces a graph structure as in Figure 1 with $\mathrm{dist}_{\mathcal{G}}(i, j) = |i - j|$. From Proposition 4.10 it follows that the solution P of the corresponding ARE (4.6) is an SED-matrix, which by Corollary 3.3 implies that $V(x) = x^T P x$ exhibits an exponentially decaying sensitivity. For dimension $n = 200$,

the decay of the first and hundredth row in P are shown in the right column in Figure 5. To train the neural network to approximate the value function V on the domain $\Omega = [-1, 1]^{200}$, we adopt a supervised learning approach with precomputed, randomly generated, and independently chosen training, validation and test data $\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{te}} \subset \Omega$. These datasets are created by uniformly sampling data points from Ω and computing the corresponding values of V using the solution P of the ARE. The Mean Squared Error (MSE) on $\mathcal{D} \in \{\mathcal{D}_{\text{tr}}, \mathcal{D}_{\text{val}}, \mathcal{D}_{\text{te}}\}$ is given as

$$\frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} (V(x) - W(x; \theta))^2. \tag{6.1}$$

Training is conducted on $\mathcal{D}_{\text{tr}}$ using a batch-size of 64, the ADAM optimizer with its default learning rate in TensorFlow, the sigmoid activation function $\sigma(x) = \frac{1}{1+e^{-x}}$, and TensorFlow's default parameter initialization. These choices of hyperparameters for training have been applied consistently to all examples presented in this section. Every 10 epochs, the MSE on the validation dataset $\mathcal{D}_{\text{val}}$ is evaluated to check whether it falls below a predefined tolerance ε. If this criterion is met or if the maximum number of epochs is reached, the training process is terminated. The final model is then verified by calculating the MSE on the test dataset.

Particularly in high-dimensional settings, it turned out that incorporating information about the gradient and the values at the origin into the training process can offer notable advantages. To address this, we define a modified loss function $L \colon \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}^n$ as follows:

$$\begin{aligned} L(x, W(x;\theta), \nabla W(x;\theta)) \coloneqq &(V(x) - W(x;\theta))^2 + \nu_g \|\nabla V(x) - \nabla_x W(x;\theta)\|_2^2 \\ &+ \nu_z \Big(W(0;\theta)^2 + \|\nabla_x W(0;\theta)\|_2^2 \Big). \end{aligned}$$

Table 1 compares training using the loss function $\nu_g = \nu_z = 0.5$ with training based on the standard MSE loss, which corresponds to setting $\nu_g = \nu_z = 0$ in L. The example is conducted using a fully connected neural network with a single hidden layer comprising 256 neurons across varying dimensions n of the described LQR problem with A having a bandwidth of 1. The network was trained with $|\mathcal{D}_{\text{tr}}| = |\mathcal{D}_{\text{val}}| = 2^{17}$ data points uniformly randomly sampled in $[-1, 1]^n$ until the MSE on the validation set as defined in (6.1) reached a tolerance of 10^{-3}. The results demonstrate that incorporating positive values for ν_g and ν_z significantly improves training efficiency, especially in high-dimensional settings. In particular, in the 50-dimensional case, setting $\nu_g = \nu_z = 0.5$ allows the network to reach the tolerance after 240 epochs, whereas training with $\nu_g = \nu_z = 0$ results in an MSE of 1.3×10^{-2} on the validation data after 10^4 epochs.

Dimension	$\nu_g = \nu_z = 0.5$		$\nu_g = \nu_z = 0$	
	Result	Epochs	Result	Epochs
10	Tolerance reached	10	Tolerance reached	50
20	Tolerance reached	20	Tolerance reached	190
50	Tolerance reached	240	Stopped after max. epochs	1000

Table 1: Experimental results for different dimensions and choices of ν_g and ν_z in L.

Training with the loss function L requires the computation of the gradient of the neural network with respect to the input, prior to evaluating the gradient of the loss with respect to the network parameters θ. This is achieved using TensorFlow's gradient tape, which enables automatic differentiation by recording operations on tensors during the forward pass and computing gradients during the backward pass. The benefits of such a modified loss function in the context of control systems have also been emphasized in the literature. For example, [1, 3, 10, 19] discuss the inclusion of the information on the derivative, while [7, 18] consider the incorporation of the loss at the origin. Note that while the loss function L is used for training, the mean squared error as defined in (6.1) is computed for the validation and test data.

We now consider the case $n = 200$ and employ a Separable-structured Neural Network (S-NN) as illustrated in Figure 4, characterized by a graph distance of $l = 10$ and a sublayer-size of $M = 16$. This configuration results in $s = 200$ subnetworks, each connected to at most 11 input nodes, with 16 nodes in its hidden layer and one output node. Consequently, the total number of trainable parameters, determined by the weights and biases of the neural network, amounts to 40721.

Figure 5 depicts the optimal value function V and the trained neural network $W(x;\theta)$ projected to the $(x_1, x_2)-$ axis and the (x_{100}, x_{200}) axis in the first two columns. The training was performed with $\nu_g = \nu_z = 0.5$, $|\mathcal{D}_{\text{tr}}| = |\mathcal{D}_{\text{val}}| = 131072$ points, and has reached a tolerance of $\varepsilon = 10^{-3}$ after 200 epochs. The MSE, computed using $|\mathcal{D}_{\text{te}}| = 2^{19}$ test points, is $5.4 \cdot 10^{-4}$.

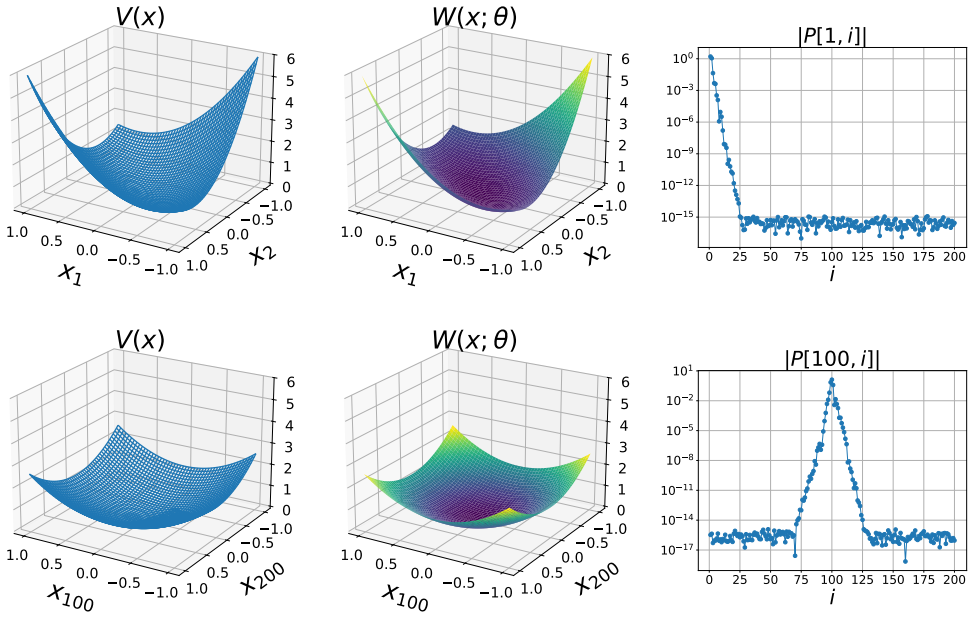

Figure 5: Plot of the optimal value function V projected onto the (x_1, x_2)-axis (upper left) and onto the (x_{100}, x_{200})-axis (lower left), the S-NN approximation onto the same hyperplanes (middle column), and the decay in the first row of P (upper right) and the row 100 of P (lower right).

From a theoretical perspective, any function that can be represented using the

neural network architecture described in Figure 4 can also be represented by a standard feedforward neural network with one hidden layer. However, by explicitly encoding the separable structure within the network architecture, we observe a significant improvement in training performance compared to a fully connected feedforward network. Table 2 provides a brief comparison of a fully connected network and a S-NN for the 200-dimensional LQR problem using the same training parameters as in the experiment corresponding to Figure 5. When the matrix A is fully connected, lacking any decaying property, the algorithm fails to converge for both network structures, with particularly poor performance observed for the S-NN. However, when A has bandwidth 1, introducing an exponential decaying sensitivity property, the fully connected network shows slightly improved performance but still fails to achieve the desired error tolerance within 10^4 training epochs. In contrast, the S-NN meets the tolerance criterion $\varepsilon = 10^{-3}$ after 200 epochs.

bandwidth	**(sub)-layersize**	**architecture**	l	**test error**	**epochs**
200	2048	fully conn.	-	1.9×10^{0}	1000
200	16	S-NN	10	8.0×10^{3}	1000
1	2048	fully conn.	-	3.5×10^{-1}	1000
1	16	S-NN	10	5.4×10^{-4}	200

Table 2: A comparison of test errors for a 200-dimensional example conducted under varying bandwidths, emphasizing the effect of utilizing a separable network structure.

Another criterion demonstrating the advantage of the S-NN architecture is the reduced number of training points and epochs required to achieve a stopping tolerance of $\varepsilon = 10^{-3}$. Consider a 50 dimensional example where the matrix A again has a bandwidth of 1. We compare a fully connected single layer network with 13313 parameters to a S-NN with $l = 10$ and 9521 parameters. Figure 6 presents the results obtained by varying the number of training points $|\mathcal{D}_{\text{tr}}|$ uniformly randomly chosen in $[-1, 1]^{50}$. The maximum number of epochs was adjusted based on the size of the training dataset to ensure the algorithm had sufficient time to train: specifically, it was set to 1000 for 2^{17} training points, 2000 for 2^{16} points, and so on, doubling with each halving of the training set size. As before, the training loss parameters were set to $\nu_g = \nu_z = 0.5$. We observe that, on one hand, the separable structured network requires significantly fewer training points to reach the desired tolerance in the test dataset, see Figure 6. On the other hand, for larger amounts of training data, where both architectures meet the tolerance, the S-NN requires considerably fewer epochs, see Figure 7.

Furthermore, we compute approximations of the optimal value function V for linear quadratic optimal control problems with increasing state dimensions, ranging from 20 to 120 in increments of 10. Specifically, we considered two different cases: one where the matrix A has bandwidth 1, and another where A has bandwidth 3. We use a S-NN as shown in Figure 4 with $n = s$ subnetworks and $\nu_g = \nu_z = 0.5$. The training dataset size was fixed at 4069 points, uniformly sampled in the unit ball, ensuring consistency with the framework established in Theorem 3.8. The graph-distance l was fixed to 3 in all cases. For each dimension, the size of the subsystems' layers was minimized while maintaining an MSE in the test data below 10^{-4}. It turned out that for both bandwidths and across all dimensions, only $M = 2$ neurons per sublayer were

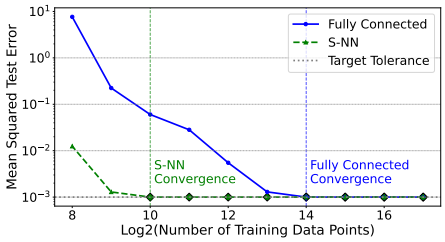

Figure 6: Comparison of test error (log-scale) against the number of training points.

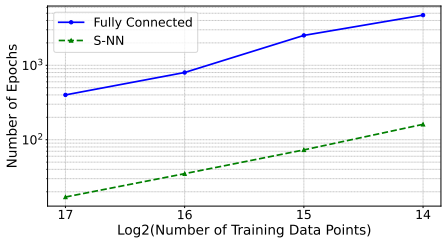

Figure 7: Comparison of training epochs (log-scale) against the number of training points.

required to satisfy this error threshold. Consequently, the total number of neurons and parameters increases linearly with the dimensionality of the problem, as illustrated in Figure 8. These findings suggest that the polynomial bound established in the theorems in Section 3 may not be sharp for LQR problems. It is important to note that, while the graph for the case with bandwidth equal to 1 is sequential, as shown in Figure 1, the increased number of connections in the case of bandwidth equal to 3 (depicted in Figure 9) leads to higher-dimensional graph neighborhoods. Specifically, for our choice of $l = 3$ we have $d = 4$ for the first case and $d = 10$ for the second. This, in turn, results in a larger number of trainable parameters for the setting with bandwidth equal to 3, as shown in Figure 8.

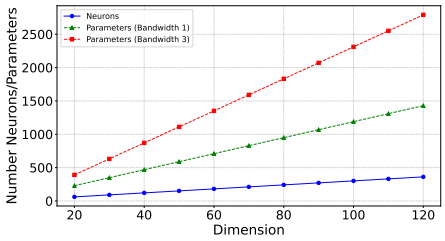

Figure 8: Number of neurons and parameters for bandwidths 1 and 3 in dependence of the dimension.

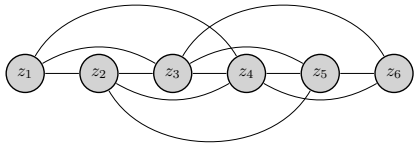

Figure 9: Graph corresponding to A with bandwidth equal to 3 and $B = Q = R = I_n$.

Remark 6.1. While the experiments in this section involving the S-NN network structure exploit the decaying sensitivity in the P-matrix to approximate $V(x) = x^T P x$, we would like to point out that this decay does not imply a decay in the singular values of P. This contrast is illustrated in Figure 10, which shows the decay of the first column of P (left) and the singular values of P (right), for a linear quadratic problem with tridiagonal $A \in \mathbb{R}^{n\times n}$ and $B = Q = R = I_n$, with $n = 100$. We conclude that standard low-rank approximation techniques cannot exploit the decaying sensitivity property.

6.2. Approximation of a non-quadratic function. In this subsection, we briefly discuss an application of the training algorithm presented in Subsection 6.1 towards the approximation of non-quadratic functions. To this end, for $\rho \in (0, 1]$ we

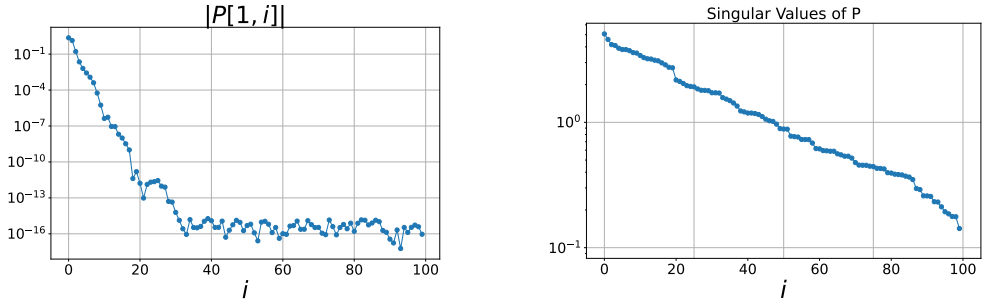

Figure 10: Comparison of the decaying sensitivity with the singular values of P.

introduce the function $V\colon [-1,1]^{100} \to \mathbb{R}$ defined as

$$V(x) = \sum_{i,j=1}^{n} \sin(x_i)\sin(x_j)\rho^{|i-j|},$$

and consider a graph $\mathcal{G}$ with nodes $\{1,\dots,n\}$ and sequential connections leading to $\operatorname{dist}_{\mathcal{G}}(i,j) = |i-j|$ for $i,j \in \{1,\dots,n\}$. Note that the function V is not separable with respect to Definition 3.4 since each entry of its Hessian matrix is non-zero. However, for $\rho < 1$ it exhibits an exponentially decaying sensitivity with basis ρ. We employ a S-NN with a distance for the graph-neighborhoods of $l = 5$, a sublayer-size of $M = 16$ in each subnetwork, $\nu_g = \nu_z = 0.5$, and 8192 training, validation, and test data. The resulting MSE after training for 10^4 epochs is shown in Figure 11, where we observe that the function is not well approximated for $\rho = 1$, which corresponds to the case without decaying sensitivity. In contrast, for the case with exponentially decaying sensitivity, where $\rho = \frac{1}{8}$, the MSE falls below 10^{-4}.
A similar behavior can be observed if we replace the exponential decaying sensitivity of V with a polynomial decay. That is, we consider the function $\widetilde{V}\colon [-1,1]^{100} \to \mathbb{R}$ given as

$$\widetilde{V}(x) = \sum_{i,j=1}^{n} \sin(x_i)\sin(x_j)(|i-j|+1)^{-\alpha},$$

for different values of $\alpha \in \{0,1,2,3\}$. We use the same network architecture and training parameters as for V and observe that the stronger the polynomial degree is, the better the resulting network approximation is, see Figure 12.

It follows that the separable structured neural network facilitates the approximation of general functions, with the degree of effectiveness being influenced by the rate of decay of the sensitivity. Specifically, the network architecture improves the ability to capture functions with decaying sensitivity, with the approximation quality being directly related to the rate at which the sensitivity diminishes. As the sensitivity decays more rapidly, the separable structure becomes increasingly advantageous, enabling more accurate and efficient representations of the function in question.

6.3. Nonlinear optimal control problem. Finally, we turn our attention to a more challenging setting: the approximation of optimal value functions for systems governed by nonlinear dynamics. As a representative example, we consider the

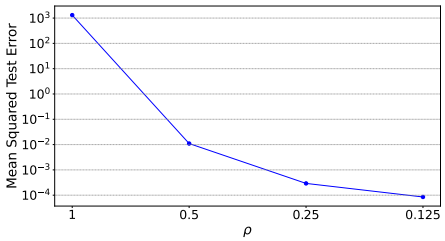

Figure 11: Test error (log-scale) for approximating V (exponential decay) in dependence of ρ.

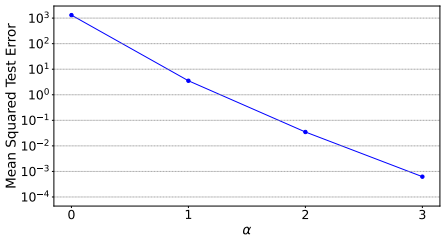

Figure 12: Test error (log-scale) for approximating $\widetilde{V}$ (polynomial decay) in dependence of α.

infinite-horizon optimal control problem associated with the Allen–Cahn equation, a nonlinear parabolic PDE of reaction-diffusion type, given by

$$\begin{cases} \partial_t y(t,x) = \sigma\, \partial_{xx} y(t,x) + y(t,x)(1 - y(t,x)^2) + u(t,x), \\ \quad y(0,x) = y_0(x), \end{cases}$$

on the spatial domain $x \in [0,1]$, equipped with homogeneous Neumann boundary conditions. The goal is to regulate the system toward the unstable steady state $\tilde{y}(x) \equiv 0$ by minimizing the quadratic cost

$$\tilde{J}(y_0, u) = \int_0^\infty \int_0^1 \left(\delta_1 |y(t,x)|^2 + \delta_2 |u(t,x)|^2\right) dx\, dt, \quad \delta_1, \delta_2 > 0.$$

The optimal value function for this nonlinear problem is not available in closed form. In order to be able to assess the performance of the separable neural network architecture for this example, we thus learn an approximation of the optimal value function to which we can then compare the function represented by the network. For obtaining this approximation, we discretize the state in space using finite differences with $n = 50$ grid points, yielding a finite-dimensional nonlinear optimal control problem of the form

$$\begin{aligned} \min_{u \in \mathcal{U}_c} J(u, y_0) = \int_0^\infty & y(t)^\top Q y(t) + u(t)^\top R u(t)\, dt, \\ & \dot{y}(t) = A(y(t)) y(t) + B u(t), \end{aligned} \tag{6.2}$$

where $y(t) \in \mathbb{R}^n$, $A(y) \in \mathbb{R}^{n \times n}$ depends nonlinearly on the state, $B = I_n$, $Q = \frac{\delta_1}{n} I_n$, and $R = \frac{\delta_2}{n} I_n$. We approximate the optimal value function V of (6.2) by a quadratic state-dependent ansatz: We set $V(y) = y^\top P(y) y$, where $P(y)$ is obtained as the solution to the State-Dependent Riccati Equation (SDRE):

$$A^\top(y) P(y) + P(y) A(y) - P(y) B R^{-1} B^\top P(y) + Q = 0. \tag{6.3}$$

This construction does not solve the associated Hamilton–Jacobi–Bellman equation exactly, but provides a pointwise approximation of the value function by enforcing local optimality at each state y. The SDRE approach constitutes a generalization of the classical LQR scheme, designed to incorporate the state-dependent nonlinearities present in the system dynamics. or a detailed discussion of the theoretical underpinnings, we refer to [8] and the references therein, while applications in the context

of optimal control for PDEs can be found in [1, 2]. Note that while $A(y) \in \mathbb{R}^{n\times n}$ depends on the state in this non-linear setting, its non-zero entries are independent of y. Thus, we can define the graph corresponding to V with subsystems $z_j = y_j$, $1 \leq j \leq n$, analogously to the linear quadratic setting described in Section 4 leading to a problem formulation fitting Setting 2.1 with $\mathrm{dist}_{\mathcal{G}}(i, j) = |i - j|$.

We now examine the emergence of a decaying sensitivity property in this nonlinear 50-dimensional setting. By Definition 2.2 this requires to investigate the Lipschitz properties of the functions g_j defined in (2.1). To this end, we fix the parameters $\sigma = 10^{-2}$, $\delta_1 = 10$, $\delta_2 = 0.1$, and consider a fixed reference point $x \in \mathbb{R}^n$. For each index $i \in \{1, \dots, n\}$, we define a perturbed vector $\tilde{x}_i \in \mathbb{R}^n$ by

$$\tilde{x}_{i,k} = \begin{cases} 0, & \text{if } k = i, \\ x_k, & \text{otherwise.} \end{cases}$$

A numerical estimate of the local Lipschitz constant of g_j with respect to the ith coordinate is then given by

$$\tilde{L}_{i,j} = \frac{|g_j(x) - g_j(\tilde{x}_i)|}{\|x - \tilde{x}_i\|}.$$

The right panels of Figure 13 report the empirical estimates $\tilde{L}_{i,j}$ for $j = 1$ (top) and $j = 25$ (bottom). Each estimate is computed at two reference points obtained by setting the j-th component to α and all other components to 0.1, with $\alpha \in \{0.1, 0.01\}$. Results are shown on a logarithmic scale. The observed exponential decay of $\tilde{L}_{i,j}$ with increasing $|i - j|$, as well as its decrease for smaller $\|z_j\|$, provide strong numerical evidence that V satisfies the decaying sensitivity property as defined in Definition 2.2.

This structural property justifies the use of the separable neural network architecture presented in Section 5 to leverage localized interactions to yield an accurate and efficient approximation of V. We employ a S-NN architecture configured with a graph-neighborhood radius of $l = 3$ and $M = 32$ neurons in each subnetwork, resulting in a total of 9409 trainable parameters. The dataset comprises 1024 samples for training, validation, and testing uniformly randomly chosen in $\Omega = \{x \in \mathbb{R}^{50} \mid \|x\|_2 \leq 1\}$, respectively. As regularization weights we chose $\nu_g = 10^4$ and $\nu_z = 10$. The large value of ν_g discourages flat regions in the surrogate, while the stringent tolerance guarantees a close fit to the reference solution; a systematic study of these effects is left for future work.

The left and middle panels in Figure 13 shows the results of a training process conducted with a stopping tolerance of $\epsilon = 10^{-7}$. Note that while training was carried out on Ω, i.e., the unit ball in $\mathbb{R}^{50}$, the plots are generated for visualization purposes on two-dimensional slices of the hypercube $[-1, 1]^{50}$. We depict two representative coordinate planes, namely (x_1, x_2) (top) and (x_{25}, x_{50}) (bottom). The left panel depicts the true solution derived from the SDRE ansatz, while the middle panel shows the corresponding approximation generated by the S-NN. The visual agreement between the reference and surrogate value functions is strong, with the SNN model achieving a mean squared error in the test data of approximately 8.32×10^{-8} after 210 training epochs.

7. Conclusion. This work has presented a comprehensive framework for constructing separable approximations of high-dimensional functions, emphasizing their representation through neural networks. The key contribution lies in leveraging the

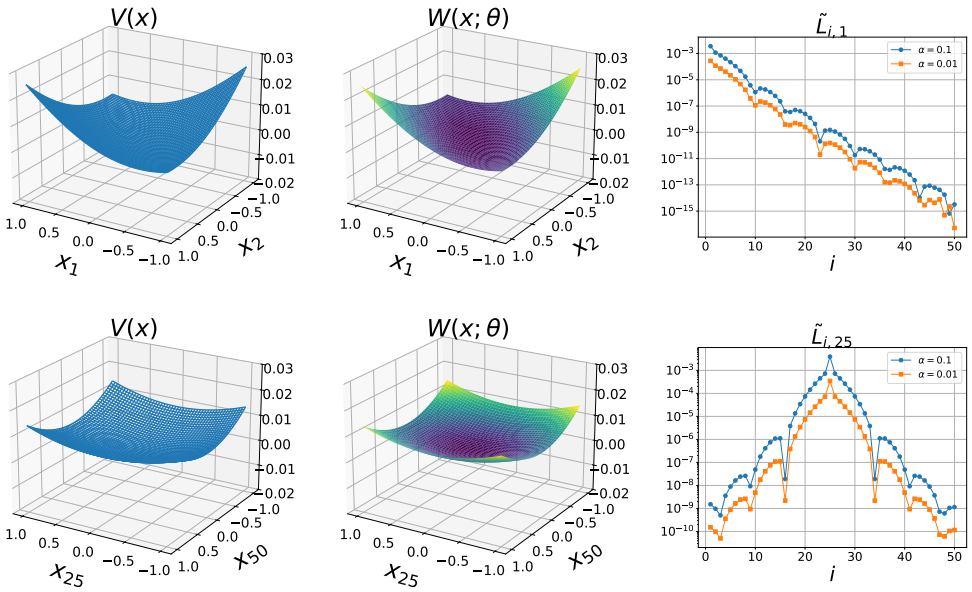

Figure 13: Plot of V projected onto the (x_1, x_2)-axis (upper left) and onto the (x_{25}, x_{50})-axis (lower left), the S-NN approximation onto the same hyperplanes (middle column), and the decay of $\tilde{L}_{i,j}$ for $j = 1$ (upper right) and $j = 25$ (lower right) varying the index i.

property of decaying sensitivity, which facilitates the decomposition of complex functions into localized components with reduced computational complexity. We demonstrated that separable approximations are not only theoretically viable but also practical in the context of optimal control problems, particularly in the LQR setting. To extend this approach to a nonlinear setting, datasets for the optimal control problem can be systematically generated utilizing either Pontryagin's Maximum Principle [3] or the State-Dependent Riccati Equation [1, 10]. Moreover, the integration of separable structures with neural networks underscores the potential for addressing the curse of dimensionality in a wide range of applications. The resulting architectures allow for efficient representation and computation of value functions.

Future research directions include extending the existence results on decaying sensitivity and the proposed numerical methods to multi-agent control problems, exploring adaptive strategies for graph-based sensitivity modeling, and further optimizing neural network architectures to enhance generalization and computational efficiency.

Acknowledgment. We thank the anonymous reviewers for their valuable and constructive comments for improving the paper.

REFERENCES

[1] G. Albi, S. Bicego, and D. Kalise, *Gradient-augmented supervised learning of optimal feedback laws using state-dependent Riccati equations*, IEEE Control Syst. Lett., 6 (2022), pp. 836–841.

[2] A. Alla, D. Kalise, and V. Simoncini, *State-dependent Riccati equation feedback stabilization for nonlinear PDEs*, Adv. Comput. Math., 49 (2023).

[3] B. Azmi, D. Kalise, and K. Kunisch, *Optimal feedback law recovery by gradient-augmented sparse polynomial regression*, J. Mach. Learn. Res, 22 (2021), pp. 1–32.
[4] M. Benzi, *Localization in matrix computations: Theory and applications*, in Exploiting Hidden Structure in Matrix Computations: Algorithms and Applications : Cetraro, Italy 2015, M. Benzi and V. Simoncini, eds., Springer, 2016, pp. 211–317.
[5] M. Benzi and G. H. Golub, *Bounds for the entries of matrix functions with applications to preconditioning*, BIT Numerical Mathematics, 39 (1999), pp. 417–438.
[6] G. Beylkin and M. J. Mohlenkamp, *Numerical operator calculus in higher dimensions*, Proc. Natl. Acad. Sci. U.S.A., 99 (2002), pp. 10246–10251.
[7] T. Breiten and K. Kunisch, *Neural network based nonlinear observers*, Systems Control Lett., 148 (2021).
[8] T. Çimen, *State-dependent Riccati equation (SDRE) control: a survey*, IFAC Proceedings Volumes, 41 (2008), pp. 3761–3775.
[9] S. Dolgov, D. Kalise, and K. K. Kunisch, *Tensor Decomposition Methods for High-dimensional Hamilton–Jacobi–Bellman Equations*, SIAM J. Sci. Comput., 43 (2021), pp. A1625–A1650.
[10] S. Dolgov, D. Kalise, and L. Saluzzi, *Data-driven tensor train gradient cross approximation for Hamilton–Jacobi–Bellman equations*, SIAM J. Sci. Comput., 45 (2023), pp. A2153–A2184.
[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*, JHU press, 2013.
[12] L. Grüne, *Computing Lyapunov functions using deep neural networks*, J. Comput. Dyn., 8 (2021), pp. 131–152.
[13] L. Grüne, M. Schaller, and A. Schiela, *Exponential sensitivity and turnpike analysis for linear quadratic optimal control of general evolution equations*, J. Differential Equations, 268 (2020), pp. 7311–7341.
[14] Göttlich, Simone, Schaller, Manuel, and Worthmann, Karl, *Perturbations in PDE-constrained optimal control decay exponentially in space*, ESAIM Control Optim. Calc. Var., 31 (2025).
[15] A. Haber and M. Verhaegen, *Sparse solution of the Lyapunov equation for large-scale interconnected systems*, Automatica, 73 (2016), pp. 256–268.
[16] N. J. Higham, *Functions of matrices: theory and computation*, SIAM, 2008.
[17] W. Kang and Q. Gong, *Feedforward neural networks and compositional functions with applications to dynamical systems*, SIAM J. Control Optim., 60 (2022), pp. 786–813.
[18] K. Kunisch and D. Walter, *Semiglobal optimal feedback stabilization of autonomous systems via deep neural network approximation*, ESAIM Control Optim. Calc. Var., 27 (2021), p. 16.
[19] K. Kunisch and D. Walter, *Optimal feedback control of dynamical systems via value-function approximation*, Comptes Rendus. Mécanique, 351 (2023), pp. 535–571.
[20] F. Kuo, I. Sloan, G. Wasilkowski, and H. Woźniakowski, *On decompositions of multivariate functions*, Math. Comp., 79 (2010), pp. 953–966.
[21] P. Lancaster and L. Rodman, *Algebraic Riccati Equations*, Clarendon press, 1995.
[22] S. Massei and L. Saluzzi, *On the data-sparsity of the solution of Riccati equations with applications to feedback control*, arXiv preprint arXiv:2408.16569, (2024).
[23] H. Mhaskar, *Neural networks for optimal approximation of smooth and analytic functions*, Neural Comput., 8 (1996), pp. 164–177.
[24] S. Na and M. Anitescu, *Exponential decay in the sensitivity analysis of nonlinear dynamic programming*, SIAM J. Optim., 30 (2020), pp. 1527–1554.
[25] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.
[26] A. B. Owen, *The dimension distribution and quadrature test functions*, Statist. Sinica, 13 (2003), pp. 1–17.
[27] G. Qu, A. Wierman, and N. Li, *Scalable reinforcement learning for multiagent networked systems*, Oper. Res., 70 (2022), p. 3601–3628.
[28] C. Rieger and H. Wendland, *On the approximability and curse of dimensionality of certain classes of high-dimensional functions*, SIAM J. Numer. Anal., 62 (2024), pp. 842–871.
[29] S. Shin, M. Anitescu, and V. M. Zavala, *Exponential decay of sensitivity in graph-structured nonlinear programs*, SIAM J. Optim., 32 (2022), pp. 1156–1183.
[30] S. Shin, Y. Lin, G. Qu, A. Wierman, and M. Anitescu, *Near-optimal distributed linear-quadratic regulator for networked systems*, SIAM J. Control Optim., 61 (2023), pp. 1113–1135.
[31] S. Shin and V. M. Zavala, *Controllability and observability imply exponential decay of sensitivity in dynamic optimization*, IFAC-PapersOnLine, 54 (2021), pp. 179–184.
[32] M. Sperl, J. Mysliwitz, and L. Grüne, *On the existence and neural network representation*

of separable control Lyapunov functions, Automatica, 182 (2025), p. 112517.

[33] M. Sperl, L. Saluzzi, L. Grüne, and D. Kalise, *Separable approximations of optimal value functions under a decaying sensitivity assumption*, in 2023 62nd IEEE Conference on Decision and Control (CDC), IEEE, 2023, pp. 259–264.

[34] R. C. Zhang, W. Li, and N. Li, *On the optimal control of network LQR with spatially-exponential decaying structure*, in 2023 American Control Conference (ACC), IEEE, 2023, pp. 1775–1780.