Coded Computing for Resilient Distributed Computing: A Learning-Theoretic Framework

Parsa Moradi
University of Minnesota
moradi@umn.edu

Behrooz Tahmasebi
MIT CSAIL
bzt@mit.edu

Mohammad Ali Maddah-Ali
University of Minnesota
maddah@umn.edu

Abstract

Coded computing has emerged as a promising framework for tackling significant challenges in large-scale distributed computing, including the presence of slow, faulty, or compromised servers. In this approach, each worker node processes a combination of the data, rather than the raw data itself. The final result then is decoded from the collective outputs of the worker nodes. However, there is a significant gap between current coded computing approaches and the broader landscape of general distributed computing, particularly when it comes to machine learning workloads. To bridge this gap, we propose a novel foundation for coded computing, integrating the principles of learning theory, and developing a framework that seamlessly adapts with machine learning applications. In this framework, the objective is to find the encoder and decoder functions that minimize the loss function, defined as the mean squared error between the estimated and true values. Facilitating the search for the optimum decoding and functions, we show that the loss function can be upper-bounded by the summation of two terms: the generalization error of the decoding function and the training error of the encoding function. Focusing on the second-order Sobolev space, we then derive the optimal encoder and decoder. We show that in the proposed solution, the mean squared error of the estimation decays with the rate of $\mathcal{O}(S^3N^{-3})$ and $\mathcal{O}(S^{8/5}N^{-3/5})$ in noiseless and noisy computation settings, respectively, where N is the number of worker nodes with at most S slow servers (stragglers). Finally, we evaluate the proposed scheme on inference tasks for various machine learning models and demonstrate that the proposed framework outperforms the state-of-the-art in terms of accuracy and rate of convergence.

1 Introduction

The theory of *coded computing* has been developed to improve the reliability and security of large-scale machine learning platforms, effectively tackling two major challenges: (1) the detrimental impact of *slow workers (stragglers)* on overall computation efficiency, and (2) the threat of *faulty or malicious workers* that can compromise data accuracy and integrity. These challenges have been well-documented in the literature, including the seminal work [1] from Google. For instance, [2] reported that in a sample set of 3000 matrix multiplication jobs on AWS Lambda, while the median job time was 40 seconds, approximately 5% of worker nodes took 100 seconds to respond, and two nodes took as long as 375 seconds. Furthermore, coded computing has also been instrumental in addressing *privacy concerns*, a crucial aspect of distributed computing systems [3–12].

The concept of coded computing has been motivated by the success of coding in communication over unreliable channels, where instead of transmitting raw data, the transmitter sends a (linear) combination of the data, known as coded data. This redundancy in the coded data enables the

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3554

receiver to recover the raw data even in the presence of errors or missing values. Similarly, coded computing includes three layers [3, 11, 13] (see Figure 1(a)):

(1) *The Encoding Layer* in which a master node sends a (linear) combination of data, as coded data, to each worker node.

(2) *The Computing Layer*, in which the worker nodes apply a predefined computation to their assigned coded data and send the results back to the master node.

(3) *The Decoding Layer*, in which the master node recovers the final results from the computation results over coded data. In this layer, the decoder leverages the coded redundancy in the computation to recover the missing results of the stragglers and detect and correct the adversarial outputs.

The existing coded computing has largely built upon algebraic coding theory, drawing inspiration from the renowned Reed-Solomon code construction in communication [14], with proven straggler and Byzantine resiliency [15]. However, the coding in communication is designed for the exact recovery of the messages, built on a foundation that is inconsistent with the computational requirements of machine learning. Developing a code that preserves its specific construction while composing with computation is extremely challenging, leading to significant restrictions. Firstly, current methods are mainly restricted to specific computation functions, such as polynomials and matrix multiplication [3, 11, 13, 16, 17]. Secondly, rooted in algebraic error correction codes, existing approaches are tailored for finite field computations, leading to numerical instability when dealing with real-valued data [18, 19]. Furthermore, these methods are unsuitable for approximate, fixed-point, or floating-point computing, where exact computation is neither possible nor necessary, such as in machine learning inference or training tasks. Finally, these schemes typically have a recovery threshold, which is the minimum number of samples required to recover results from coded outputs of worker nodes [3, 13]. If the number of workers falls below this threshold, the recovery process fails entirely.

Several works have attempted to mitigate the aforementioned issues and transform the coded computing scheme into a more robust and adaptable one, applicable to a wide range of computation functions. These efforts include approximating non-polynomial functions with polynomial ones [5, 20], refining the coding mechanism to enhance stability [21–25], and leveraging approximation computing techniques to reduce the recovery threshold and increase recovery flexibility [26–29]. However, these attempts fail to bridge the existing gap between coded computing and general distributed computing systems. The root cause of these issues lies in the fact that they are grounded in coding theory, based on a foundation that is not compatible with the requirements of large-scale machine learning. Therefore, this paper aims to address the following objective:

Objective: The main objective of this paper is to develop a new foundation for coded computing, not solely based on *coding theory*, but also grounded in *learning theory*, that seamlessly integrates with machine learning applications, offering a more natural and effective solution for *general computing*.

In this paper, we establish a learning-theoretic foundation for coded computing, applicable to general computations. We adopt an end-to-end system perspective, that integrates an end-to-end loss function, to find the optimum encoding and decoding functions, focusing on straggler resiliency. We show that the loss function is upper-bounded by the sum of two terms: one characterizing the *generalization error* of the decoder function and the other capturing the *training error* of the encoder function. Regularizing the decoder layer, we derive the optimal decoder in the Reproducing Kernel Hilbert space (RKHS) of second-order Sobolev functions. This provides an explicit solution for the optimum decoder function and allows us to characterize the resulting loss of the decoding layer. The decoder loss appears as a regularizing term in optimizing the encoding function and represents the norm in another RKHS. Thus, the optimum solution for the encoding function can be derived, too. We address two noise-free and noisy computation settings, for which we derive the *optimal encoder and decoder* and corresponding convergence rate. We prove that the proposed framework exhibits a faster convergence rate compared to the state-of-the-art and the numerical evaluations support the theoretical derivations (see Figure 1(b)).

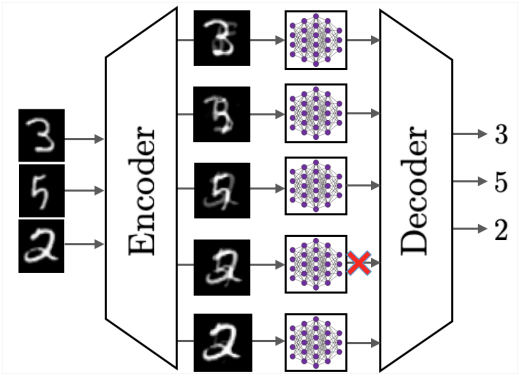


Figure 1(a): Coded Computing: Each worker node processes a combination of data (coded data). The decoder recovers the final results, even in the presence of missing outputs from some worker nodes.

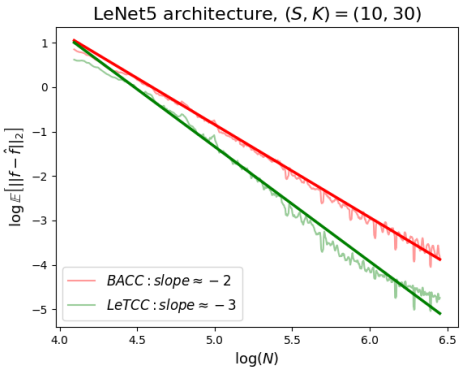


Figure 1(b): The log-log plot of the expected error versus the number of workers (N) for the proposed framework (LeTCC) and the state-of-the-art BACC [29]. LeTCC framework not only achieves a lower estimation error but also has a faster convergence rate.

Contributions: The main contributions of this paper are:

- We develop a new foundation for coded computing by integrating it with learning theory, rather than relying solely on coding theory. We define the loss function as the mean square error of the computation estimation, averaged over all possible sets of at most S stragglers (Section 3.1). To be able to find the best encoding and decoding functions, we bound the loss function with the summation of two terms, one characterizing the generalization error of the decoder function and the other capturing the training error of the encoder function (Section 3).
- Assuming that the encoder and decoder functions reside in the Hilbert space of second-order Sobolev functions, we use the theory of RKHSs to find the optimum encoding and decoding functions and characterize the convergence rate for the expected loss in both noise-free and noisy computation regimes (Section 4).
- We have extensively evaluated the proposed scheme across different data points and computing functions including state-of-the-art deep neural networks and demonstrated that our proposed framework considerably outperforms the state-of-the-art in terms of recovery accuracy (Section 5).

2 Preliminaries and Problem Definition

2.1 Notations

Throughout this paper, uppercase and lowercase bold letters denote matrices and vectors, respectively. Coded vectors and matrices are indicated by a $\sim$ sign, as in $\tilde{\mathbf{x}}, \tilde{\mathbf{A}}$. The set $\{1, 2, \ldots, n\}$ is denoted as $[n]$ and symbol $|S|$ denotes the cardinality of the set S. Finally, we represent first, second and k-th order derivative of function f as f', f'', and $f^{(k)}$, respectively.

2.2 Problem Setting

Consider a master node and a set of N workers. The master node is tasked with computing $\{\mathbf{f}(x_k)\}_{k=1}^{K}$ using a cluster of N worker nodes, given a set of K data points $\{\mathbf{x}_k\}_{k=1}^{K}, \mathbf{x}_k \in \mathbb{R}^d$. Here, $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^m$ represents an arbitrary function, which could be a simple one-dimensional function or a complex deep neural network, and K, d, m are integers. A naive approach would be to assign the computation of $\mathbf{f}(\mathbf{x}_k)$ to one worker node for $k \in [K]$. However, some worker nodes may act as stragglers, failing to complete their tasks within the required deadline. To mitigate this issue, the master node employs coding and sends N coded data points to each worker node using an encoder function. Each coded data point is a combination of raw data points. Subsequently, each

https://doi.org/10.52202/079017-3554

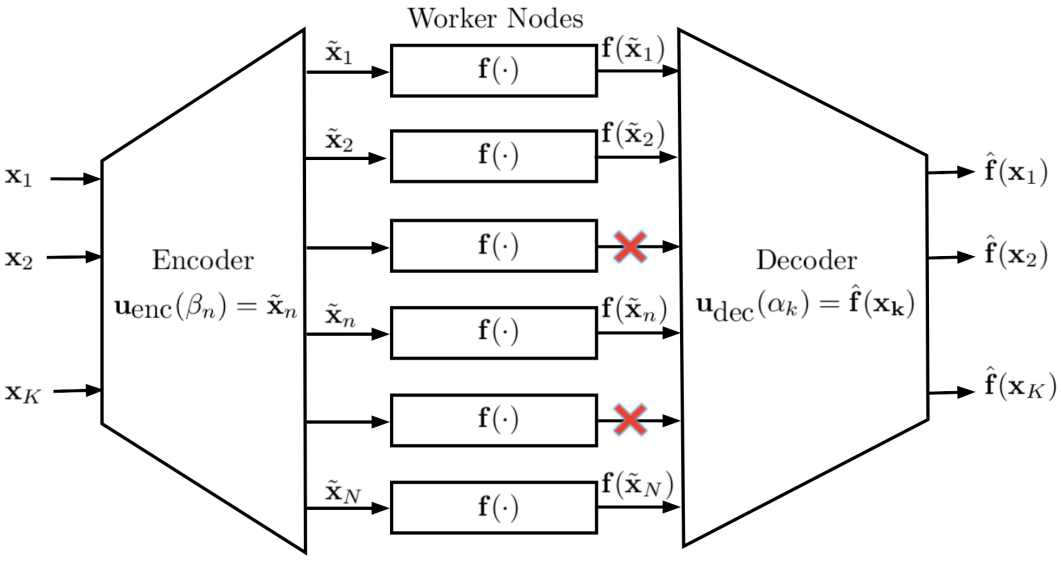


Figure 2: LeTCC framework.

worker applies the function $\mathbf{f}(\cdot)$ to the received coded data and sends the result, coded results, back to the master node. The master node's goal is to approximately recover $\hat{\mathbf{f}}(\mathbf{x}_k) \approx \mathbf{f}(\mathbf{x}_k)$ using a decoder function, even if some worker nodes appear to be stragglers. The redundancy in the coded data and corresponding coded results enables the master node to recover the desirable results, $\{\mathbf{f}(\mathbf{x}_k)\}_{k=1}^K$.

3 Proposed Framework: LeTCC

Here, we propose a novel straggler-resistant Learning-Theoretic Coded Computing (LeTCC) framework for general distributed computing. As depicted in Figure 2, our framework comprises two encoding and decoding layers, with a computing layer sandwiched between them. The framework operates according to the following steps:

(1) **Encoding Layer:** The master node fits an encoder regression function $\mathbf{u}_{\text{enc}} : \mathbb{R} \to \mathbb{R}^d$ at points $\{(\alpha_k, \mathbf{x}_k)\}_{k=1}^K$ for fixed, distinct, and ordered values $\alpha_1 < \alpha_2 < \cdots < \alpha_K \in \mathbb{R}$. Then, it computes the encoder function $\mathbf{u}_{\text{enc}}(\cdot)$ on another set of fixed, distinct, and ordered values $\{\beta_n\}_{n=1}^N$ where $\beta_1 < \beta_2 < \cdots < \beta_N \in \mathbb{R}$, with $k \in [K]$ and $n \in [N]$. Subsequently, the master node sends the coded data points $\tilde{\mathbf{x}}_n = \mathbf{u}_{\text{enc}}(\beta_n) \in \mathbb{R}^d$ to worker n for $n \in [N]$. Note that each coded data point $\tilde{\mathbf{x}}_n$ is a combination of all initial points $\{\mathbf{x}_k\}_{k=1}^K$.

(2) **Computing Layer:** Each worker node $n \in [N]$ computes $\mathbf{f}(\tilde{\mathbf{x}}_n) = \mathbf{f}(\mathbf{u}_{\text{enc}}(\beta_n))$ on its assigned input and sends the result back to the master node.

(3) **Decoding Layer:** The master node receives the results $\mathbf{f}(\tilde{\mathbf{x}}_v)_{v\in\mathcal{F}}$ from the non-straggler worker nodes in the set $\mathcal{F}$. Next, it fits a decoder regression function $\mathbf{u}_{\text{dec}} : \mathbb{R} \to \mathbb{R}^m$ at points $(\beta_v, \mathbf{f}(\tilde{\mathbf{x}}_v))_{v\in\mathcal{F}} = (\beta_v, \mathbf{f}(\mathbf{u}_{\text{enc}}(\beta_v)))_{v\in\mathcal{F}}$. Finally, using the function $\mathbf{u}_{\text{dec}}(\cdot)$, the master node computes $\hat{\mathbf{f}}(\mathbf{x}_k) := \mathbf{u}_{\text{dec}}(\alpha_k)$ as an approximation of $\mathbf{f}(\mathbf{x}_k)$ for $k \in [K]$. Recall that $\mathbf{u}_{\text{dec}}(\alpha_k) \approx \mathbf{f}(\mathbf{u}_{\text{enc}}(\alpha_k)) \approx \mathbf{f}(\mathbf{x}_k)$.

As mentioned above, the master node selects and fixes the regression points, $\{\alpha_k\}_{k=1}^K$ and $\{\beta_n\}_{n=1}^N$, which remain constant throughout the entire process. The encoder and decoder functions are the only components subject to optimization.

Note that the computational efficiency of the encoding and decoding layers is crucial. This includes the fitting process of the encoder and decoder regression functions, as well as the computation of these regression functions at points $\{\beta_v\}_{v\in\mathcal{F}}$ and $\{\alpha_k\}_{k=1}^K$. If the master node's computation time is not substantially decreased compared to computing $\{\mathbf{f}(\mathbf{x}_k)\}_{k=1}^K$ by itself, then adopting this framework would not provide any benefits for the master node.

3.1 Objective

We view the whole scheme as a unified predictive framework that provides an approximate estimation of the values $\{\mathbf{f}(\mathbf{x}_k)\}_{k=1}^K$. We denote the estimator function of the LeTCC scheme as $\hat{\mathbf{f}}_{\boldsymbol{\alpha},\boldsymbol{\beta}}[\mathbf{u}_{\text{enc}}, \mathbf{u}_{\text{dec}}, \mathcal{F}](\cdot)$, where $\boldsymbol{\alpha} := [\alpha_1, \ldots, \alpha_K]^T$, $\boldsymbol{\beta} := [\beta_1, \ldots, \beta_N]^T$, and $\mathcal{F} := \{i_1, \ldots, i_{|\mathcal{F}|}\}$ represents the set of non-straggler worker nodes.

Let us define a random variable $F_{S,N}$ distributed over the set of all subsets of N workers with maximum S stragglers, $\{\mathcal{F} : \mathcal{F} \subseteq [N], |\mathcal{F}| \geqslant N - S\}$. Also, suppose each worker node $n \in [N]$ computes the function $\mathbf{f}_n(x) = \mathbf{f}(x) + \boldsymbol{\epsilon}_n$, where $\boldsymbol{\epsilon}_n$, $n \in [N]$ are independent zero-mean noise vectors with covariance $\sigma^2\mathbf{I}$.

This enables us to define the following loss function, which evaluates the framework's performance:

$$\mathcal{R}(\hat{\mathbf{f}}) := \mathop{\mathbb{E}}_{\boldsymbol{\epsilon},\mathcal{F}\sim F_{S,N}} \left[\frac{1}{K} \sum_{k=1}^{K} \left\| \hat{\mathbf{f}}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_k) \right\|_2^2 \right] = \mathop{\mathbb{E}}_{\boldsymbol{\epsilon},\mathcal{F}\sim F_{S,N}} \left[\frac{1}{K} \sum_{k=1}^{K} \|\mathbf{u}_{\text{dec}}(\alpha_k) - \mathbf{f}(\mathbf{x}_k)\|_2^2 \right], \quad (1)$$

where $\hat{\mathbf{f}}(\mathbf{x}) := \hat{\mathbf{f}}_{\boldsymbol{\alpha},\boldsymbol{\beta}}[\mathbf{u}_{\text{enc}}, \mathbf{u}_{\text{dec}}, \mathcal{F}](\mathbf{x})$ to simplify the notation, $\|\cdot\|_2$ represents the ℓ_2-norm, and $\boldsymbol{\epsilon} = [\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_N]^T$. Our objective is to find $\mathbf{u}_{\text{enc}}(.)$ and $\mathbf{u}_{\text{dec}}(.)$ that minimize the objective function (1), which is very challenging, given that $\hat{\mathbf{f}}(.)$ is a composition of $\mathbf{u}_{\text{enc}}(.)$ and $\mathbf{u}_{\text{dec}}(.)$ and the computation in the middle. Here, we take an important step to decompose these two, to gain a deeper understanding of interactions. Adding and subtracting $\mathbf{f}(\mathbf{u}_{\text{enc}}(\alpha_k))$ and utilizing inequality of arithmetic and geometric means (AM-GM), one can obtain an upper bound for (1):

$$\begin{aligned}
\mathcal{R}(\hat{\mathbf{f}}) &= \mathop{\mathbb{E}}_{\boldsymbol{\epsilon},\mathcal{F}\sim F_{S,N}} \left[\frac{1}{K} \sum_{k=1}^{K} \|(\mathbf{u}_{\text{dec}}(\alpha_k) - \mathbf{f}(\mathbf{u}_{\text{enc}}(\alpha_k))) + (\mathbf{f}(\mathbf{u}_{\text{enc}}(\alpha_k)) - \mathbf{f}(\mathbf{x}_k))\|_2^2 \right] \\
&\leqslant \underbrace{\mathop{\mathbb{E}}_{\boldsymbol{\epsilon},\mathcal{F}\sim F_{S,N}} \left[\frac{2}{K} \sum_{k=1}^{K} \|\mathbf{u}_{\text{dec}}(\alpha_k) - \mathbf{f}(\mathbf{u}_{\text{enc}}(\alpha_k))\|_2^2 \right]}_{\mathcal{L}_{\text{dec}}(\hat{\mathbf{f}})} + \underbrace{\frac{2}{K} \sum_{k=1}^{K} \|\mathbf{f}(\mathbf{u}_{\text{enc}}(\alpha_k)) - \mathbf{f}(\mathbf{x}_k)\|_2^2}_{\mathcal{L}_{\text{enc}}(\hat{\mathbf{f}})}. \quad (2)
\end{aligned}$$

The right-hand side of (2) comprises two terms, which uncover an interesting interplay between the encoder and decoder regression functions. Let us elaborate on what each term corresponds to.

- $\mathcal{L}_{\text{dec}}(\hat{\mathbf{f}})$ – **The expected generalization error of the decoder regression:** Recall that the master node fits a decoder regression function, $\mathbf{u}_{\text{dec}}(\cdot)$, at a set of points denoted as $\{(\beta_v, \mathbf{f}(\mathbf{u}_{\text{enc}}(\beta_v)))\}_{v\in\mathcal{F}}$. $\mathcal{L}_{\text{dec}}$ represents the ℓ_2-norm of the decoder regression function's error on a distinct set of points $\{\alpha_k\}_{k=1}^K$, which are *different* from its training data $\{\beta_v\}_{v\in\mathcal{F}}$. Consequently, this term provides an unbiased estimate of the decoder's generalization error. Given that the decoder regression function develops to estimate $\mathbf{f}(\mathbf{u}_{\text{enc}}(\cdot))$, the generalization error of the decoder regression is inherently tied to the properties of $\mathbf{f}(\mathbf{u}_{\text{enc}}(\cdot))$. This, in turn, is influenced by characteristics of both the $\mathbf{f}(\cdot)$ and $\mathbf{u}_{\text{enc}}(\cdot)$ functions, making the $\mathcal{L}_{\text{dec}}(\hat{\mathbf{f}})$ a complex interplay of these two functions.
- $\mathcal{L}_{\text{enc}}(\hat{\mathbf{f}})$ – **A proxy to the training error of the encoder regression:** Remember that the encoder regression is fitted at points $\{(\alpha_k, \mathbf{x}_k)\}_{k=1}^K$. Consequently, the training error is calculated as $\frac{1}{K}\sum_{k=1}^K \|\mathbf{u}_{\text{enc}}(\alpha_k) - \mathbf{x}_k\|_2^2$. Therefore, $\mathcal{L}_{\text{enc}}$ represents the encoder training error magnified by the effect of computing function $\mathbf{f}(\cdot)$. Specifically, if $\mathbf{f}(\cdot)$ is q-Lipschitz, then $\mathcal{L}_{\text{enc}}(\hat{\mathbf{f}})$ can be upper bounded by:

$$\frac{2}{K} \sum_{k=1}^{K} \|\mathbf{f}(\mathbf{u}_{\text{enc}}(\alpha_k)) - \mathbf{f}(\mathbf{x}_k)\|_2^2 \leqslant \frac{2q^2}{K} \sum_{k=1}^{K} \|\mathbf{u}_{\text{enc}}(\alpha_k) - \mathbf{x}_k\|_2^2. \quad (3)$$

4 Main Results

In this section, we examine the proposed framework from a theoretical standpoint. We provide a comprehensive explanation of the design process for the decoder and encoder functions and subsequently analyze the convergence rate. For simplicity, we present the results for a one-dimensional

https://doi.org/10.52202/079017-3554

function $f : \mathbb{R} \to \mathbb{R}$. These results are generalizable to the case where $f : \mathbb{R} \to \mathbb{R}^m$, as discussed in Appendix F.

Suppose the regression points, $\{\alpha_k\}_{k=1}^K, \{\beta_n\}_{n=1}^N$, are confined to the interval $\Omega := (-1, 1)$ and $u_{\text{enc}}, u_{\text{dec}} \in \widetilde{\mathcal{H}}^2(\Omega; \mathbb{R})$, where $\widetilde{\mathcal{H}}^2(\Omega; \mathbb{R})$ is the reproducing kernel Hilbert space (RKHS) of second-order Sobolev functions on the interval Ω induced with the norm $\|f\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} := \int_\Omega (f''(t))^2\, dt + f(-1)^2 + f'(-1)^2$ which is an equivalent norm on Sobolev space introduced by [30] (see (28) in Appendix A.1). The definition and properties of Sobolev spaces, along with their reproducing kernels and norms, are reviewed in Appendix A.1.

Decoder Design: Since $\mathcal{L}_{\text{dec}}(\hat{\mathbf{f}})$ in the decomposition (2) characterizes the generalization error of the decoder function, we propose a regularized objective function for the decoder:

$$u^\star_{\text{dec}} = \underset{u \in \widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}{\operatorname{argmin}} \frac{1}{|\mathcal{F}|} \sum_{v \in \mathcal{F}} \left(u\left(\beta_v\right) - f\left(u_{\text{enc}}\left(\beta_v\right)\right)\right)^2 + \lambda_{\text{d}} \int_\Omega \left(u''(t)\right)^2\, dt. \tag{4}$$

The first term in (4) corresponds to the mean squared error, while the second term characterizes the smoothness of the decoder function on the interval Ω. Equation (4) represents a Kernel Ridge Regression problem (KRR). It can be shown that the solution of (4) has the following form [31, 32]:

$$d_0 + d_1 t + \sum_{v=1}^{|\mathcal{F}|} c_v \phi_0(t, \beta_{i_v}), \tag{5}$$

where $d_0, d_1 \in \mathbb{R}$, $\phi_0(\cdot, \cdot)$ is the kernel function of $\mathcal{H}_0^2(\Omega; \mathbb{R})$ (see Definition 2 and (44) in Appendix A.1), and $\mathbf{c} = [c_1, \ldots, c_{|\mathcal{F}|}]^T \in \mathbb{R}^{|\mathcal{F}|}$. Substituting (5) into the main objective (4), the coefficient vectors $\mathbf{c}$ and $\mathbf{d} := [d_0, d_1]^T$ can be efficiently computed by optimizing a quadratic equation [33]. This solution is known as the *second-order smoothing spline* function. The theoretical properties of smoothing splines are reviewed in Appendix A.2.

Let us define the following variables, which represent the maximum and minimum distances between consecutive data points in the decoder layer, $\{\beta_n\}_{n=1}^N$:

$$\Delta_{\max} := \max_{n \in \{0\} \cup [N]} \{\beta_{n+1} - \beta_n\}, \quad \Delta_{\min} := \min_{n \in [N-1]} \{\beta_{n+1} - \beta_n\}, \tag{6}$$

with $\beta_0 := -1$ and $\beta_{N+1} := 1$. The following theorems provide crucial insights for designing the encoder function as well as deriving the convergence rates.

Theorem 1 (Upper bound for noiseless computation, $\sigma_0 = 0$)**.** *Consider the* `LeTCC` *framework with N worker nodes and at most S stragglers with $\lambda_d \leqslant N^{-4}$. Assume $\{\alpha_k\}_{k=1}^K$ are arbitrary and distinct points in $\Omega = (-1, 1)$ and there is constant B such that $\frac{\Delta_{max}}{\Delta_{min}} \leqslant B$. If $f(\cdot)$ is a q-Lipschitz continuous function, then:*

$$\mathcal{R}(\hat{f}) \leqslant C_1 \left(\frac{S+1}{N}\right)^3 \cdot \|(f \circ u_{enc})''\|^2_{L^2(\Omega;\mathbb{R})} + \frac{2q^2}{K} \sum_{k=1}^K (u_{enc}(\alpha_k) - x_k)^2, \tag{7}$$

where C_1 is a constant.

The proof of Theorem 1 and the detailed expression for C_1 can be found in Appendix B.1.

Theorem 2 (Upper bound for noisy computation)**.** *Consider the* `LeTCC` *framework with N worker nodes and at most S stragglers and $\frac{1}{(N-S)^4} \leqslant \lambda_d \leqslant \lambda_0$ for constant $\lambda_0 \in \mathbb{R}$. Assume each worker node computes $f_n(x) = f(x) + \epsilon_n$ with $\mathbb{E}[\epsilon_n] = 0$ and $\mathbb{E}[\epsilon_n^2] \leqslant \sigma_0^2$. Assume $\{\alpha_k\}_{k=1}^K$ are arbitrary and distinct points in $\Omega = (-1, 1)$ and suppose there is constant B such that $\frac{\Delta_{max}}{\Delta_{min}} \leqslant B$. Assume $f(\cdot)$ is a q-Lipschitz continuous function. Then,*

$$\mathcal{R}(\hat{f}) \leqslant 4 \left(\frac{\sigma_0^2}{N-S}\right)^{\frac{3}{5}} \left(C_2 \cdot C(\lambda_0) \cdot p_4(S) \cdot \|(f \circ u_{enc})''\|^2_{L^2(\Omega;\mathbb{R})}\right)^{\frac{2}{5}} + \frac{2q^2}{K} \sum_{k=1}^K (u_{enc}(\alpha_k) - x_k)^2, \tag{8}$$

if

$$\frac{1}{(N-S)^{\frac{1}{5}}\lambda_0^{\frac{1}{4}}} \leqslant \left(\frac{C_2 \cdot C(\lambda_0) \cdot p_4(S) \cdot \left\|(f \circ u_{enc})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}}{\sigma_0^2}\right)^{\frac{1}{5}} \leqslant (N-S)^{\frac{4}{5}} \tag{9}$$

where C_2 is a constant, $C(\lambda_0) = \mathcal{O}(\lambda_0^{\frac{1}{2}})$ is an increasing function of λ_0, and $p_4(S)$ is a degree-4 polynomial in S with positive constant coefficients.

The proof and expressions for C_2 and $C(\lambda_0)$ are provided in Appendix B.2.

Encoder Design: The upper bounds established in Theorems 1, 2 hold for all $u_{\text{enc}} \in \widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})$. However, they do not directly lead to a design for $u_{\text{enc}}(\cdot)$. To address this, we present the following theorem which bounds the $\|f \circ u_{\text{enc}}\|^2_{L^2(\Omega;\mathbb{R})}$, enabling us to construct $u_{\text{enc}}(\cdot)$ without compromising the convergence rate.

Theorem 3. *Consider a* `LeTCC` *scheme. Assume computing function $f(\cdot)$ is q-Lipschitz continuous and $\|f''\|_{L^\infty(\Omega;\mathbb{R})} \leqslant \nu$. Then:*

$$\mathcal{R}(\hat{f}) \leqslant \frac{2q^2}{K}\sum_{k=1}^{K}(u_{enc}(\alpha_k) - x_k)^2 + \lambda_e \cdot \psi\big(\|u_{enc}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}\big), \tag{10}$$

for some monotonically increasing function $\psi : \mathbb{R}^+ \to \mathbb{R}^+$, where λ_e is depending on (N, S, σ_0, q, ν).

The proof can be found in Appendix B.3. By applying the representer theorem [34], we can deduce that the optimal encoder $u_{\text{enc}}(\cdot)$, which minimizes the right-hand side of (10) takes the form $u_{\text{enc}}(\cdot) = \sum_{k=1}^{K} z_k \phi(\alpha_k, \cdot)$, where $\mathbf{z} \in \mathbb{R}^K$, and ϕ is the kernel function of $\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})$, as discussed in Appendix A.2 and in (46). However, due to the non-linearity of $g(\cdot)$, calculating the values of the coefficients $\mathbf{z}$ is challenging. Nevertheless, we demonstrate that the coefficients can be efficiently derived under certain mild assumptions.

Proposition 1. *In the noiseless case, there exists $M \in \mathbb{R}$ that depends only on $\{\alpha_k\}_{k=1}^K$ and $\{x_k\}_{k=1}^K$, such that:*

(i) If $\|u_{enc}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} \leqslant M$, then:

$$\mathcal{R}(\hat{f}) \leqslant \widetilde{R}(u_{enc}), \tag{11}$$

where $\widetilde{R}(u)$ is defined as follows:

$$\widetilde{R}(u) := \frac{2q^2}{K}\sum_{k=1}^{K}(u(\alpha_k) - x_k)^2 + \lambda_e \cdot (m_1 + m_2 M)\left(M + \int_\Omega u''(t)^2\, dt\right), \tag{12}$$

and m_1, m_2 are constants.

(ii) If $u^*(\cdot)$ is the minimizer of (12), then $\|u^*\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} \leqslant M$.

See Appendix B.4 for the proof. Proposition 1 states that, under mild assumptions there exists a *smoothing spline* that minimizes the upper bound given in (12) without changing in convergence rate.

Convergence Rate: Using Theorems 1, 2, and 3, we can derive the convergence rate of the proposed scheme as stated in the following theorem.

Theorem 4 (Convergence rate)**.** *For* `LeTCC` *scheme with N worker nodes and a maximum of S stragglers, $\mathcal{R}(\hat{f}) \leqslant \mathcal{O}\left(S^{\frac{8}{5}}N^{-\frac{3}{5}}\right)$ for the noisy computation, and $\mathcal{R}(\hat{f}) \leqslant \mathcal{O}\left(S^3N^{-3}\right)$ for the noiseless setting.*

Refer to Appendix B.5 for the proof. Notably, the convergence rate yields from Theorem 4 surpasses the state-of-the-art Berrut coded computing approach upper bound [29] (Figure 1), both with respect to S and N (see Appendix C.1 for detailed comparison).

https://doi.org/10.52202/079017-3554

Table 1: Comparison of the proposed framework (LeTCC) and the state-of-the-art (BACC) in terms of the Root Mean Squared Error (RMSE) and the Relative Accuracy (RelAcc).

$(N, K, \lvert\mathcal{F}\rvert)$	LeNet5 $(100, 20, 60)$		RepVGG $(60, 20, 20)$		ViT $(20, 8, 3)$	
Method	RMSE	RelAcc	RMSE	RelAcc	RMSE	RelAcc
BACC	2.55 ± 0.43	0.92 ± 0.04	2.44 ± 0.38	0.83 ± 0.05	0.68 ± 0.13	0.90 ± 0.07
LeTCC	$\mathbf{2.18\pm 0.51}$	$\mathbf{0.94\pm 0.04}$	$\mathbf{2.04\pm 0.42}$	$\mathbf{0.87\pm 0.05}$	$\mathbf{0.62\pm 0.11}$	$\mathbf{0.94\pm 0.06}$

5 Experimental Results

In this section, we extensively evaluate the proposed scheme across various scenarios. Our assessments involve examining multiple deep neural networks as computing functions and exploring the impact of different numbers of stragglers on the scheme's efficiency. The experiments are run using PyTorch [35] in a single GPU machine. We evaluate the performance of the LeTCC scheme in three different model architectures:

- **Shallow model**: We choose LeNet5 [36] architecture as a known shallow network with approximately 6×10^4 parameters, trained on the MNIST [37].
- **Deep model with low-dimensional output**: In this scenario, we evaluate the proposed scheme when the function is a deep neural network trained on color images in CIFAR-10 [38] dataset. We use the recently introduced RepVGG [39] network with around 26 million parameters which was trained on CIFAR-10[1].
- **Deep model with high-dimensional output**: Finally, we demonstrate the performance of the LeTCC scheme in a scenario where the input and output of the computing function are high-dimensional, and the function is a relatively large neural network. We consider the Vision Transformer (ViT) [40] as one of the state-of-the-art base neural networks in computer vision for our prediction model, with more than 80 million parameters (in the base version). The network was trained and fine-tuned on the ImageNet-1K dataset [41][2].

We use the output of the last softmax layer of each model as the output.

Hyper-parameters: The entire encoding and decoding process is the same for different functions, as we adhere to a non-parametric approach. The sole hyper-parameters involved are the two smoothing parameters $(\lambda_{\text{enc}}, \lambda_{\text{dec}})$ which are determined using cross-validation and greed search over different values of the smoothing parameters.

Baseline: We compare LeTCC with the Berrut approximate coded computing (BACC) introduced by [29] as the state-of-the-art coded computing scheme for general computing. The BACC framework is used in [29] for training neural networks and in [42] for inference. Although Berrut coded computing [29] is the only existing coded computing scheme for general functions, we include a comparison of the proposed framework with the Lagrange coded computing scheme [3] for polynomial computation in Appendix D.

Interpolation Points: We choose Chebyshev points of the first and second kind, $\{\alpha_i\}_{k=1}^{K} = \cos\left(\frac{(2k-1)\pi}{2K}\right)$ and $\{\beta_n\}_{n=1}^{N} = \cos\left(\frac{(n-1)\pi}{N-1}\right)$, for fair comparison with [29].

Evaluation Metrics: We employ two evaluation metrics to assess the performance of the proposed framework: Relative Accuracy (RelAcc) and Root Mean Squared Error (RMSE). RelAcc is defined as the ratio of the base model's prediction accuracy to the accuracy of the estimated model on the initial data points. RMSE, on the other hand, is our main loss defined in (1) which measures the empirical average of the root mean square difference over multiple batches of and non-straggler set $\mathcal{F}$, providing an unbiased estimation of expected mean square error, $\mathbb{E}_{\mathbf{x}\sim\mathcal{X},\mathcal{F}}\left[\frac{1}{K}\sum_{k=1}^{K}\|\mathbf{f}(\mathbf{x}_k) - \hat{\mathbf{f}}(\mathbf{x}_k)\|_2\right]$, for data distribution $\mathcal{X}$.

[1]The pre-trained weights can be found here.

[2]We use the official PyTorch pre-trained ViT network from here.

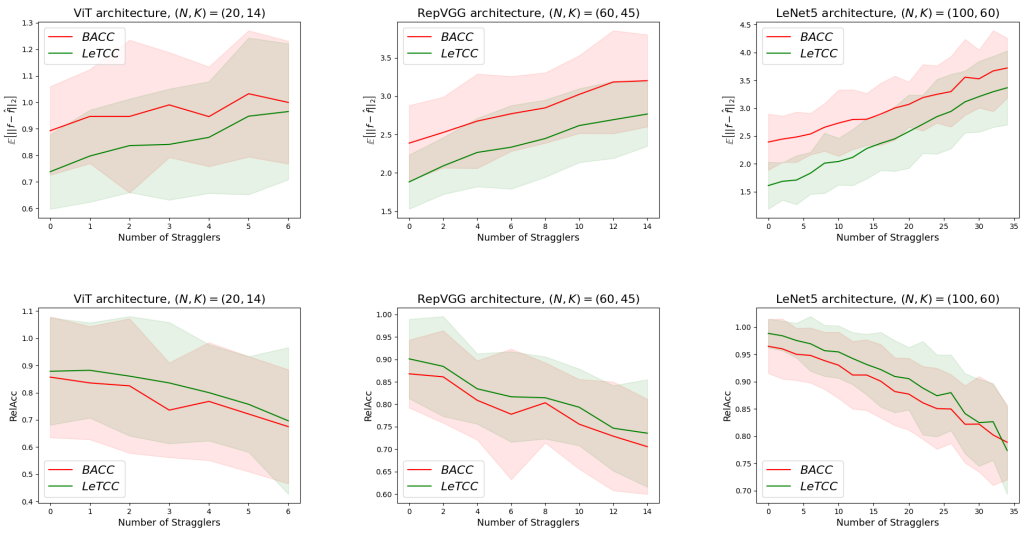


Figure 3: Performance comparison of LeTCC and BACC with a 95% confidence interval across a diverse range of stragglers for different models in a low-redundancy regime (smaller $\frac{N}{K}$).

Performance Evaluation: Table 1 presents both RMSE and RelAcc metrics side by side. The results demonstrate that LeTCC outperforms BACC across various architectures and configurations, with an average improvement of 15%, 17%, and 9% in RMSE for LeNet, RepVGG, and ViT architectures, respectively, and a 2%, 5%, and 4% enhancement in RelAcc.
In a subsequent analysis, we evaluate the performance of LeTCC in comparison to BACC across a variety of straggler scenarios. For each number of stragglers, S, we randomly select S workers to act as stragglers. Both schemes are then run with the same input data points and straggler configurations, and the process is repeated 20 times. We record the average values of the RelAcc and RMSE metrics, along with their 95% confidence intervals. Figures 3 and 4 illustrate the performance of both schemes across three model architectures. As shown in both figures, the proposed scheme consistently outperforms BACC for nearly all straggler values. In Figure 3, where $\frac{N}{K}$ is relatively small–indicating a system design without excessive redundancy, which is more practical–the proposed scheme demonstrates even greater improvements in both metrics.

Computational Complexity: The calculation and inference of smoothing spline coefficients can be performed linearly in the number of regression points by leveraging B-spline basis functions [43–45]. Consequently, the encoding and decoding processes in LeTCC, which involve evaluating new points and calculating the fitted coefficients, have computational complexities of $\mathcal{O}(K \cdot d)$ and $\mathcal{O}((N - S) \cdot m)$, respectively, where d is the input dimension and m is the output dimension of the computing function $\mathbf{f}(\cdot)$. This complexity is comparable to that of BACC, which has complexities of $\mathcal{O}(K)$ and $\mathcal{O}(N - S)$ for its encoding and decoding layers [29]. Table 2 in Appendix C.2 presents a comparison of the total end-to-end processing time statistics for the LeTCC and BACC schemes.

Sensitivity Analysis: We additionally investigate the sensitivity of the proposed schemes performance to the value of the smoothing parameter, as well as the sensitivity of the optimal smoothing parameter to the number of stragglers (or workers). The results are presented in Appendix E. As shown in Table 3 and Figure 6, the optimal smoothing parameters and the scheme's performance exhibit low sensitivity to the number of stragglers (or worker nodes) and to the smoothing parameters, respectively.

Coded Points: We also compare the coded points $\{\tilde{\mathbf{x}}_\mathbf{n}\}_{n=1}^{N}$ sent to the workers in LeTCC and BACC schemes. The results, shown in Figure 7, demonstrate that BACC coded samples exhibit high-frequency noise which causes the scheme to approximate the original prediction worse than LeTCC.

https://doi.org/10.52202/079017-3554

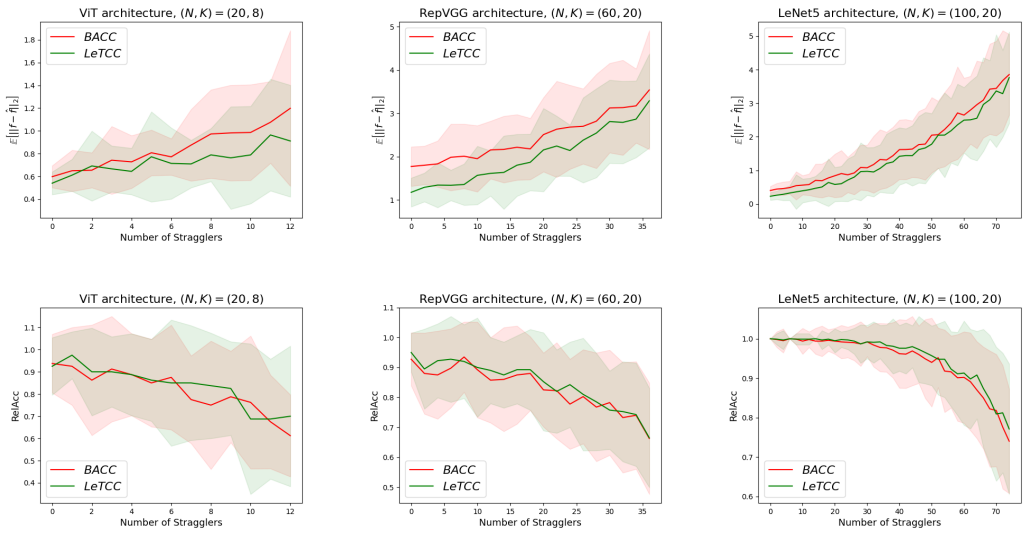


Figure 4: Performance comparison of LeTCC and BACC with a 95% confidence interval across a diverse range of stragglers for different models in a high-redundancy regime (larger $\frac{N}{K}$).

6 Related Work

Coded computing was initially introduced to tackle the challenges of distributed computation, particularly the existence of stragglers or slow workers, and also faulty or adversarial nodes. Traditional approaches to deal with stragglers primarily rely on repetition [1, 46–50], where each task is assigned to multiple workers either proactively or reactively. Recently, coded computing approaches have reduced the overhead of repetition by leveraging coding theory and embedding redundancy in the worker's input data [3, 13, 26, 29, 51–55]. This technique, which mainly relies on theory of coding, has been developed for specific types of structured computations, such as polynomial computation and matrix multiplication [3, 7, 12, 13, 17, 52, 56–58]. Recently, there have been attempts to generalize coded computing for general computations [4, 5, 29, 59]. Towards extending the application of coding computing to machine learning computation, Kosaian et al. [59] suggest training a neural network to predict coded outputs from coded data points. However, the scheme of Kosaian et al. [59] requires a complex training process and tolerates only one straggler. In another work, Jahani-Nezhad and Maddah-Ali [29] proposes BACC, a model-agnostic and numerically stable framework for general computations. They successfully employed BACC to train neural networks on a cluster of workers, while tolerating a larger number of stragglers. Building on the BACC framework, Soleymani et al. [42] introduced ApproxIFER scheme, as a straggler resistance and Byzantine-robust prediction serving system. However, the scheme of BACC uses a reasonable rational interpolation (Berrut interpolation [60]), off the shelf, for encoding and decoding, without considering any end-to-end cost function to optimize. In contrast, we theoretically formalize a new foundation of coded computing grounded in learning theory, which can be naturally used for machine learning applications.

7 Conclusions and Future Work

In this paper, we developed a new foundation for coded computing based on learning theory, contrasting with existing works that rely on coding theory and use metrics like minimum distance and recovery threshold for design. This shift in foundations removes barriers to using coded computing for machine learning applications, allows us to design optimal encoding and decoding functions, and achieves convergence rates that outperform the state of the art. Moreover, the experimental evaluations validate the theoretical guarantees. While this work focuses on straggler mitigation, future work will extend our proposed scheme to achieve Byzantine robustness and privacy, offering promising avenues for further research.

8 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant CIF-2348638. Behrooz Tahmasebi is supported by NSF Award CCF-2112665 (TILOS AI Institute) and NSF Award 2134108.

References

[1] Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2): 74–80, 2013.

[2] Vipul Gupta, Shusen Wang, Thomas Courtade, and Kannan Ramchandran. Oversketch: Approximate matrix multiplication for the cloud. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 298–304. IEEE, 2018.

[3] Qian Yu, Songze Li, Netanel Raviv, Seyed Mohammadreza Mousavi Kalan, Mahdi Soltanolkotabi, and Salman A Avestimehr. Lagrange coded computing: Optimal design for resiliency, security, and privacy. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1215–1225. PMLR, 2019.

[4] Jinhyun So, Basak Guler, and Amir Salman Avestimehr. A scalable approach for privacy-preserving collaborative machine learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 8054–8066, 2020.

[5] Jinhyun So, Başak Güler, and Amir Salman Avestimehr. CodedPrivateML: A fast and privacy-preserving framework for distributed machine learning. *IEEE Journal on Selected Areas in Information Theory*, 2(1):441–451, 2021.

[6] Zhuqing Jia and Syed Ali Jafar. On the capacity of secure distributed batch matrix multiplication. *IEEE Transactions on Information Theory*, 67(11):7420–7437, 2021.

[7] Malihe Aliasgari, Osvaldo Simeone, and Jörg Kliewer. Private and secure distributed matrix multiplication with flexible communication load. *IEEE Transactions on Information Forensics and Security*, 2020.

[8] Minchul Kim and Jungwoo Lee. Private secure coded computation. In *2019 IEEE International Symposium on Information Theory (ISIT)*, pages 1097–1101. IEEE, 2019.

[9] Wei-Ting Chang and Ravi Tandon. On the upload versus download cost for secure and private matrix multiplication. In *2019 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2019.

[10] Wei-Ting Chang and Ravi Tandon. On the capacity of secure distributed matrix multiplication. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2018.

[11] Qian Yu, Mohammad Ali Maddah-Ali, and A Salman Avestimehr. Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding. *IEEE Transactions on Information Theory*, 66(3):1920–1933, 2020.

[12] Rafael G.L. DOliveira, Salim El Rouayheb, and David Karpuk. GASP codes for secure distributed matrix multiplication. *IEEE Transactions on Information Theory*, 66(7):4038–4050, 2020.

[13] Qian Yu, Mohammad Maddah-Ali, and Salman Avestimehr. Polynomial codes: an optimal design for high-dimensional coded matrix multiplication. *Advances in Neural Information Processing Systems*, 30, 2017.

[14] Irving S Reed and Gustave Solomon. Polynomial codes over certain finite fields. *Journal of the society for industrial and applied mathematics*, 8(2):300–304, 1960.

[15] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. Draft is Available, 2022.

 https://doi.org/10.52202/079017-3554

[16] K. Lee, M. Lam, R. Pedarsani, D. Papailiopoulos, and K. Ramchandran. Speeding up distributed machine learning using codes. *IEEE Transactions on Information Theory*, 64(3):1514–1529, 2018.

[17] K. Lee, C. Suh, and K. Ramchandran. High-dimensional coded matrix multiplication. In *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, pages 2418–2422, 2017.

[18] Walter Gautschi and Gabriele Inglese. Lower bounds for the condition number of vandermonde matrices. *Numerische Mathematik*, 52(3):241–250, 1987.

[19] Gene H. Golub and Charles F. Van Loan. *Matrix computations*. JHU press, 2013.

[20] Jinhyun So, Basak Guler, and Salman Avestimehr. A scalable approach for privacy-preserving collaborative machine learning. *Advances in Neural Information Processing Systems*, 33:8054–8066, 2020.

[21] Adarsh M. Subramaniam, Anoosheh Heidarzadeh, and Krishna R. Narayanan. Random Khatri-Rao-product codes for numerically-stable distributed matrix multiplication. *CoRR*, abs/1907.05965, 2019.

[22] Aditya Ramamoorthy and Li Tang. Numerically stable coded matrix computations via circulant and rotation matrix embeddings. *IEEE Transactions on Information Theory*, 68(4):2684–2703, 2022.

[23] Anindya Bijoy Das, Aditya Ramamoorthy, and Namrata Vaswani. Efficient and robust distributed matrix computations via convolutional coding. *IEEE Transactions on Information Theory*, 67(9):6266–6282, 2021.

[24] Mahdi Soleymani, Hessam Mahdavifar, and Amir Salman Avestimehr. Analog Lagrange coded computing. *IEEE Journal on Selected Areas in Information Theory*, 2(1):283–295, 2021.

[25] Mohammad Fahim and Viveck R Cadambe. Numerically stable polynomially coded computing. *IEEE Transactions on Information Theory*, 67(5):2758–2785, 2021.

[26] Tayyebeh Jahani-Nezhad and Mohammad Ali Maddah-Ali. CodedSketch: A coding scheme for distributed computation of approximated matrix multiplication. *IEEE Transactions on Information Theory*, 67(6):4185–4196, 2021.

[27] Vipul Gupta, Shusen Wang, Thomas Courtade, and Kannan Ramchandran. OverSketch: Approximate matrix multiplication for the cloud. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 298–304, 2018.

[28] Vipul Gupta, Swanand Kadhe, Thomas Courtade, Michael W. Mahoney, and Kannan Ramchandran. Oversketched Newton: Fast convex optimization for serverless systems. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 288–297, 2020.

[29] Tayyebeh Jahani-Nezhad and Mohammad Ali Maddah-Ali. Berrut approximated coded computing: Straggler resistance beyond polynomial computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):111–122, 2022.

[30] George Kimeldorf and Grace Wahba. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95, 1971.

[31] Jean Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables: Proceedings of a Conference Held at Oberwolfach April 25–May 1, 1976*, pages 85–100. Springer, 1977.

[32] Grace Wahba. *Spline models for observational data*. SIAM, 1990.

[33] Grace Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 24(5): 383–393, 1975.

[34] Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.

[35] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[36] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[37] Yann LeCun, Corinna Cortes, Chris Burges, et al. MNIST handwritten digit database, 2010.

[38] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[39] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13733–13742, 2021.

[40] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[41] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[42] Mahdi Soleymani, Ramy E Ali, Hessam Mahdavifar, and A Salman Avestimehr. ApproxIFER: A model-agnostic approach to resilient and robust prediction serving systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8342–8350, 2022.

[43] Paul HC Eilers and Brian D Marx. Flexible smoothing with b-splines and penalties. *Statistical science*, 11(2):89–121, 1996.

[44] Carl De Boor. Calculation of the smoothing spline with weighted roughness measure. *Mathematical Models and Methods in Applied Sciences*, 11(01):33–41, 2001.

[45] Chiu Li Hu and Larry L Schumaker. Complete spline smoothing. *Numerische Mathematik*, 49: 1–10, 1986.

[46] Matei Zaharia, Andy Konwinski, Anthony D Joseph, Randy H Katz, and Ion Stoica. Improving mapreduce performance in heterogeneous environments. In *Osdi*, volume 8, page 7, 2008.

[47] Lalith Suresh, Marco Canini, Stefan Schmid, and Anja Feldmann. C3: Cutting tail latency in cloud data stores via adaptive replica selection. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 513–527, 2015.

[48] Nihar B Shah, Kangwook Lee, and Kannan Ramchandran. When do redundant requests reduce latency? *IEEE Transactions on Communications*, 64(2):715–722, 2015.

[49] Kristen Gardner, Samuel Zbarsky, Sherwin Doroudi, Mor Harchol-Balter, and Esa Hyytia. Reducing latency via redundant requests: Exact analysis. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):347–360, 2015.

[50] Manmohan Chaubey and Erik Saule. Replicated data placement for uncertain scheduling. In *2015 IEEE International Parallel and Distributed Processing Symposium Workshop*, pages 464–472. IEEE, 2015.

[51] Songze Li, Seyed Mohammadreza Mousavi Kalan, Amir Salman Avestimehr, and Mahdi Soltanolkotabi. Near-optimal straggler mitigation for distributed gradient methods. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 857–866. IEEE, 2018.

[52] Anindya B Das, Aditya Ramamoorthy, and Namrata Vaswani. Random convolutional coding for robust and straggler resilient distributed matrix computation. *arXiv preprint arXiv:1907.08064*, 2019.

 https://doi.org/10.52202/079017-3554

[53] Aditya Ramamoorthy, Anindya Bijoy Das, and Li Tang. Straggler-resistant distributed matrix computation via coding theory: Removing a bottleneck in large-scale data processing. *IEEE Signal Processing Magazine*, 37(3):136–145, 2020.

[54] Can Karakus, Yifan Sun, Suhas Diggavi, and Wotao Yin. Straggler mitigation in distributed optimization through data encoding. *Advances in Neural Information Processing Systems*, 30, 2017.

[55] R. Tandon, Q. Lei, A. G. Dimakis, and N. Karampatziakis. Gradient coding: Avoiding stragglers in distributed learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3368–3376, Sydney, Australia, August 2017.

[56] Aditya Ramamoorthy and Li Tang. Numerically stable coded matrix computations via circulant and rotation matrix embeddings. *IEEE Transactions on Information Theory*, 68(4):2684–2703, 2021.

[57] Qian Yu, Mohammad Ali Maddah-Ali, and Amir Salman Avestimehr. Straggler mitigation in distributed matrix multiplication: Fundamental limits and optimal coding. *IEEE Transactions on Information Theory*, 66(3):1920–1933, 2020.

[58] Mohammad Fahim and Viveck R. Cadambe. Numerically stable polynomially coded computing. *IEEE Transactions on Information Theory*, 67(5):2758–2785, 2021.

[59] Jack Kosaian, KV Rashmi, and Shivaram Venkataraman. Parity models: A general framework for coding-based resilience in ml inference. *arXiv preprint arXiv:1905.00863*, 2019.

[60] J-P Berrut. Rational functions for guaranteed and experimentally well-conditioned global interpolation. *Computers & Mathematics with Applications*, 15(1):1–16, 1988.

[61] Giovanni Leoni. *A first course in Sobolev spaces*, volume 181. American Mathematical Society, 2024.

[62] Robert A Adams and John JF Fournier. *Sobolev spaces*. Elsevier, 2003.

[63] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[64] David L Ragozin. Error bounds for derivative estimates based on spline smoothing of exact or noisy data. *Journal of approximation theory*, 37(4):335–355, 1983.

[65] Florencio I Utreras. Convergence rates for multivariate smoothing spline functions. *Journal of approximation theory*, 52(1):1–27, 1988.

[66] Heinrich W Guggenheimer, Alan S Edelman, and Charles R Johnson. A simple estimate of the condition number of a linear system. *The College Mathematics Journal*, 26(1):2–5, 1995.

A Preliminaries

A.1 Sobolev spaces and Sobolev norms

Let Ω be an open interval in $\mathbb{R}$ and M be a positive integer. We denote by $L^p\left(\Omega;\mathbb{R}^M\right)$ the class of all measurable functions $g:\mathbb{R}\to\mathbb{R}^M$ that satisfy:

$$\int_\Omega |g_j(t)|^p\,dt\ <\infty,\quad \forall j\in[M], \tag{13}$$

where $g(\cdot)=[g_1(\cdot),\ldots,g_M(\cdot)]^T$. The space $L^p\left(\Omega;\mathbb{R}^M\right)$ can be endowed with the following norm, known as the L^p norm:

$$\|g\|_{L^p(\Omega;\mathbb{R}^M)}:=\left(\sum_{j=1}^M\int_\Omega |g_i(t)|^p\,dt\right)^{\frac{1}{p}}, \tag{14}$$

for $1\leqslant p<\infty$, and

$$\|g\|_{L^\infty(\Omega;\mathbb{R}^M)}:=\max_{j\in[M]}\sup_{t\in\Omega}|g_j(t)|. \tag{15}$$

for $p=\infty$. Additionally, a function $g:\Omega\to\mathbb{R}^M$ is in $L^p_{\text{loc}}\left(\Omega;\mathbb{R}^M\right)$ if it lies in $L^p(V;\mathbb{R}^M)$ for all compact subsets $V\subseteq\Omega$.

Definition 1 (Sobolev Space). *The* Sobolev space $\mathbb{W}^{m,p}\left(\Omega;\mathbb{R}^M\right)$ *is the space of all functions* $g\in L^p\left(\Omega;\mathbb{R}^M\right)$ *such that all weak derivatives of order* i*, denoted by* $g^{(i)}$*, belong to* $L^p\left(\Omega;\mathbb{R}^M\right)$ *for* $i\in[m]$*. This space is endowed with the norm:*

$$\|g\|_{\mathbb{W}^{m,p}(\Omega;\mathbb{R}^M)}:=\left(\|g\|^p_{L^p(\Omega;\mathbb{R}^M)}+\sum_{i=1}^m\left\|g^{(i)}\right\|^p_{L^p(\Omega;\mathbb{R}^M)}\right)^{\frac{1}{p}}, \tag{16}$$

for $1\leqslant p<\infty$*, and*

$$\|g\|_{\mathbb{W}^{m,\infty}(\Omega;\mathbb{R}^M)}:=\max\left\{\|g\|_{L^\infty(\Omega;\mathbb{R}^M)},\max_{i\in[m]}\left\|g^{(i)}\right\|_{L^\infty(\Omega;\mathbb{R}^M)}\right\}, \tag{17}$$

for $p=\infty$*.*

Similarly, $\mathbb{W}^{m,p}_{\text{loc}}\left(\Omega;\mathbb{R}^M\right)$ is defined as the space of all functions $g\in L^p_{\text{loc}}\left(\Omega;\mathbb{R}^M\right)$ with all weak derivatives of order i belonging to $L^p_{\text{loc}}\left(\Omega;\mathbb{R}^M\right)$ for $i\in[m]$. $\|g\|_{\mathbb{W}^{m,p}_{\text{loc}}(\Omega;\mathbb{R}^M)}$ and $\|g\|_{\mathbb{W}^{m,\infty}_{\text{loc}}(\Omega;\mathbb{R}^M)}$ are defined similar to (16) and (19) respectively, using $L^p_{\text{loc}}(\Omega;\mathbb{R})$ instead of $L^p(\Omega;\mathbb{R})$.

Definition 2. ***(Sobolev Space with compact support)****: Denoted by* $\mathbb{W}^{m,p}_0\left(\Omega;\mathbb{R}^M\right)$ *collection of functions* g *defined on interval* $\Omega=(a,b)$ *such that* $g(a)=\mathbf{0},g'(a)=\mathbf{0},\ldots,g^{(m-1)}(a)=\mathbf{0}$ *and* $\left\|g^{(m)}\right\|_{L^2(\Omega;\mathbb{R})}<\infty$*. This space can be endowed with the following norm:*

$$\|g\|_{\mathbb{W}^{m,p}(\Omega;\mathbb{R}^M)}:=\left\|g^{(m)}\right\|_{L^p(\Omega;\mathbb{R}^M)}, \tag{18}$$

for $1\leqslant p<\infty$*, and*

$$\|g\|_{\mathbb{W}^{m,\infty}(\Omega;\mathbb{R}^M)}:=\left\|g^{(m)}\right\|_{L^\infty(\Omega;\mathbb{R}^M)}, \tag{19}$$

for $p=\infty$*.*

The next theorem provides an upper bound for L^p norm of functions in the Sobolev space, which plays a crucial role in the proof of the main theorems of the paper.

Theorem 5 (Theorem 7.34, [61])**.** *Let* $\Omega\subseteq\mathbb{R}$ *be an open interval and let* $g\in\mathbb{W}^{1,1}_{loc}\left(\Omega;\mathbb{R}^M\right)$*. Assume* $1\leqslant p,q,r\leqslant\infty$ *and* $r\geqslant q$*. Then:*

$$\|g\|_{L^r(\Omega;\mathbb{R}^M)}\leqslant\ell^{\frac{1}{r}-\frac{1}{q}}\|g\|_{L^q(\Omega;\mathbb{R}^M)}+\ell^{1-\frac{1}{p}+\frac{1}{r}}\|g'\|_{L^p(\Omega;\mathbb{R}^M)}, \tag{20}$$

for every $0<\ell<\mathcal{L}^1(\Omega)$*.*

https://doi.org/10.52202/079017-3554

Note that $\mathcal{L}^1(\Omega)$ in Theorem 5 is the length of the interval Ω.

Corollary 1. *Suppose $g \in \mathbb{W}^{1,1}_{loc}\left(\Omega;\mathbb{R}^M\right)$ and $\Omega \subseteq \mathbb{R}$ be an open interval. If $\frac{\|g\|_{L^2(\Omega;\mathbb{R}^M)}}{\|g'\|_{L^2(\Omega;\mathbb{R}^M)}} < \mathcal{L}^1(\Omega)$, then:*

$$\|g\|_{L^\infty(\Omega;\mathbb{R}^M)} \leqslant 2\sqrt{\|g\|_{L^2(\Omega;\mathbb{R}^M)} \cdot \|g'\|_{L^2(\Omega;\mathbb{R}^M)}}. \tag{21}$$

Proof. Substituting $p, q = 2$ and $r = \infty$ in Theorem 5 and optimizing over ℓ, one can derive the optimum value of ℓ, denoted by ℓ^* as

$$\ell^* = \frac{\|g\|_{L^2(\Omega;\mathbb{R}^M)}}{\|g'\|_{L^2(\Omega;\mathbb{R}^M)}}. \tag{22}$$

Since $\frac{\|g\|_{L^2(\Omega;\mathbb{R}^M)}}{\|g'\|_{L^2(\Omega;\mathbb{R}^M)}} < \mathcal{L}^1(\Omega)$, the optimum value is in the valid interval mentioned in Theorem 5. □

Corollary 2 (Corollary 7.36, [61])**.** *Let $\Omega = (a, b)$, let $1 \leq p, q, r \leq \infty$ be such that $1 + 1/r \geq 1/p$ and $r \geq q$ and let $g \in \mathbb{W}^{1,1}_{loc}\left(\Omega;\mathbb{R}^M\right)$ with $g' \in L^p\left(\Omega;\mathbb{R}^M\right)$. Let $x_0 \in [a, b]$ be such that $|g(x_0)| = \min_{[a,b]} |g|$. Then*

$$\|g - g(x_0)\|_{L^r(\Omega;\mathbb{R}^M)} \leq 8\|g\|^{\alpha}_{L^q(\Omega;\mathbb{R}^M)} \|g'\|^{1-\alpha}_{L^p(\Omega;\mathbb{R}^M)} \tag{23}$$

where $\alpha := 0$ if $r = q$ and $1 - 1/p + 1/r = 0$ and otherwise

$$\alpha := \frac{1 - 1/p + 1/r}{1 - 1/p + 1/q}.$$

Corollary 3. *Theorem 5 and Corollary 2 hold true when $f \in \mathbb{W}^{2,2}\left(\Omega;\mathbb{R}^M\right)$.*

Proof. We prove the above corollary by showing that $\mathbb{W}^{2,2}\left(\Omega;\mathbb{R}^M\right) \subseteq \mathbb{W}^{1,1}\left(\Omega;\mathbb{R}^M\right) \subseteq \mathbb{W}^{1,1}_{\text{loc}}\left(\Omega;\mathbb{R}^M\right)$. Using Cauchy-Schwartz inequality, one can show:

$$\begin{aligned}\|f\|_{L^1(\Omega;\mathbb{R}^M)} = \sum_{j=1}^{M} \int_\Omega |f_j(t)|\, dt &\leqslant \sum_{j=1}^{M} \left(\int_\Omega 1^2\, dt \cdot \int_\Omega |f_j(t)|^2\, dt\right)^{\frac{1}{2}} \\ &\leqslant \left(\mathcal{L}^1(\Omega)\right)^{\frac{1}{2}} \cdot \sum_{j=1}^{M} \left(\int_\Omega |f_j(t)|^2\, dt\right)^{\frac{1}{2}} \overset{(a)}{<} \infty,\end{aligned} \tag{24}$$

where (a) follows from the bounded length of Ω and $f \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$. Similarly:

$$\begin{aligned}\|f'\|_{L^1(\Omega;\mathbb{R}^M)} = \sum_{j=1}^{M} \int_\Omega |f_j'(t)|\, dt &\leqslant \sum_{j=1}^{M} \left(\int_\Omega 1^2\, dt \cdot \int_\Omega |f_j'(t)|^2\, dt\right)^{\frac{1}{2}} \\ &\leqslant \left(\mathcal{L}^1(\Omega)\right)^{\frac{1}{2}} \cdot \sum_{j=1}^{M} \left(\int_\Omega |f_j'(t)|^2\, dt\right)^{\frac{1}{2}} \overset{(a)}{<} \infty.\end{aligned} \tag{25}$$

Equations (24) and (25) prove that $\mathbb{W}^{2,2}\left(\Omega;\mathbb{R}^M\right) \subseteq \mathbb{W}^{1,1}\left(\Omega;\mathbb{R}^M\right)$. Now, suppose $f \in \mathbb{W}^{1,1}\left(\Omega;\mathbb{R}^M\right)$. For every closed subset $[c, d] \subset \Omega$ we have:

$$\sum_{j=1}^{M} \left(\int_c^d |f_j(t)|\, dt\right) \overset{(a)}{\leqslant} \sum_{j=1}^{M} \left(\int_\Omega |f_j(t)|\, dt\right) \overset{(b)}{\leqslant} \infty, \tag{26}$$

$$\sum_{j=1}^{M} \left(\int_c^d |f_j'(t)|\, dt\right) \overset{(a)}{\leqslant} \sum_{j=1}^{M} \left(\int_\Omega |f_j'(t)|\, dt\right) \overset{(b)}{\leqslant} \infty, \tag{27}$$

where (a) is due to $[c, d] \subset \Omega$ and (b) is because of $f \in \mathbb{W}^{1,1}\left(\Omega;\mathbb{R}^M\right)$. Therefore, $f \in \mathbb{W}^{1,1}_{\text{loc}}\left(\Omega;\mathbb{R}^M\right)$. □

Equivalent norms. There have been various norms defined on Sobolev spaces in the literature that are equivalent to (16) (see [62], [63, Ch. 7], and [32, Sec. 10.2]). Note that two norms $\|\cdot\|_{W_1}, \|\cdot\|_{W_2}$ are equivalent if there exist positive constants η_1, η_2 such that $\eta_1 \cdot \|g\|_{W_2} \leqslant \|g\|_{W_1} \leqslant \eta_2 \cdot \|g\|_{W_2}$. The equivalent norm in which we are interested is the one introduced in [30]. Let $\Omega = (a, b) \subset \mathbb{R}$. We define $\widetilde{\mathbb{W}}^{m,p}\left(\Omega; \mathbb{R}^M\right)$ as the Sobolev space endowed with the following norm:

$$\|g\|_{\widetilde{\mathbb{W}}^{m,p}(\Omega;\mathbb{R}^M)} := \left(\sum_{j=1}^{M} \left(g_j(a)^p + \sum_{i=1}^{m-1} \left(g_j^{(i)}(a) \right)^p \right) + \left\| g^{(m)} \right\|_{L^p(\Omega;\mathbb{R}^M)}^p \right)^{\frac{1}{p}}. \tag{28}$$

The following lemma derives the equivalence constants (η_1, η_2) for the norms $\|\cdot\|_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}$ and $\|\cdot\|_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}$.

Lemma 1. *Let $\Omega = (a, b)$ be an arbitrary open interval in $\mathbb{R}$. Then for every $g \in \mathbb{W}^{2,2}(\Omega; \mathbb{R})$:*

$$\|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})} \leqslant \left[2(b-a) \max\{1, (b-a)\} \left(2 \max\{1, (b-a)\}^2 + 1 \right) + 1 \right] \cdot \|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})} \tag{29}$$

$$\|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})} \leqslant \left(\frac{4}{(b-a)} \max\{1, (b-a)\}^2 + 1 \right) \cdot \|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}. \tag{30}$$

Proof. By expanding g around the point a and utilizing the integral remainder form of Taylor's expansion, for every $x \in (a, b)$ we have:

$$g(x) = g(a) + \int_a^x g'(t)\, dt. \tag{31}$$

Therefore:

$$\begin{aligned} g(x)^2 &= g(a)^2 + \left(\int_a^x g'(t)\, dt \right)^2 + 2g(a) \cdot \int_a^x g'(t)\, dt \\ &\overset{(a)}{\leqslant} 2g(a)^2 + 2 \left(\int_a^x g'(t)\, dt \right)^2 \\ &\overset{(b)}{\leqslant} 2g(a)^2 + 2(x-a) \int_a^x g'(t)^2\, dt \\ &\overset{(c)}{\leqslant} 2g(a)^2 + 2(b-a) \int_a^b g'(t)^2\, dt, \end{aligned} \tag{32}$$

where (a) follows from the $2g(a) \cdot \int_a^x g'(t)\, dt \leqslant g(a)^2 + \left(\int_a^x g'(t)\, dt \right)^2$, (b) is based on Cauchy-Schwartz inequality, and (c) is due to $x \leqslant b$. Following the same steps as (32), we have:

$$g'(x)^2 \leqslant 2g'(a)^2 + 2(b-a) \int_a^b g''(t)^2\, dt. \tag{33}$$

Integrating both sides of (33) we have:

$$\begin{aligned} \int_a^b g'(y)^2\, dy &\leqslant 2 \int_a^b g'(a)^2\, dy + 2(b-a) \int_a^b \left(\int_a^b g''(t)^2\, dt \right) dy \\ &= 2(b-a).g'(a)^2 + 2(b-a)^2 \int_a^b g''(t)^2\, dt \\ &\overset{(a)}{\leqslant} 2(b-a) \cdot \max\{1, (b-a)\} \cdot \left(g(a)^2 + g'(a)^2 + \int_a^b g''(t)^2\, dt \right) \\ &= 2(b-a) \cdot \max\{1, (b-a)\} \cdot \|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}, \end{aligned} \tag{34}$$

https://doi.org/10.52202/079017-3554

where (a) is based on adding the positive term $(b-a)\cdot g(a)^2$ to the right-hand side. Based on the (32) and (34) we have:

$$\begin{aligned}\int_a^b g(x)^2\,dx &\overset{(a)}{\leqslant} 2\int_a^b g(a)^2\,dx + 2(b-a)\int_a^b\left(\int_a^b g'(t)^2\,dt\right)dx\\ &= 2(b-a).g(a)^2 + 2(b-a)^2\int_a^b g'(t)^2\,dt\\ &\overset{(b)}{\leqslant} 2(b-a)\cdot g(a)^2 + 4(b-a)^3\cdot g'(a)^2 + 4(b-a)^4\int_a^b g''(t)^2\,dt\\ &\leqslant 4(b-a)\cdot\max\{1,(b-a)^3\}\cdot\|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}\,, \end{aligned}\tag{35}$$

where (a) follows by integrating both sides of (32) and (b) is due to (33). Using (35) and (34) and the fact that $\int_a^b g''(t)^2\,dt \leqslant \|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}$, we have:

$$\begin{aligned}\|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})} &= \int_a^b g''(t)^2\,dt + \int_a^b g'(t)^2\,dt + \int_a^b g(t)^2\,dt\\ &\leqslant \left[2(b-a)\max\{1,(b-a)\}\left(2\max\{1,(b-a)\}^2+1\right)+1\right]\cdot\|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}\,. \end{aligned}\tag{36}$$

For the other side, using the same steps as in (32), we have:

$$\begin{aligned}g(a)^2 &= g(x)^2 + \left(\int_a^x g'(t)\,dt\right)^2 - 2g(a).\int_a^x g'(t)\,dt\\ &\overset{(a)}{\leqslant} 2g(x)^2 + 2\left(\int_a^x g'(t)\,dt\right)^2\\ &\overset{(b)}{\leqslant} 2g(x)^2 + 2(x-a)\int_a^x g'(t)^2\,dt\\ &\overset{(c)}{\leqslant} 2g(x)^2 + 2(b-a)\int_a^b g'(t)^2\,dt, \end{aligned}\tag{37}$$

where (a) is because of $-2g(a)\cdot\int_a^x g'(t)\,dt \leqslant g(a)^2 + \left(\int_a^x g'(t)\,dt\right)^2$, (b) follows from Cauchy-Schwartz inequality, and (c) is due to $x \leqslant b$. Integrating both sides of (37), we have

$$\begin{aligned}(b-a)\cdot g(a)^2 = \int_a^b g(a)^2\,dx &\leqslant 2\int_a^b g(x)^2\,dx + 2(b-a)\int_a^b\left(\int_a^b g'(t)^2\,dt\right)dx\\ &\overset{(a)}{=} 2\int_a^b g(t)^2\,dt + 2(b-a)^2\int_a^b g'(t)^2\,dt\\ &\overset{(b)}{\leqslant} 2\max\{1,(b-a)\}^2\left(\int_a^b g(t)^2\,dt + \int_a^b g'(t)^2\,dt + \int_a^b g''(t)^2\,dt\right)\\ &\leqslant 2\max\{1,(b-a)\}^2\,\|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}\,, \end{aligned}\tag{38}$$

where (a) follows by a change of variable from x to t and (b) follows by adding the positive term $2\max\{1,(b-a)\}^2\cdot\int_a^b g''(t)^2\,dt$ to the right side. Thus, (38) directly results in

$$g(a)^2 \leqslant \frac{2}{(b-a)}\max\{1,(b-a)\}^2\,\|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}\,.\tag{39}$$

Following same steps as (38) and (39) results in

$$g'(a)^2 \leqslant \frac{2}{(b-a)}\max\{1,(b-a)\}^2\,\|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}\,.\tag{40}$$

Considering the fact that $\int_a^b g''(t)^2\,dt \leqslant \|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}$, we can complete the proof:

$$\begin{aligned}\|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})} &= g(a)^2 + g'(a)^2 + \int_a^b g''(t)^2\,dt\\ &\leqslant \left(\frac{4}{(b-a)}\max\{1,(b-a)\}^2+1\right)\cdot\|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}\,. \end{aligned}\tag{41}$$

□

Corollary 4. *Based on Lemma 1, for $\Omega = (-1, 1)$ we have*

$$\frac{1}{9} \cdot \|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})} \leqslant \|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})} \leqslant 73 \cdot \|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}. \tag{42}$$

Corollary 4 directly follows from Lemma 1 by substituting $(a, b) = (-1, 1)$.

Corollary 5. *The result of Lemma 1 remains valid for multi-dimensional cases, where $g : \mathbb{R} \to \mathbb{R}^M$, for some $M > 1$.*

Corollary 5 directly follows from applying Lemma 1 to each component of the function $g(\cdot)$ and using the definition of vector-valued function norm:

$$\|g\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R}^M)} = \sum_{j=1}^{M} \left\|g^{(j)}\right\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}, \quad \|g\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R}^M)} = \sum_{j=1}^{M} \left\|g^{(j)}\right\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}.$$

Proposition 2. *([61, Section 7.2], [63, Theorem 121],[32]) For any open interval $\Omega \subseteq \mathbb{R}$ and $m, M \in \mathbb{N}$,*

$$\begin{aligned} \mathcal{H}^m\left(\Omega;\mathbb{R}^M\right) &:= \mathbb{W}^{m,2}\left(\Omega;\mathbb{R}^M\right), \\ \widetilde{\mathcal{H}}^m\left(\Omega;\mathbb{R}^M\right) &:= \widetilde{\mathbb{W}}^{m,2}\left(\Omega;\mathbb{R}^M\right), \\ \mathcal{H}_0^m\left(\Omega;\mathbb{R}^M\right) &:= \mathbb{W}_0^{m,2}\left(\Omega;\mathbb{R}^M\right), \end{aligned} \tag{43}$$

are Reproducing Kernel Hilbert Spaces (RKHSs).

The full expression of the kernel function of $\mathcal{H}^m(\Omega;\mathbb{R})$ and $\widetilde{\mathcal{H}}^m(\Omega;\mathbb{R})$ and other equivalent norms of Sobolev spaces can be found in [63, Section 4]. For $\widetilde{\mathcal{H}}^m(\Omega;\mathbb{R})$ the kernel function is as follows:

$$R(t,s) = \sum_{j=0}^{m-1} \frac{t^j s^j}{j!^2} + \int_\Omega \frac{(t-x)_+^{m-1}(s-x)_+^{m-1}}{(m-1)!^2}\, dx, \tag{44}$$

where $(\cdot)_+$ is positive part function.

A.2 Smoothing Splines

Consider the data model $y_i = f(t_i) + \epsilon_i$ for $i = 1, \ldots, n$, where $t_i \in \Omega = (a, b) \subset \mathbb{R}$, $\mathbb{E}[\epsilon_i] = 0$, and $\mathbb{E}[\epsilon_i^2] \leqslant \sigma_0^2$. Assuming $f \in \widetilde{\mathbb{W}}^{m,2}(\Omega;\mathbb{R})$, the solution to the following optimization problem is referred to as the smoothing spline:

$$\mathbf{S}_{\lambda,n,m}[\mathbf{y}] := \underset{g \in \widetilde{\mathbb{W}}^{m,2}(\Omega;\mathbb{R})}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(g(t_i) - y_i\right)^2 + \lambda \int_\Omega \left(g^{(m)}(t)\right)^2 dt, \tag{45}$$

where $\mathbf{y} = [y_1, \ldots, y_n]$. Based on Proposition 2, $\widetilde{\mathcal{H}}^m(\Omega;\mathbb{R}) := \widetilde{\mathbb{W}}^{m,2}(\Omega;\mathbb{R})$ with the norm $\|\cdot\|_{\widetilde{\mathbb{W}}^{m,2}(\Omega;\mathbb{R})}$ is a RKHS for some kernel function $\phi(\cdot,\cdot)$. Therefore for any $v \in \widetilde{\mathbb{W}}^{m,2}(\Omega;\mathbb{R})$, we have:

$$v(t) = \langle v(\cdot), \phi(\cdot, t)\rangle_{\widetilde{\mathcal{H}}^m(\Omega;\mathbb{R})}. \tag{46}$$

It can be shown that $\phi(t,s) = R^P(t,s) + R^0(t,s)$ where $R^0(t,s)$ is kernel function of $\mathcal{H}_0^m(\Omega;\mathbb{R})$ and $R^P(t,s)$ is a null space of $\mathcal{H}_0^m(\Omega;\mathbb{R})$ which is the space of all polynomials with degree less than m.

The solution of (45) has the following form [31, 32]:

$$u^*(\cdot) = \sum_{i=1}^{m} d_i \zeta_i(\cdot) + \sum_{j=1}^{n} c_j \nu_j(\cdot), \tag{47}$$

https://doi.org/10.52202/079017-3554

where $\nu_j(\cdot) = R^0(\cdot, t_j)$ for $j \in [n]$, and $\{\zeta_i(\cdot)\}_{i=1}^m$ are the basis functions of the space of polynomials of degree at most $m-1$. Substituting u^* into (45) and optimizing over $\mathbf{c} = [c_1, \ldots, c_n]^T$ and $\mathbf{d} = [d_1, \ldots, d_m]^T$, we obtain the following result [32]:

$$\mathbf{S}_{\lambda,n,m}[\mathbf{y}](\mathbf{y}) = \mathbf{Q}\left(\mathbf{Q}^\mathbf{T}\mathbf{Q} + \lambda\mathbf{\Gamma}\right)^{-1}\mathbf{Q}^\mathbf{T}\mathbf{y}, \tag{48}$$

where

$$\begin{aligned}
\mathbf{Q}_{n\times(n+m)} &= \begin{bmatrix} \mathbf{T}_{n\times m} & \mathbf{\Sigma}_{n\times n} \end{bmatrix}, \\
\mathbf{\Gamma}_{(n+m)\times(n+m)} &= \begin{bmatrix} \mathbf{0}_{m\times m} & \mathbf{0}_{m\times n} \\ \mathbf{0}_{n\times 1} & \mathbf{\Sigma}_{n\times n} \end{bmatrix}, \\
\boldsymbol{T}_{ij} &= \zeta_j(t_i), \\
\mathbf{\Sigma}_{ij} &= R^0(t_i, t_j).
\end{aligned} \tag{49}$$

Equation (48) states that the smoothing spline fitted on the data points $\mathbf{y}$ is a linear operator:

$$\mathbf{S}_{\lambda,n,m}[\mathbf{y}](\mathbf{z}) := \mathbf{A}_\lambda\mathbf{z}, \tag{50}$$

for $\mathbf{z} \in \mathbb{R}^n$, where $\mathbf{A}_\lambda := \mathbf{Q}\left(\mathbf{Q}^\mathbf{T}\mathbf{Q} + \lambda\mathbf{\Gamma}\right)^{-1}\mathbf{Q}^\mathbf{T}$. It can be shown that the $u^*(\cdot)$ is a natural spline [32, 33]. Thus, if $\{b_i(\cdot)\}_{i=1}^n$ is a basis function for an m-th order natural spline (such as truncated power or B-spline basis functions), we have:

$$u^*(t) = \sum_{i=1}^n \xi_i b_i(t), \tag{51}$$

$$\boldsymbol{\xi} = \left(\mathbf{N}^T\mathbf{N} + \lambda\Phi\right)^{-1}\mathbf{N}^T\mathbf{y}, \tag{52}$$

where $\mathbf{N}_{ij} = b_i(t_j)$, $\Phi_{ij} = \int_\Omega b_i''(t)b_j''(t)\,dt$ for $i, j \in [n]$, and $\boldsymbol{\xi} := [\xi_1, \ldots, \xi_n]^T$.

To characterize the estimation error of the smoothing spline, $|f - \mathbf{S}_{\lambda,n,m}(\mathbf{y})|$, we need to define two variables analogous to those in (6), which quantify the minimum and maximum consecutive distance of the regression points $\{t_i\}_{i=1}^n$:

$$\Delta_{\text{max}} := \max_{i\in\{0\}\cup[n]}\{t_{i+1} - t_i\}, \quad \Delta_{\text{min}} := \min_{i\in[n-1]}\{t_{i+1} - t_i\}, \tag{53}$$

where boundary points are defined as $(t_0, t_{n+1}) := (a, b)$. The following theorem offers an upper bound for the j-th derivative of the smoothing spline estimator error function in the absence of noise ($\sigma_0 = 0$).

Theorem 6. *([64, Theorem 4.10]) Consider data model $y_i = f(t_i)$ with $\{t_i\}_{i=1}^n$ belong to $\Omega = [a, b]$ for $i \in [n]$. Let*

$$L = p_{2(m-1)}\left(\frac{\Delta_{max}}{\Delta_{min}}\right)\cdot\frac{n\Delta_{max}}{b-a}\frac{\lambda}{2} + D(m)\cdot(\Delta_{max})^{2m}, \tag{54}$$

where $p_d(\cdot)$ is a degree d polynomial with positive weights and $D(m)$ is a function of m. Then for each $j \in \{0, 1, \ldots, m\}$ and any $f \in \mathbb{W}^{m,2}(\Omega; \mathbb{R})$, there exist a function $H(m, j)$ such that:

$$\left\|\left(f - \mathbf{S}_{\lambda,n,m}(\mathbf{y})\right)^{(j)}\right\|^2_{L^2(\Omega;\mathbb{R})} \leqslant H(m,j)\left(1 + \left(\frac{L}{(b-a)^{2m}}\right)\right)^{\frac{j}{m}}\cdot L^{\frac{(m-j)}{m}}\cdot\left\|f^{(m)}\right\|^2_{L^2(\Omega;\mathbb{R})}. \tag{55}$$

Note that $\left(f - \mathbf{S}_{\lambda,n,m}(\mathbf{y})\right)^{(0)} := f - \mathbf{S}_{\lambda,n,m}(\mathbf{y})$. In the presence of noise, where $\sigma_0 > 0$, we can exploit the linearity of the smoothing spline operator and the mutual independence of the noise terms to conclude that:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\epsilon}}\left[\|(f - \mathbf{S}_{\lambda,n,m}(\mathbf{y}))\|^2_{L^2(\Omega;\mathbb{R})}\right] &= \mathbb{E}_{\boldsymbol{\epsilon}}\left[\|(f - \mathbf{S}_{\lambda,n,m}[\mathbf{y}](\mathbf{f} + \boldsymbol{\epsilon}))\|^2_{L^2(\Omega;\mathbb{R})}\right] \\
&= \|(f - \mathbf{S}_{\lambda,n,m}[\mathbf{y}](\mathbf{f}))\|^2_{L^2(\Omega;\mathbb{R})} \\
&\quad + \mathbb{E}_{\boldsymbol{\epsilon}}\left[\|(f - \mathbf{S}_{\lambda,n,m}[\mathbf{y}](\boldsymbol{\epsilon}))\|^2_{L^2(\Omega;\mathbb{R})}\right],
\end{aligned} \tag{56}$$

where $\mathbf{f} = [f(t_1), \ldots, f(t_n)]^T$. The first term in (56) can be upper bounded using Theorem 6. The following theorem establishes an upper bound for the second term when $\frac{\Delta_{\text{max}}}{\Delta_{\text{min}}}$ is bounded:

Theorem 7. *([65, Section 5]) Consider data model $y_i = f(t_i)+\epsilon_i$, where $\mathbb{E}[\epsilon_i] = 0$ and $\mathbb{E}[\epsilon_i^2] \leqslant \sigma_0^2$ for $i \in [n]$. Assume there exist a constant $B > 0$ such that $\frac{\Delta_{max}}{\Delta_{min}} \leqslant B$. Then for each $j \in \{0, 1, \ldots, m\}$, there exist a constant $\lambda_0 > 0$ and function $Q(m, j, \lambda_0)$ such that:*

$$\mathbb{E}_{\boldsymbol{\epsilon}}\left[\left\|\left(f - \mathbf{S}_{\lambda,n,m}[\mathbf{y}](\boldsymbol{\epsilon})\right)^{(j)}\right\|^2_{L^2(\Omega;\mathbb{R})}\right] \leqslant \frac{Q(m, j, \lambda_0) \cdot \sigma_0^2}{n}\lambda^{\frac{-(2j+1)}{2m}} \tag{57}$$

for $\lambda \leqslant \lambda_0$ and $n\lambda^{\frac{1}{2m}} \geqslant 1$.

Note that based on [65], $Q(m, j, \lambda_0) = w(m, j)\lambda_0^{\frac{1}{2m}} + \tilde{w}(m, j)$.

B Proof of Theorems

Recall from (2) that $\mathcal{R}(\hat{f}) \leqslant \mathcal{L}_{\text{enc}}(\hat{f}) + \mathcal{L}_{\text{dec}}(\hat{f})$, where

$$\mathcal{L}_{\text{dec}}(\hat{f}) = \mathbb{E}_{\epsilon,\mathcal{F}\sim F_{S,N}}\left[\frac{2}{K}\sum_{k=1}^{K}\left(u_{\text{dec}}(\alpha_k) - f(u_{\text{enc}}(\alpha_k))\right)^2\right], \tag{58}$$

$$\mathcal{L}_{\text{enc}}(\hat{f}) = \frac{2}{K}\sum_{k=1}^{K}\left(f(u_{\text{enc}}(\alpha_k)) - f(x_k)\right)^2. \tag{59}$$

We begin by deriving a general intermediate bound for $\mathcal{L}_{\text{dec}}$ and $\mathcal{L}_{\text{enc}}$ which will be a key component in the proofs of Theorems 2 and 1. The subsequent subsections will then provide the remaining details to complete the proofs of both theorems.

Lemma 2. *Let $f : \mathbb{R} \to \mathbb{R}$ be a q-Lipschitz continuous function. Then:*

$$\mathcal{L}_{enc} \leqslant \frac{2q^2}{K}\sum_{k=1}^{K}(u_{enc}(\alpha_k) - x_k)^2. \tag{60}$$

Proof. Using Lipschitz property, we have:

$$\begin{aligned}
\mathcal{L}_{\text{enc}} &= \frac{2}{K}\sum_{k=1}^{K}\left(f(u_{\text{enc}}(\alpha_k)) - f(x_k)\right)^2 \\
&= \frac{2}{K}\sum_{k=1}^{K}\left(|f(u_{\text{enc}}(\alpha_k)) - f(x_k)|\right)^2 \\
&\leqslant \frac{2}{K}\sum_{k=1}^{K}\left(q \cdot |u_{\text{enc}}(\alpha_k) - x_k|\right)^2 \\
&= \frac{2q^2}{K}\sum_{k=1}^{K}(u_{\text{enc}}(\alpha_k) - x_k)^2.
\end{aligned} \tag{61}$$

□

As previously mentioned, $\mathcal{L}_{\text{dec}}$ represents the expected generalization error of the decoder function. To leverage the results from Theorems 6 and 7 we must establish that the composition of f with the encoder u_{enc} belongs to the Sobolev space $\mathbb{W}^{2,2}(\Omega;\mathbb{R})$.

Lemma 3. *Let $f : \mathbb{R} \to \mathbb{R}$ be a q-Lipschitz continuous function with $\|f''\|_{L^\infty(\Omega;\mathbb{R})} \leqslant \nu$ and $\Omega \subset \mathbb{R}$ be an open interval. If $u_{enc} \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$ then $f \circ u_{enc} \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$.*

Proof. Let us define $f_0(t) := f(t) - f(0)$. Thus $f_0(0) = 0$ and f_0 is q-Lipschitz. Using Lipschitz property

$$|f_0(u_{\text{enc}}(t))|^2 = |f_0(u_{\text{enc}}(t)) - f_0(0)|^2 \leqslant q^2 \cdot u_{\text{enc}}(t)^2. \tag{62}$$

https://doi.org/10.52202/079017-3554

Integrating both sides of (62):

$$\int_{\Omega} (f_0 \circ u_{\text{enc}}(t))^2 \, dt \leqslant q^2 \cdot \int_{\Omega} u_{\text{enc}}(t)^2 \, dt \overset{\text{(a)}}{<} \infty, \tag{63}$$

where (a) follows by $u_{\text{enc}} \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$. Given that f_0 is q-Lipschitz, its derivative is bounded in the $L^{\infty}(\Omega;\mathbb{R})$-norm, i.e., $\|f_0'\|_{L^{\infty}(\Omega;\mathbb{R})} \leqslant q$. Thus

$$\int_{\Omega} ((f_0 \circ u_{\text{enc}}(t))')^2 \, dt \overset{\text{(a)}}{=} \int_{\Omega} (f_0'(u_{\text{enc}}(t)))^2 \cdot u_{\text{enc}}'(t)^2 \, dt \leqslant q^2 \cdot \int_{\Omega} u_{\text{enc}}'(t)^2 \, dt \overset{\text{(b)}}{<} \infty, \tag{64}$$

where (a) follows by chain rule and (b) follows by $u_{\text{enc}} \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$. For the second derivative we have:

$$\begin{aligned}
\int_{\Omega} [f_0(u_{\text{enc}}(t))'']^2 \, dt &\overset{\text{(a)}}{=} \int_{\Omega} \left[u_{\text{enc}}''(t) \cdot f_0'(u_{\text{enc}}(t)) + u_{\text{enc}}'(t)^2 \cdot f_0''(u_{\text{enc}}(t))\right]^2 \, dt \\
&\overset{\text{(b)}}{\leqslant} \int_{\Omega} \left[u_{\text{enc}}''(t)^2 + u_{\text{enc}}'(t)^4\right] \left[f_0'(u_{\text{enc}}(t))^2 + f_0''(u_{\text{enc}}(t))^2\right] \, dt \\
&\overset{\text{(c)}}{\leqslant} (q^2+\nu^2) \int_{\Omega} \left[u_{\text{enc}}''(t)^2 + u_{\text{enc}}'(t)^4\right] \, dt \\
&= (q^2+\nu^2) \left(\|u_{\text{enc}}''(t)\|_{L^2(\Omega;\mathbb{R})}^2 + \|u_{\text{enc}}'(t)\|_{L^4(\Omega;\mathbb{R})}^4\right) \\
&\overset{\text{(d)}}{\leqslant} (q^2+\nu^2) \left(\|u_{\text{enc}}''(t)\|_{L^2(\Omega;\mathbb{R})}^2 + \left(\|u_{\text{enc}}'(t)\|_{L^2(\Omega;\mathbb{R})} + \|u_{\text{enc}}''(t)\|_{L^2(\Omega;\mathbb{R})}\right)^4\right) \\
&\overset{\text{(e)}}{<} \infty,
\end{aligned} \tag{65}$$

where (a) follows from the chain rule, (b) is derived using the Cauchy-Schwartz inequality, (c) is due to $\|f_0''\|_{L^{\infty}(\Omega;\mathbb{R})} \leqslant \nu$, (d) follows from Theorem 5 with $r = 4, p = q = 2, l = 1$, and (e) is a result of $u_{\text{enc}} \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$. Equations (63), (64), and (63) demonstrate that $f_0 \circ u_{\text{enc}} \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$. Note that Ω is bounded, and every constant function belongs to $\mathbb{W}^{2,2}(\Omega;\mathbb{R})$. Thus, we can conclude that $f_0 \circ u_{\text{enc}}(t) + f(0) = f \circ u_{\text{enc}}(t) \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$. □

Let us define the function $h(t) := u_{\text{dec}}(t) - f(u_{\text{enc}}(t))$. Based on Lemma 3, and given that u_{dec} and $f \circ u_{\text{enc}}$ belong to the Sobolev space $\mathbb{W}^{2,2}(\Omega;\mathbb{R})$, it follows that $h \in \mathbb{W}^{2,2}(\Omega;\mathbb{R})$. The subsequent lemmas establish upper bounds for $\|h\|_{L^{\infty}(\Omega;\mathbb{R})}$ and $\|h'\|_{L^{\infty}(\Omega;\mathbb{R})}$, leveraging properties of functions in Sobolev spaces.

Lemma 4. *If $\Omega = (-a, a)$, then:*

$$\|h\|_{L^2(\Omega;\mathbb{R})} \leqslant \|h'\|_{L^2(\Omega;\mathbb{R})} \cdot \sqrt{x_0^2 + a^2} \leqslant \|h'\|_{L^2(\Omega;\mathbb{R})} \cdot \sqrt{2}a \tag{66}$$

Proof. Assume $\exists\, x_0 \in \Omega : h(x_0) = 0$. Therefore, $|h(x)| = \left|\int_{x_0}^{x} h'(x)\, dx\right|$ for $x \in [x_0, a)$. Thus

$$|h(x)| = \left|\int_{x_0}^{x} h'(x)\, dx\right| \overset{\text{(a)}}{\leqslant} \int_{x_0}^{x} |h'(x)| \, dx \overset{\text{(b)}}{\leqslant} \left(\int_{x_0}^{x} 1^2 \, dx\right)^{\frac{1}{2}} \left(\int_{x_0}^{x} |h'(x)|^2 \, dx\right)^{\frac{1}{2}}, \tag{67}$$

where (a) and (b) are followed by the triangle and Cauchy-Schwartz inequalities respectively. Integrating the square of both sides over the interval $[x_0, a)$ yields:

$$\int_{x_0}^{a} |h(x)|^2 \, dx \leqslant \int_{x_0}^{a} (z - x_0) \cdot \left(\int_{x_0}^{a} |h'(x)|^2 \, dx\right) dz \overset{\text{(a)}}{\leqslant} \int_{x_0}^{a} (z - x_0) \, dz \cdot \int_{-a}^{a} |h'(x)|^2 \, dx, \tag{68}$$

where (a) follows by $x_0 < a$. On the other side, we have the following for every $x \in (-a, x_0]$:

$$|h(x)| = \left|\int_{x}^{x_0} h'(x)\, dx\right| \leqslant \int_{x}^{x_0} |h'(x)| \, dx \leqslant \left(\int_{x}^{x_0} 1^2 \, dx\right)^{\frac{1}{2}} \cdot \left(\int_{x}^{x_0} |h'(x)|^2 \, dx\right)^{\frac{1}{2}}. \tag{69}$$

Therefore, we have a similar inequality:

$$\int_{-a}^{x_0} |h(x)|^2 \, dx \leqslant \int_{-a}^{x_0} (x_0 - x) \, dx \cdot \int_{-a}^{a} |h'(x)|^2 \, dx. \tag{70}$$

Using (68) and (70) completes the proof:

$$\begin{aligned}
\|h\|^2_{L^2(\Omega)} &= \int_{-a}^{x_0} |h(x)|^2 \, dx + \int_{x_0}^{a} |h(x)|^2 \, dx \\
&\leqslant \|h'\|^2_{L^2(\Omega;\mathbb{R})} \cdot \left(\int_{-a}^{x_0} (x_0 - x) \, dx + \int_{x_0}^{a} (x - x_0) \, dx \right) \\
&\leqslant \|h'\|^2_{L^2(\Omega;\mathbb{R})} \cdot \left(x_0 \left(x_0 + a\right) - \left(\frac{x_0^2 - a^2}{2} \right) + \left(\frac{a^2 - x_0^2}{2} \right) - x_0 \left(a - x_0\right) \right) \qquad (71) \\
&= \|h'\|^2_{L^2(\Omega;\mathbb{R})} \cdot \left(x_0^2 + a^2\right) \leqslant \|h'\|^2_{L^2(\Omega;\mathbb{R})} \, 2a^2. \qquad (72)
\end{aligned}$$

Thus, if x_0 exists, the proof is complete. In the next step, we prove the existence of $x_0 \in \Omega$ such that $h\left(x_0\right) = 0$. Recall that $h(t) = u_{\text{dec}}(t) - f(u_{\text{enc}}(t))$ and $u_{\text{dec}}(\cdot)$ is the solution of (4). Assume there is no such x_0. Since $h \in \mathbb{W}^{2,2}\left(\Omega;\mathbb{R}\right)$, then $h(\cdot)$ is continuous. Therefore, if there exist $t_1, t_2 \in \Omega$ such that $h(t_1) < 0$ and $h(t_2) > 0$, then the intermediate value theorem states that there exists $x_0 \in (t_1, t_2)$ such that $h(x_0) = 0$. Thus, $h(t) > 0$ or $h(t) < 0$ for all $t \in \Omega$. Without loss of generality, assume the first case where $h(t) > 0$ for all $t \in \Omega$. It means that $u_{\text{dec}}(t) > f(u_{\text{enc}}(t))$ for all $t \in \Omega$. Let us define

$$\beta^* := \underset{\beta \in \mathcal{F}}{\operatorname{argmin}} \, u_{\text{dec}}(\beta) - f(u_{\text{enc}}(\beta)).$$

Let $\bar{u}_{\text{dec}}(t) := u_{\text{dec}}(t) - u_{\text{dec}}(\beta^*)$. Note that $\int_\Omega \left(\bar{u}''_{\text{dec}}(t)\right)^2 \, dt = \int_\Omega \left(u''_{\text{dec}}(t)\right)^2 \, dt$. Therefore,

$$\begin{aligned}
\sum_{v \in \mathcal{F}} \left[u_{\text{dec}}\left(\beta_v\right) - f\left(u_{\text{enc}}\left(\beta_v\right)\right)\right]^2 &= \sum_{v \in \mathcal{F}} \left[\bar{u}_{\text{dec}}\left(\beta_v\right) + u_{\text{dec}}(\beta^*) - f\left(u_{\text{enc}}\left(\beta_v\right)\right)\right]^2 \\
&= \sum_{v \in \mathcal{F}} \left[\bar{u}_{\text{dec}}\left(\beta_v\right) - f\left(u_{\text{enc}}\left(\beta_v\right)\right)\right]^2 + |\mathcal{F}| \cdot u_{\text{dec}}(\beta^*)^2 \\
&\qquad + 2u_{\text{dec}}(\beta^*) \sum_{v \in \mathcal{F}} [\bar{u}_{\text{dec}}(\beta_v) - f(u_{\text{enc}}(\beta_v))] \\
&\overset{\text{(a)}}{\geqslant} \sum_{v \in \mathcal{F}} \left[\bar{u}_{\text{dec}}\left(\beta_v\right) - f\left(u_{\text{enc}}\left(\beta_v\right)\right)\right]^2, \qquad (73)
\end{aligned}$$

where (a) follows from $u_{\text{dec}}(\beta^*) > 0$ and $\bar{u}_{\text{dec}}(\beta_v) > f(u_{\text{enc}}(\beta_v))$ for all $v \in \mathcal{F}$. This leads to a contradiction since it implies that u_{dec} is not the solution of the (4). Therefore, our initial assumption must be wrong. Thus, there exists $x_0 \in \Omega$ such that $h(x_0) = 0$. □

Lemma 5. *Let* $\Omega = (-1, 1)$. *For* $h(t) = u_{dec}(t) - f(u_{enc}(t))$ *we have:*

$$\|h\|_{L^\infty(\Omega;\mathbb{R})} \leqslant 2 \, \|h\|^{\frac{1}{2}}_{L^2(\Omega;\mathbb{R})} \cdot \|h'\|^{\frac{1}{2}}_{L^2(\Omega;\mathbb{R})} < \infty, \tag{74}$$

and

$$\|h'\|_{L^\infty(\Omega;\mathbb{R})} \leqslant \|h'\|_{L^2(\Omega;\mathbb{R})} + \|h''\|_{L^2(\Omega;\mathbb{R})} < \infty. \tag{75}$$

Proof. Using Lemma 4 one can conclude

$$\frac{\|h\|_{L^2(\Omega;\mathbb{R})}}{\|h'\|_{L^2(\Omega;\mathbb{R})}} \leqslant \sqrt{2}. \tag{76}$$

Since $h \in \mathbb{W}^{2,2}\left(\Omega;\mathbb{R}\right)$ we can apply Corollary 1 and Theorem 5 with $r = \infty$ and $p, q = 2$ to complete the proof of (74). Furthermore, using Theorem 5 with $r = \infty$ and $p, q = 2$ and $\ell = 1$ completes the proof of (75). □

https://doi.org/10.52202/079017-3554

Building upon Lemma 5 and starting from (58), we can derive an upper bound for $\mathcal{L}_{\text{dec}}$:

$$\begin{aligned}
\mathcal{L}_{\text{dec}}(u_{\text{dec}}) &= \mathbb{E}_{\epsilon,\mathcal{F}}\left[\frac{2}{K}\sum_{k=1}^{K}\left(u_{\text{dec}}(\alpha_k) - f(u_{\text{enc}}(\alpha_k))\right)_2^2\right] \\
&\overset{\text{(a)}}{=} \frac{2}{K}\sum_{k=1}^{K}\mathbb{E}_{\epsilon,\mathcal{F}}\left[h(\alpha_k)^2\right] \\
&\overset{\text{(b)}}{\leqslant} \frac{2}{K}\sum_{k=1}^{K}\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h\|^2_{L^\infty(\Omega;\mathbb{R})}\right] \\
&\overset{\text{(c)}}{\leqslant} 2\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h\|_{L^2(\Omega;\mathbb{R})}\cdot\|h'\|_{L^2(\Omega;\mathbb{R})}\right] \\
&\overset{\text{(d)}}{\leqslant} 2\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h\|^2_{L^2(\Omega;\mathbb{R})}\right]^{\frac{1}{2}}\cdot\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h'\|^2_{L^2(\Omega;\mathbb{R})}\right]^{\frac{1}{2}},
\end{aligned} \tag{77}$$

where (a) follows from the definition of $h(t)$, (b) is due to the fact that $h(t) \leqslant \|h\|_{L^\infty(\Omega;\mathbb{R})}$ for $t \in \Omega$, (c) follows by applying Lemma 5, and (d) is a result of applying the Cauchy-Schwartz inequality.

B.1 Proof of Theorem 1

Proof. As previously mentioned, $u_{\text{dec}}(\cdot)$ is a second-order smoothing spline fitted on the data points $\left\{(\beta_{i_1}, f(u_{\text{enc}}(\beta_{i_1})), \ldots, (\beta_{i_{|\mathcal{F}|}}, f(u_{\text{enc}}(\beta_{i_{|\mathcal{F}|}}))\right\}$, where $\mathcal{F} := \left\{\beta_{i_1}, \ldots, \beta_{i_{|\mathcal{F}|}}\right\}$ represents the set of non-straggler worker nodes, and $\boldsymbol{f} \circ \boldsymbol{u}_{\text{enc}} := \left\{f(u_{\text{enc}}(\beta_{i_1})), \ldots, f(u_{\text{enc}}(\beta_{i_{|\mathcal{F}|}}))\right\}$ is the corresponding set of computation results from these non-straggler workers. By the definition given in (50), $\mathbf{S}_{\lambda_{\text{d}},|\mathcal{F}|,2}(\cdot)$ denotes the smoothing spline operator for the decoder layer. Hence, we have the following:

$$\mathbb{E}_{\mathcal{F}}\left[\|h\|^2_{L^2(\Omega;\mathbb{R})}\right] = \mathbb{E}_{\mathcal{F}}\left[\left\|f \circ u_{\text{enc}} - \mathbf{S}_{\lambda_{\text{d}},|\mathcal{F}|,2}[\mathbf{f}]\right\|^2_{L^2(\Omega;\mathbb{R})}\right], \tag{78}$$

where $\mathbf{f} = \left\{f(u_{\text{enc}}(\beta_{i_1})), \ldots, f(u_{\text{enc}}(\beta_{i_{|\mathcal{F}|}}))\right\}$. Let us define the following variables analogous to those in (6):

$$\Delta^{\mathcal{F}}_{\max} := \max_{f\in\{0,\ldots,|\mathcal{F}|\}}\left\{\beta_{i_{f+1}} - \beta_{i_f}\right\}, \quad \Delta^{\mathcal{F}}_{\min} := \min_{f\in\{1,\ldots,|\mathcal{F}|-1\}}\left\{\beta_{i_{f+1}} - \beta_{i_f}\right\}. \tag{79}$$

Since there are at most S stragglers among the worker nodes, we have $\Delta^{\mathcal{F}}_{\max} \leqslant (S+1)\cdot\Delta_{\max}$ and $\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}} \leqslant (S+1)\cdot\frac{\Delta_{\max}}{\Delta_{\min}} \leqslant (S+1)B$. Additionally, because $\Delta_{\min} \leqslant \frac{2}{N}$, there exists a constant J such that $\Delta_{\max} \leqslant \frac{J}{N}$, and consequently, $\Delta^{\mathcal{F}}_{\max} \leqslant \frac{J(S+1)}{N} \leqslant \frac{J(S+1)}{N-S}$. Otherwise, this would contradict the condition $\frac{\Delta_{\max}}{\Delta_{\min}} \leqslant B$.

Applying Theorem 6 with $\Omega = (-1,1), m = 2$, we have

$$\mathbb{E}_{\mathcal{F}}\left[\|h\|^2_{L^2(\Omega;\mathbb{R})}\right] \leqslant H_0\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\cdot\mathbb{E}_{\mathcal{F}}\left[L\right], \tag{80}$$

and

$$\mathbb{E}_{\mathcal{F}}\left[\|h'\|^2_{L^2(\Omega;\mathbb{R})}\right] \leqslant H_1\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\cdot\mathbb{E}_{\mathcal{F}}\left[L^{\frac{1}{2}}(1+\frac{L}{16})^{\frac{1}{2}}\right], \tag{81}$$

where $L = p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\cdot\frac{(N-S)\Delta^{\mathcal{F}}_{\max}}{4}\lambda_{\text{d}} + D(2)\cdot\left(\Delta^{\mathcal{F}}_{\max}\right)^4$ and $H_0, H_1 := H(2,0), H(2,1)$ as defined in Theorem 6. Thus, we have:

$$\begin{aligned}
\mathbb{E}_{\mathcal{F}}[L] &\leqslant \mathbb{E}_{\mathcal{F}}\left[p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\cdot\frac{(N-S)\Delta^{\mathcal{F}}_{\max}}{4}\lambda_{\text{d}} + D\cdot\left(\Delta^{\mathcal{F}}_{\max}\right)^4\right] \\
&\overset{\text{(a)}}{\leqslant} \mathbb{E}_{\mathcal{F}}\left[p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\right]\cdot\frac{J(S+1)}{4}\lambda_{\text{d}} + D\cdot\mathbb{E}_{\mathcal{F}}\left[\left(\Delta^{\mathcal{F}}_{\max}\right)^4\right] \qquad (82) \\
&\overset{\text{(b)}}{\leqslant} \mathbb{E}_{\mathcal{F}}\left[p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\right]\cdot\frac{J(S+1)}{4}\lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{N^4}, \qquad (83)
\end{aligned}$$

where $D := D(2)$ and (a) and (b) follow from $\Delta^{\mathcal{F}}_{\max} \leqslant \frac{J(S+1)}{N-S}$ and $\Delta^{\mathcal{F}}_{\max} \leqslant \frac{J(S+1)}{N}$ respectively. Substituting in (91),(92), and (92), we have:

$$\mathbb{E}_{\mathcal{F}}\left[\|h\|^2_{L^2(\Omega;\mathbb{R})}\right] \leqslant H_0 \left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \cdot \left(\mathbb{E}_{\mathcal{F}}\left[p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\right]\cdot\frac{J(S+1)}{4}\lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{N^4}\right), \tag{84}$$

$$\mathbb{E}_{\mathcal{F}}\left[\|h'\|^2_{L^2(\Omega;\mathbb{R})}\right] \leqslant H_1 \left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \cdot \left(\mathbb{E}_{\mathcal{F}}\left[p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\right]\cdot\frac{J(S+1)}{4}\lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{N^4}\right)^{\frac{1}{2}} \cdot \left(1+\frac{\mathbb{E}_{\mathcal{F}}\left[p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\right]\cdot\frac{J(S+1)}{4}\lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{N^4}}{16}\right)^{\frac{1}{2}}. \tag{85}$$

Therefore, we can derive an upper bound for (77) based on the above inequalities. This upper bound holds for any λ_{d}. Since $\lambda_{\text{d}} \leqslant N^{-4}$ and $\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}} \leqslant (S+1)B$, we have:

$$\begin{aligned}\mathbb{E}_{\mathcal{F}}\left[p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right)\right]\cdot\frac{J(S+1)}{4}\lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{N^4} &\leqslant \tilde{p}_3\left(S+1\right)N^{-4} + DJ^4\frac{(S+1)^4}{N^4}\\ &\overset{\text{(a)}}{\leqslant} \widetilde{D}\frac{(S+1)^4}{N^4},\end{aligned} \tag{86}$$

where $\tilde{p}_3$ is a degree-3 polynomial in $(S+1)$ with positive constant coefficients, and $\widetilde{D}$ is the sum of the coefficients of $\tilde{p}_3$ and DJ^4. Therefore, we have:

$$\begin{aligned}\mathcal{L}_{\text{dec}} &\leqslant \left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\left[\left(H_0\cdot r(S,N)\right)^{\frac{1}{2}}\left(H_1\cdot r(S,N)^{\frac{1}{2}}\left(1+\frac{r(S,N)}{16}\right)^{\frac{1}{2}}\right)^{\frac{1}{2}}\right]\\ &= \left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\left[H_0^{\frac{1}{2}}H_1^{\frac{1}{2}}\cdot r(S,N)^{\frac{3}{4}}\cdot\left(1+\frac{r(S,N)}{16}\right)^{\frac{1}{4}}\right],\end{aligned} \tag{87}$$

where $r(S,N) := \widetilde{D}\frac{(S+1)^4}{N^4}$. Note that, since $S+1 \leqslant N$ then $1+\frac{r(S,N)}{16} \leqslant 2\max(1,\widetilde{D})$. Defining $\eta := 2\max(1,\widetilde{D})$, we have:

$$\mathcal{L}_{\text{dec}} \leqslant \left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\left[H_0^{\frac{1}{2}}H_1^{\frac{1}{2}}\eta^{\frac{1}{4}}\cdot r(S,N)^{\frac{3}{4}}\right], \tag{88}$$

Defining $C := H_0^{\frac{1}{2}}H_1^{\frac{1}{2}}\eta^{\frac{1}{4}}$ and applying Lemma 2, completes the proof. □

B.2 Proof of Theorem 2

Using the decomposition (56) we have:

$$\begin{aligned}\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h\|^2_{L^2(\Omega;\mathbb{R})}\right] &= \mathbb{E}_{\mathcal{F}}\left[\left\|f\circ u_{\text{enc}} - \mathbf{S}_{\lambda_{\text{d}},|\mathcal{F}|,2}[\mathbf{f}]\right\|^2_{L^2(\Omega;\mathbb{R})}\right]\\ &\quad + \mathbb{E}_{\epsilon,\mathcal{F}}\left[\left\|f\circ u_{\text{enc}} - \mathbf{S}_{\lambda_{\text{d}},|\mathcal{F}|,2}[\boldsymbol{\epsilon}]\right\|^2_{L^2(\Omega;\mathbb{R})}\right],\end{aligned} \tag{89}$$

where $\mathbf{f} = \left\{f(u_{\text{enc}}(\beta_{i_1})),\ldots,f(u_{\text{enc}}(\beta_{i_{|\mathcal{F}|}}))\right\}$ and $\boldsymbol{\epsilon} = \left\{\epsilon_{i_1},\ldots,\epsilon_{i_{|\mathcal{F}|}}\right\}$. Same as (79) we define the following variables:

$$\Delta^{\mathcal{F}}_{\max} := \max_{f\in\{0,\ldots,|\mathcal{F}|\}}\left\{\beta_{i_{f+1}}-\beta_{i_f}\right\}, \quad \Delta^{\mathcal{F}}_{\min} := \min_{f\in\{1,\ldots,|\mathcal{F}|-1\}}\left\{\beta_{i_{f+1}}-\beta_{i_f}\right\}. \tag{90}$$

Again we have $\Delta^{\mathcal{F}}_{\max} \leqslant (S+1)\cdot\Delta_{\max}$ and $\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}} \leqslant (S+1)\cdot\frac{\Delta_{\max}}{\Delta_{\min}} \overset{\text{(a)}}{\leqslant} (S+1)B$, where (a) is because of the bounded condition that we have. Therefore, $\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}$ is bounded. Additionally, since $\Delta_{\min} \leqslant \frac{2}{N}$,

https://doi.org/10.52202/079017-3554

then $\frac{\Delta_{\max}}{\Delta_{\min}} \leqslant B$ implies that both $\Delta_{\max} = \mathcal{O}(\frac{1}{N})$. Thus, there exists a constant J such that $\Delta_{\max} \leqslant \frac{J}{N}$. Therefore, we have $\Delta^{\mathcal{F}}_{\max} \leqslant \Delta_{\max} \cdot (S+1) \leqslant \frac{J(S+1)}{N} \leqslant \frac{J(S+1)}{N-S}$. Applying Theorems 6 and 7 with $\Omega = (-1,1), m = 2$, we have:

$$\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h\|^2_{L^2(\Omega;\mathbb{R})}\right] \leqslant H_0 \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \cdot \mathbb{E}_{\mathcal{F}}[L] + \frac{Q_0\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{1}{4}}, \tag{91}$$

and

$$\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h'\|^2_{L^2(\Omega;\mathbb{R})}\right] \leqslant H_1 \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \cdot \mathbb{E}_{\mathcal{F}}\left[L^{\frac{1}{2}}(1+\frac{L}{16})^{\frac{1}{2}}\right] + \frac{Q_1\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{3}{4}}, \tag{92}$$

where $L = p_2\left(\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}}\right) \cdot \frac{(N-S)\Delta^{\mathcal{F}}_{\max}}{4}\lambda_{\text{d}} + D(2) \cdot \left(\Delta^{\mathcal{F}}_{\max}\right)^4$, $H_0, H_1 := H(2,0), H(2,1)$, and $Q_0(\lambda_0), Q_1(\lambda_0) := Q(2,0,\lambda_0), Q(2,1,\lambda_0)$ as defined in Theorems 6 and 7. Since $\frac{\Delta^{\mathcal{F}}_{\max}}{\Delta^{\mathcal{F}}_{\min}} \leqslant (S+1)B$, we have:

$$\begin{aligned}
\mathbb{E}_{\mathcal{F}}[L] &\leqslant \mathbb{E}_{\mathcal{F}}\left[\tilde{p_2}(S+1) \cdot \frac{(N-S)\Delta^{\mathcal{F}}_{\max}}{4}\lambda_{\text{d}} + D \cdot \left(\Delta^{\mathcal{F}}_{\max}\right)^4\right] \\
&\overset{(a)}{\leqslant} \tilde{p_2}(S+1) \cdot \frac{J(S+1)}{4}\lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{(N-S)^4} \\
&= p_3(S+1) \cdot \lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{(N-S)^4},
\end{aligned} \tag{93}$$

where $D := D(2), \tilde{p_2}(S+1) = p_2\left(B(S+1)\right), p_3(S+1) := \tilde{p_2}(S+1) \cdot \frac{J(S+1)}{4}$ is a degree three polynomial of $(S+1)$, and (a) follows from the $\Delta^{\mathcal{F}}_{\max} \leqslant \frac{J(S+1)}{N-S}$. Substituting in (91), we have:

$$\begin{aligned}
\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h\|^2_{L^2(\Omega;\mathbb{R})}\right] &\leqslant H_0 \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \left(p_3(S+1) \cdot \lambda_{\text{d}} + DJ^4\frac{(S+1)^4}{(N-S)^4}\right) + \frac{Q_0(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{1}{4}} \\
&\overset{(a)}{\leqslant} H_0 \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \lambda_{\text{d}} \cdot \left(p_3(S+1) + DJ^4(S+1)^4\right) + \frac{Q_0(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{1}{4}} \\
&\overset{(b)}{=} H_0 \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \lambda_{\text{d}} \cdot p_4(S+1) + \frac{Q_0(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{1}{4}},
\end{aligned} \tag{94}$$

where (a) follows from the fact that $\lambda_{\text{d}}^{-1}(N-S)^{-4} \leqslant 1$, as assumed in Theorem 7 and (b) is by definition $p_4(S+1) := p_3(S+1) + DJ^4(S+1)^4$ is a degree four polynomial of $(S+1)$. An analogous upper bound can be derived for $\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h'\|^2_{L^2(\Omega;\mathbb{R})}\right]$ as

$$\begin{aligned}
\mathbb{E}_{\epsilon,\mathcal{F}}\left[\|h'\|^2_{L^2(\Omega;\mathbb{R})}\right] &\leqslant H_1 \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \cdot \lambda_{\text{d}}^{\frac{1}{2}} \cdot p_4(S+1)^{\frac{1}{2}} \cdot \left(1 + \lambda_{\text{d}}\frac{p_4(S+1)}{16}\right)^{\frac{1}{2}} \\
&\qquad + \frac{Q_1(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{3}{4}} \\
&\overset{(a)}{\leq} H_1 \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \cdot \lambda_{\text{d}}^{\frac{1}{2}} \cdot p_4(S+1)^{\frac{1}{2}} \cdot \left(1 + \lambda_0\frac{p_4(S+1)}{16}\right)^{\frac{1}{2}} \\
&\qquad + \frac{Q_1(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{3}{4}} \\
&\overset{(b)}{=} \left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})} \tilde{\eta} \cdot \lambda_{\text{d}}^{\frac{1}{2}} \cdot p_4(S+1)^{\frac{1}{2}} \quad + \frac{Q_1(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{3}{4}},
\end{aligned} \tag{95}$$

where (a) follows from the definition $\lambda_{\text{d}} \leqslant \lambda_0$, (b) is derived from the definition $\tilde{\eta} := H_1\left(1 + \lambda_0\frac{p_4(S+1)}{16}\right)^{\frac{1}{2}}$. By applying the upper bound for $\mathcal{L}_{\text{dec}}$ from (77) and incorporating the

results from (94) and (95), we can deduce the following:

$$\begin{aligned}\mathcal{L}_{\text{dec}} &\overset{\text{(a)}}{\leqslant} 2\left(\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\cdot\mu_0(S)\lambda_{\text{d}}+\frac{Q_0(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{1}{4}}\right)^{\frac{1}{2}}\\ &\quad\cdot\left(\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\cdot\mu_1(S)\lambda_{\text{d}}^{\frac{1}{2}}+\frac{Q_1(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{3}{4}}\right)^{\frac{1}{2}}\\ &\overset{\text{(b)}}{\leqslant} 2\lambda_{\text{d}}^{\frac{1}{4}}\left(\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\cdot\mu_{\max}(S)\lambda_{\text{d}}^{\frac{1}{2}}+\frac{Q_{\max}(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{3}{4}}\right)\\ &= 2\lambda_{\text{d}}^{\frac{3}{4}}\cdot\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}\cdot\mu_{\max}(S)+\frac{Q_{\max}(\lambda_0)\sigma_0^2}{N-S}\lambda_{\text{d}}^{-\frac{1}{2}},\end{aligned}\tag{96}$$

where (a) follows by the definitions $\mu_0(S):=H_0\cdot p_4(S+1)$ and $\mu_1(S):=\tilde{\eta}\cdot p_4(S+1)^{\frac{1}{2}}$, and (b) is due to $Q_0(\lambda_0),Q_1(\lambda_0)\leqslant Q_{\max}(\lambda_0):=\max\{Q_0(\lambda_0),Q_1(\lambda_0)\}$ and $\mu_0(S),\mu_1(S)\leqslant\mu_{\max}(S):=\max\{\mu_0(S),\mu_1(S)\}$. Therefore, we can conclude that:

$$\mathcal{L}_{\text{dec}}\leqslant 2\lambda_{\text{d}}^{\frac{3}{4}}\cdot\mu_{\max}(S)\cdot\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}+\lambda_{\text{d}}^{-\frac{1}{2}}\cdot\frac{Q_{\max}(\lambda_0)\sigma_0^2}{N-S}.\tag{97}$$

Based on the definition of $\mu_{\max}(S)$ we have:

$$\begin{aligned}\mu_{\max}(S)&=\max\left\{H_0\cdot p_4(S),H_1\left(1+\lambda_0\frac{p_4(S)}{16}\right)^{\frac{1}{2}}p_4(S)^{\frac{1}{2}}\right\}\\ &\leqslant H_{\max}\cdot p_4(S)^{\frac{1}{2}}\cdot\max\left\{p_4(S),1+\frac{\lambda_0 p_4(S)}{16}\right\}^{\frac{1}{2}},\\ &\leqslant H_{\max}\cdot p_4(S)^{\frac{1}{2}}\cdot\left(\frac{1+p_4(S)}{16}\right)^{\frac{1}{2}}\max\{16,\lambda_0\}^{\frac{1}{2}},\\ &= H_{\max}\cdot\widetilde{p_4}(S)\max\{4,\lambda_0\}^{\frac{1}{2}},\end{aligned}\tag{98}$$

where $H_{\max}:=\max\{H_0,H_1\}$ and $\widetilde{p_4}(S):=p_4(S)^{\frac{1}{2}}\cdot\left(\frac{1+p_4(S)}{16}\right)^{\frac{1}{2}}$. Based on definition of $Q_{\max}(\lambda_0)$ (mentioned in Theorem 7), we have:

$$\begin{aligned}Q_{\max}(\lambda_0)&=\max\{w_0\lambda_0^{\frac{1}{4}}+\tilde{w}_0,w_1\lambda_0^{\frac{1}{4}}+\tilde{w}_1\}\\ &\leqslant w_{\max}\lambda_0^{\frac{1}{4}}+\tilde{w}_{\max},\\ &\leqslant 2w_{\max}\max\left\{\lambda_0,\left(\frac{\tilde{w}_{\max}}{w_{\max}}\right)^4\right\}^{\frac{1}{4}}\end{aligned}\tag{99}$$

where $w_{\max}:=\max\{w_0,w_1\}$ and $\tilde{w}_{\max}:=\max\{\tilde{w}_0,\tilde{w}_1\}$. Therefore we have:

$$\mathcal{L}_{\text{dec}}\leqslant 2\lambda_{\text{d}}^{\frac{3}{4}}\cdot H_{\max}\cdot\widetilde{p_4}(S)\max\{4,\lambda_0\}^{\frac{1}{2}}\cdot\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}+\lambda_{\text{d}}^{-\frac{1}{2}}\frac{2\sigma_0^2 w_{\max}\max\left\{\lambda_0^{\frac{1}{4}},\frac{\tilde{w}_{\max}}{w_{\max}}\right\}}{N-S}.\tag{100}$$

Since (97) holds for all λ_{d}, by minimizing the right-hand side of (97) with respect to λ_{d}, we obtain:

$$\lambda_{\text{d}}^*=\left(\frac{3H_{\max}\cdot\widetilde{p_4}(S)\cdot(N-S)\cdot\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}}{4w_{\max}\sigma_0^2}\right)^{-\frac{4}{5}}\left(\frac{\max\{4,\lambda_0\}^{\frac{1}{2}}}{\max\left\{\lambda_0^{\frac{1}{4}},\frac{\tilde{w}_{\max}}{w_{\max}}\right\}}\right)^{-\frac{4}{5}}\tag{101}$$

By substituting the expression for λ_{d}^* into (97), we have:

$$\mathcal{L}_{\text{dec}}\leqslant 4\left(\frac{3H_{\max}}{4w_{\max}}\right)^{\frac{2}{5}}\left(\frac{\sigma_0}{N-S}\right)^{\frac{3}{5}}\cdot\widetilde{p_4}(S)^{\frac{2}{5}}\cdot\left\|(f\circ u_{\text{enc}})^{(2)}\right\|^{\frac{4}{5}}_{L^2(\Omega;\mathbb{R})}\left(\frac{\max\{4,\lambda_0\}^{\frac{1}{2}}}{\max\left\{\lambda_0^{\frac{1}{4}},\frac{\tilde{w}_{\max}}{w_{\max}}\right\}}\right)^{\frac{2}{5}}.\tag{102}$$

https://doi.org/10.52202/079017-3554

Thus, defining $C_2 := \frac{3H_{\max}}{4w_{\max}}, C(\lambda_0) := \left(\frac{\max\{4,\lambda_0\}^{\frac{1}{2}}}{\max\left\{\lambda_0^{\frac{1}{4}}, \frac{\tilde{w}_{\max}}{w_{\max}}\right\}} \right)$, and using previously driven upper bound for $\mathcal{L}_{\text{enc}}$ in Lemma 2 completes the proof.

B.3 Proof of Theorem 3

The upper bounds presented in Theorems 1 and 2 depend on $\left\|(f \circ u_{\text{enc}})^{(2)}\right\|^2_{L^2(\Omega;\mathbb{R})}$, with exponents of $\frac{2}{5}$ and 1, respectively. By applying the chain rule, we can demonstrate that:

$$
\begin{aligned}
\int_\Omega \left[f(u_{\text{enc}}(t))''\right]^2 \, dt &\overset{\text{(a)}}{=} \int_\Omega \left[u''_{\text{enc}}(t) \cdot f'(u_{\text{enc}}(t)) + u'_{\text{enc}}(t)^2 \cdot f''(u_{\text{enc}}(t))\right]^2 \, dt \\
&\overset{\text{(b)}}{\leqslant} \int_\Omega \left[u''_{\text{enc}}(t)^2 + u'_{\text{enc}}(t)^4\right] \left[f'(u_{\text{enc}}(t))^2 + f''(u_{\text{enc}}(t))^2\right] \, dt \\
&\overset{\text{(c)}}{\leqslant} (q^2+\nu^2) \int_\Omega \left[u''_{\text{enc}}(t)^2 + u'_{\text{enc}}(t)^4\right] \, dt \\
&= (q^2+\nu^2) \left(\|u''_{\text{enc}}(t)\|^2_{L^2(\Omega;\mathbb{R})} + \|u'_{\text{enc}}(t)\|^4_{L^4(\Omega;\mathbb{R})}\right) \\
&\overset{\text{(d)}}{\leqslant} (q^2+\nu^2) \left(\|u''_{\text{enc}}(t)\|^2_{L^2(\Omega;\mathbb{R})} + \left(\|u'_{\text{enc}}(t)\|_{L^2(\Omega;\mathbb{R})} + \|u''_{\text{enc}}(t)\|_{L^2(\Omega;\mathbb{R})}\right)^4\right) \\
&\overset{\text{(e)}}{\leqslant} (q^2+\nu^2) \left(\|u''_{\text{enc}}(t)\|^2_{L^2(\Omega;\mathbb{R})} + 4\left(\|u'_{\text{enc}}(t)\|^2_{L^2(\Omega;\mathbb{R})} + \|u''_{\text{enc}}(t)\|^2_{L^2(\Omega;\mathbb{R})}\right)^2\right) \\
&\overset{\text{(f)}}{\leqslant} (q^2+\nu^2) \left(\|u_{\text{enc}}\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})} + 4\|u_{\text{enc}}\|^4_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}\right) \\
&\overset{\text{(g)}}{\leqslant} (q^2+\nu^2) \left(73\|u_{\text{enc}}\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})} + 4\times 73^2\|u_{\text{enc}}\|^4_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}\right) \\
&\overset{\text{(h)}}{=} (q^2+\nu^2)\cdot\psi\left(\|u_{\text{enc}}\|^2_{\widetilde{\mathbb{W}}^{2,2}(\Omega;\mathbb{R})}\right), \\
&\overset{\text{(i)}}{=} (q^2+\nu^2)\cdot\psi\left(\|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}\right),
\end{aligned} \tag{103}
$$

where (a) follows from the chain rule, (b) is derived using the Cauchy-Schwartz inequality, (c) is due to $\|f''_0\|_{L^\infty(\Omega;\mathbb{R})} \leqslant \nu$, (d) follows from Theorem 5 with $r=4, p=q=2, l=1$, (e) follows from AM-GM inequality, (f) is a result of adding positive terms $\|u_{\text{enc}}\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}$ and $\|u'_{\text{enc}}\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}$ to the first term and $\|u_{\text{enc}}\|^2_{\mathbb{W}^{2,2}(\Omega;\mathbb{R})}$ to the second term in the parenthesis, (g) is result of applying Corollary 4, (h) is by defining $\psi(t) := 73t + 4\times 73^2 t^2$, and (i) is because of Proposition 2.

Combining (103) with Theorems 1 and 2 we have

$$
\mathcal{R}(\hat{f}) \leqslant \frac{2q^2}{K}\sum_{k=1}^{K}(u_{\text{enc}}(\alpha_k)-x_k)^2 + \lambda_{\text{e}}\cdot\psi\left(\|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}\right), \tag{104}
$$

for the noiseless setting and

$$
\mathcal{R}(\hat{f}) \leqslant \frac{2q^2}{K}\sum_{k=1}^{K}(u_{\text{enc}}(\alpha_k)-x_k)^2 + \widetilde{\lambda}_{\text{e}}\cdot\psi\left(\|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}\right)^{\frac{2}{5}} \tag{105}
$$

for the noisy setting, where the parameters λ_{e} and $\widetilde{\lambda}_{\text{e}}$ are as follows:

$$
\lambda_{\text{e}} = C_1\frac{(S+1)^3}{N^3}\cdot(q^2+\nu^2) \tag{106}
$$

$$
\widetilde{\lambda}_{\text{e}} = 2C(\lambda_0)^{\frac{3}{4}}\left(\frac{\sigma_0^2}{N-S}\right)^{\frac{3}{5}}\cdot p_4(S)^{\frac{2}{5}}\cdot(q^2+\nu^2)^{\frac{2}{5}}. \tag{107}
$$

Note that since $\psi(t)$ and $\gamma(t) := t^{\frac{2}{5}}$ are monotonically increasing in $\mathbb{R}^+$, its composition is monotonically increasing as well. Moreover, λ_{e} and $\widetilde{\lambda}_{\text{e}}$ share the same exponent of N as in Theorems 1 and 2, respectively. Consequently, the provided upper bound does not compromise the convergence rate.

B.4 Proof of Proposition 1

For part (i), we know that for $t \leqslant M$, we have:

$$\psi(t) = 73t + 4 \times 73^2 t^2 \leqslant t(73 + 4 \times 73^2 t) \leqslant t(73 + 4 \times 73^2 \cdot M) = t(m_1 + m_2 M), \quad (108)$$

where $m_1 := 73, m_2 := 4 \times 73^2$. Therefore, if $\|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} \leqslant M$, then:

$$\begin{aligned}
\psi(\|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}) &\leqslant (m_1 + m_2 M) \|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} \\
&= (m_1 + m_2 M) \left(u_{\text{enc}}(-1)^2 + u'_{\text{enc}}(-1)^2 + \int_\Omega |u''_{\text{enc}}(t)|^2 \, dt \right) \\
&\overset{\text{(a)}}{\leqslant} (m_1 + m_2 M) \left(M + \int_\Omega |u''_{\text{enc}}(t)|^2 \, dt \right), \quad (109)
\end{aligned}$$

where (a) is because of $\|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} \leqslant M$. Thus, we have:

$$\begin{aligned}
\mathcal{R}(\hat{f}) &\leqslant \frac{2q^2}{K} \sum_{k=1}^{K} (u_{\text{enc}}(\alpha_k) - x_k)^2 + \lambda_{\text{e}} \cdot \psi \left(\|u_{\text{enc}}\|^2_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} \right) \\
&\leqslant \frac{2q^2}{K} \sum_{k=1}^{K} (u_{\text{enc}}(\alpha_k) - x_k)^2 + \lambda_{\text{e}} \cdot (m_1 + m_2 M) \left(M + \int_\Omega |u''_{\text{enc}}(t)|^2 \, dt \right) \\
&\leqslant \widetilde{R}(u_{\text{enc}}). \quad (110)
\end{aligned}$$

For part (ii), let $\widetilde{u_{\text{enc}}}(t)$ be a natural spline used as the encoder function fitted to the data points $\{(\alpha_k, x_k)\}_{k=1}^K$. Then, we have:

$$\begin{aligned}
\lambda_{\text{e}}(m_1 + m_2 M) \left(M + \int_\Omega |(u^*)''(t)|^2 \, dt \right) &\leqslant \widetilde{R}(u^*) \\
&\overset{\text{(a)}}{\leqslant} \widetilde{R}(u_{\text{enc}}) \overset{\text{(b)}}{=} \lambda_{\text{e}}(m_1 + m_2 M) \left(M + \int_\Omega |\widetilde{u_{\text{enc}}}''(t)|^2 \, dt \right), \quad (111)
\end{aligned}$$

where (a) follows from the optimality of $u^*(\cdot)$, and (b) follows from $\widetilde{u_{\text{enc}}}(\alpha_k) = x_k$ for $k \in [K]$. Therefore, $\int_\Omega |(u^*)''(t)|^2 \, dt \leqslant \int_\Omega |\widetilde{u_{\text{enc}}}''(t)|^2 \, dt$.

Since $u^*(\cdot)$ is smoothing spline, it has the representation in natural spline space (as mentioned in (51)):

$$u^*(t) = \sum_{k=1}^{K+4} \xi_k b_k(t), \quad (112)$$

where, $\{b_k(\cdot)\}_{k=1}^K$ is a basis functions of second order natural splines. Therefore, using Cauchy-Schwartz inequality, we have:

$$|u^*(t)|^2 \leqslant \left(\sum_{k=1}^{K+4} \xi^2 \right) \left(\sum_{k=1}^{K+4} |b_k(t)|^2 \right), \quad (113)$$

and

$$|(u^*(t))'|^2 \leqslant \left(\sum_{k=1}^{K+4} \xi^2 \right) \left(\sum_{k=1}^{K+4} |b'_k(t)|^2 \right). \quad (114)$$

Both (113) and (114) hold for all $t \in \Omega$. Thus:

$$|(u^*)'(-1)|^2 \leqslant \|u^*\|^2_{L^\infty(\Omega;\mathbb{R})} \leqslant \|\boldsymbol{\xi}\|_2^2 \cdot \left(\sum_{k=1}^{K+4} \|b_k\|^2_{L^\infty(\Omega;\mathbb{R})} \right), \quad (115)$$

$$|(u^*)'(-1)|^2 \leqslant \|(u^*)'\|^2_{L^\infty(\Omega;\mathbb{R})} \leqslant \|\boldsymbol{\xi}\|_2^2 \cdot \left(\sum_{k=1}^{K+4} \|b'_k\|^2_{L^\infty(\Omega;\mathbb{R})} \right), \quad (116)$$

https://doi.org/10.52202/079017-3554

where $\boldsymbol{\xi} := [\xi_1, \ldots, \xi_{K+4}]^T = \left(\mathbf{N}^T\mathbf{N} + \lambda\Phi\right)^{-1}\mathbf{N}^T\mathbf{x}$ with $\mathbf{N}_{ij} = b_i(\alpha_j), \Phi_{ij} = \int_\Omega b_i''(t)b_j''(t)\,dt$ for $i, j \in [K+4]$ as defined in (51), and $\mathbf{x} := [x_1, \ldots, x_K]$. Noted that $\{\|b_k'\|^2_{L^\infty(\Omega;\mathbb{R})}\}_{k=1}^{K+4}$ and $\{\|b_k\|^2_{L^\infty(\Omega;\mathbb{R})}\}_{k=1}^{K+4}$ depend only on $\{\alpha_k\}_{k=1}^K$.

Lemma 6. *If* $\widetilde{\boldsymbol{\xi}} := \left(\mathbf{N}^T\mathbf{N}\right)^{-1}\mathbf{N}^T\mathbf{x}$*, then* $\|\boldsymbol{\xi}\|_2^2 \leqslant \kappa(\Phi)\cdot\left\|\widetilde{\boldsymbol{\xi}}\right\|_2^2 < \frac{2}{|\det\Phi|}\left(\frac{\|\Phi\|_F^2}{K}\right)^{\frac{K}{2}}\left\|\widetilde{\boldsymbol{\xi}}\right\|_2^2$*, where* $\kappa(\Phi)$ *is condition number of* Φ*.*

Proof. By defining $\widetilde{\mathbf{N}} := \mathbf{N}\Phi^{-\frac{1}{2}}$ and rearranging the expression for $\widetilde{\boldsymbol{\xi}}$, we obtain:

$$\begin{aligned}\widetilde{\boldsymbol{\xi}} = \left(\mathbf{N}^T\mathbf{N} + \lambda\Phi\right)^{-1}\mathbf{N}^T\mathbf{x} &= \Phi^{-1/2}\left(\left[\mathbf{N}\Phi^{-1/2}\right]^T\left[\mathbf{N}\Phi^{-1/2}\right] + \lambda\mathbf{I}\right)^{-1}\left[\mathbf{N}\Phi^{-1/2}\right]^T\mathbf{x}\\ &= \Phi^{-1/2}\left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}} + \lambda\mathbf{I}\right)^{-1}\widetilde{\mathbf{N}}^T\mathbf{x}. \end{aligned} \tag{117}$$

Define $\mathbf{z} := \left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}}\right)^{-1}\widetilde{\mathbf{N}}^T\mathbf{x}$. Thus, $\widetilde{\mathbf{N}}^T\mathbf{x} = \widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}}\mathbf{z}$. Thus, by applying the Cauchy-Schwartz inequality, we have:

$$\left\|\left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}} + \lambda\mathbf{I}\right)^{-1}\widetilde{\mathbf{N}}^T\mathbf{x}\right\|_2 = \left\|\left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}} + \lambda\mathbf{I}\right)^{-1}\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}}\mathbf{z}\right\|_2 \leqslant \left\|\left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}} + \lambda\mathbf{I}\right)^{-1}\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}}\right\|_2\cdot\|\mathbf{z}\|_2. \tag{118}$$

Let $\widetilde{\mathbf{N}} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ be the singular value decomposition of $\widetilde{\mathbf{N}}$. Therefore, we have:

$$\begin{aligned}\left\|\left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}} + \lambda\mathbf{I}\right)^{-1}\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}}\right\|_2 &= \left\|\left(\mathbf{V}\mathbf{D}^2\mathbf{V}^T + \lambda\mathbf{I}\right)^{-1}\mathbf{V}\mathbf{D}^2\mathbf{V}^T\right\|_2\\ &= \left\|\mathbf{V}^{-T}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{V}^{-1}\mathbf{V}\mathbf{D}^2\mathbf{V}^T\right\|_2\\ &= \left\|\mathbf{V}^{-T}\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^2\mathbf{V}^T\right\|_2\\ &\overset{(a)}{=} \left\|\left(\mathbf{D}^2 + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^2\right\|_2\\ &= \left\|\operatorname{diag}\left(\frac{\lambda_1^2}{\lambda_1^2+\lambda}, \ldots, \frac{\lambda_K^2}{\lambda_K^2+\lambda}\right)\right\|_2\\ &\leqslant 1, \end{aligned} \tag{119}$$

where (a) is because $\mathbf{V}$ is an unitary matrix and $\lambda_1, \ldots, \lambda_K$ are eigenvalues of $\widetilde{\mathbf{N}}$. Continuing from (117), we obtain:

$$\left\|\Phi^{1/2}\widetilde{\boldsymbol{\xi}}\right\|_2 = \left\|\left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}} + \lambda\mathbf{I}\right)^{-1}\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}}\right\|_2 \leqslant \|\mathbf{z}\|_2. \tag{120}$$

Let us define $\mathbf{x}_0 := \left(\mathbf{N}^T\mathbf{N}\right)^{-1}\mathbf{N}^T\mathbf{x}$. Thus, we have:

$$\begin{aligned}\mathbf{z} &= \left(\widetilde{\mathbf{N}}^T\widetilde{\mathbf{N}}\right)^{-1}\widetilde{\mathbf{N}}^T\mathbf{x}\\ &= \left(\Phi^{-1/2}\mathbf{N}^T\mathbf{N}\Phi^{-1/2}\right)^{-1}\Phi^{-1/2}\mathbf{N}^T\mathbf{x}\\ &= \Phi^{1/2}\left(\mathbf{N}^T\mathbf{N}\right)^{-1}\mathbf{N}^T\mathbf{x}\\ &= \Phi^{1/2}\mathbf{x}_0. \end{aligned} \tag{121}$$

Therefore, we can bound the $\|\boldsymbol{\xi}\|_\Phi := \sqrt{\boldsymbol{\xi}^T\Phi\boldsymbol{\xi}}$:

$$\|\boldsymbol{\xi}\|_\Phi = \left\|\Phi^{1/2}\boldsymbol{\xi}\right\|_2 \leqslant \|\mathbf{z}\|_2 = \|\mathbf{x}_0\|_\Phi. \tag{122}$$

Since Φ is symmetric, by Rayleigh-Ritz theorem we know that:

$$0 \overset{(a)}{<} \lambda_{\min}^{\Phi} \leqslant \frac{\boldsymbol{\xi}^T \Phi \boldsymbol{\xi}}{\boldsymbol{\xi}^T \boldsymbol{\xi}} = \frac{\|\boldsymbol{\xi}\|_{\Phi}^2}{\|\boldsymbol{\xi}\|_2^2} \leqslant \lambda_{\max}^{\Phi}, \tag{123}$$

where $\lambda_{\min}^{\Phi}, \lambda_{\max}^{\Phi}$ are minimum and maximum eigenvalues of Φ, and (a) is due to the fact that since Φ is kernel matrix of RKHS space and $\{b_k(\cdot)\}_{k=1}^K$ are basis functions, it is positive definite. Thus, we have:

$$\|\boldsymbol{\xi}\|_2^2 \leqslant \frac{1}{\lambda_{\min}^{\Phi}} \|\boldsymbol{\xi}\|_{\Phi}^2 \leqslant \frac{1}{\lambda_{\min}^{\Phi}} \cdot \|\mathbf{x}_0\|_{\Phi}^2 \leqslant \frac{\lambda_{\max}^{\Phi}}{\lambda_{\min}^{\Phi}} \|\mathbf{x}_0\|_2^2 = \kappa(\Phi) \|\mathbf{x}_0\|_2^2 . \tag{124}$$

Applying the bound for condition number introduce in [66], we can complete the proof:

$$\kappa(\Phi) < \frac{2}{|\det \Phi|} \left(\frac{\|\Phi\|_F^2}{K+4} \right)^{\frac{K+4}{2}}, \tag{125}$$

where $\|\cdot\|_F$ is the Frobenius norm. □

Using Lemma 6, (115), and (116) we have:

$$\begin{aligned}
\|u^*\|_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})} &\leqslant \|\boldsymbol{\xi}\|_2^2 \cdot \left(\sum_{k=1}^{K+4} \|b_k\|_{L^\infty(\Omega;\mathbb{R})}^2 + \sum_{k=1}^{K+4} \|b_k'\|_{L^\infty(\Omega;\mathbb{R})}^2 \right) + \int_\Omega |\widetilde{u_{\text{enc}}}''(t)|^2 \, dt \\
&\overset{(a)}{\leqslant} \frac{2}{|\det \Phi|} \left(\frac{\|\Phi\|_F^2}{K+4} \right)^{\frac{K+4}{2}} \left\| \widetilde{\boldsymbol{\xi}} \right\|_2^2 \left(\sum_{k=1}^{K+4} \|b_k\|_{L^\infty(\Omega;\mathbb{R})}^2 + \sum_{k=1}^{K+4} \|b_k'\|_{L^\infty(\Omega;\mathbb{R})}^2 \right) \\
&\quad + \int_\Omega |\widetilde{u_{\text{enc}}}''(t)|^2,
\end{aligned} \tag{126}$$

where (a) follows by applying Lemma 6. Setting M equal to the right-hand side of Equation (126) completes the proof.

B.5 Proof of Theorem 4

Consider a natural spline $\widetilde{u_{\text{enc}}}(t)$ as the encoder function fitted on the data points $\{(\alpha_k, x_k)\}_{k=1}^K$. Let $u_{\text{enc}}^*(t)$ denote the optimal encoder minimizing the upper bound in (10). Then, we have:

$$\begin{aligned}
\mathcal{R}(\hat{f}) &\leqslant \frac{2q^2}{K} \sum_{k=1}^{K} (u_{\text{enc}}^*(\alpha_k) - x_k)^2 + \lambda_{\text{e}} \cdot g\big(\|u_{\text{enc}}^*\|_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}^2 \big) \\
&\overset{(a)}{\leqslant} \frac{2q^2}{K} \sum_{k=1}^{K} (\widetilde{u_{\text{enc}}}(\alpha_k) - x_k)^2 + \lambda_{\text{e}} \cdot g\big(\|\widetilde{u_{\text{enc}}}\|_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}^2 \big) \\
&\overset{(b)}{=} \lambda_{\text{e}} \cdot g\big(\|\widetilde{u_{\text{enc}}}\|_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}^2 \big),
\end{aligned} \tag{127}$$

where (a) follows from the optimality of $u_{\text{enc}}^*(t)$, and (b) is due to the fact that $\widetilde{u_{\text{enc}}}(\alpha_k) = x_k$, since $\widetilde{u_{\text{enc}}}(\cdot)$ is a natural spline. Note that $g\big(\|\widetilde{u_{\text{enc}}}\|_{\widetilde{\mathcal{H}}^2(\Omega;\mathbb{R})}^2 \big)$ is independent of N and S, and depends only on α_k and x_k for $k \in [K]$. Additionally, based on the Theorem 3 and (106), $\lambda_{\text{e}} = \mathcal{O}(S^3 N^{-3})$ and $\lambda_{\text{e}} = \mathcal{O}(S^{\frac{8}{5}} N^{-\frac{3}{5}})$ for the noiseless and noisy cases, respectively. Thus, the upper bound provided in (10) converges at most at the rate of $\mathcal{O}(S^3 N^{-3})$ for the noiseless case and $\mathcal{O}(S^{\frac{8}{5}} N^{-\frac{3}{5}})$ for the noisy case.

C Comparison with Berrut Coded Computing

C.1 Convergence rate

The upper bound of the infinity norm for the estimation provided in [29] for the coded computing scheme with N workers and maximum of S stragglers is as follows:

$$\left\| \hat{f}_{\text{BACC}}(t) - f \circ u_{\text{enc}}(t) \right\|_{L^\infty(\Omega;\mathbb{R})} \leqslant 2(1+R) \sin\left(\frac{(S+1)\pi}{2N} \right) \|f \circ u_{\text{enc}}''(t)\|_{L^\infty(\Omega;\mathbb{R})}, \tag{128}$$

https://doi.org/10.52202/079017-3554

if $N-s$ is odd, and

$$\left\|\hat{f}_{\texttt{BACC}}(t)-f\circ u_{\text{enc}}(t)\right\|_{L^\infty(\Omega;\mathbb{R})}\leqslant 2(1+R)\sin\left(\frac{(S+1)\pi}{2N}\right)\Big(\|f\circ u''_{\text{enc}}(t)\|_{L^\infty(\Omega;\mathbb{R})} + \|f\circ u'_{\text{enc}}(t)\|_{L^\infty(\Omega;\mathbb{R})}\Big), \quad (129)$$

if $N-s$ is even, where $R=\frac{(s+1)(s+3)\pi^2}{4}$. Since $\|\cdot\|_{L^2(\Omega;\mathbb{R})}$ is upper bounded by $\|\cdot\|_{L^\infty(\Omega;\mathbb{R})}$, we can directly derive a convergence rate for the squared $L^2(\Omega;\mathbb{R})$-norm of the error as N increases:

$$\left\|\hat{f}_{\texttt{BACC}}(t)-f\circ u_{\text{enc}}(t)\right\|^2_{L^2(\Omega;\mathbb{R})}\leqslant\mathcal{L}(\Omega)\cdot\left\|\hat{f}_{\texttt{BACC}}(t)-f\circ u_{\text{enc}}(t)\right\|^2_{L^\infty(\Omega;\mathbb{R})}\leqslant\mathcal{O}(S^4N^{-2}). \quad (130)$$

Compared to our results, the upper bound for LeTCC provided in Theorem 1, $\mathcal{O}(S^3N^{-3})$, is less sensitive to the number of stragglers and converges faster with increasing N. Note that, since the $\|\cdot\|_{L^2(\Omega;\mathbb{R})}$ is upper bounded by $\|\cdot\|_{L^\infty(\Omega;\mathbb{R})}$, the statement above does not guarantee faster convergence of the proposed scheme compared to Berrut approach.

It should be noted that [29] does not analyze the noisy setting.

C.2 Computational complexity

From the experimental view, we compare the whole encoding and decoding time (on a single CPU machine) for LeTCC and BACC frameworks, as shown in the following table:

Table 2: Average and std of end-to-end processing time of LeTCC and BACC for different architectures

	BACC	LeTCC
LeNet5, $(N,K)=(100,20)$	$0.013s\pm0.002$	$0.007s\pm0.001$
RepVGG, $(N,K)=(60,20)$	$1.62s\pm0.18$	$1.59s\pm0.14$
ViT, $(N,K)=(20,8)$	$1.60s\pm0.28$	$1.74s\pm0.29$

As shown in Table 2, the end-to-end processing time of the proposed framework is on par with BACC.

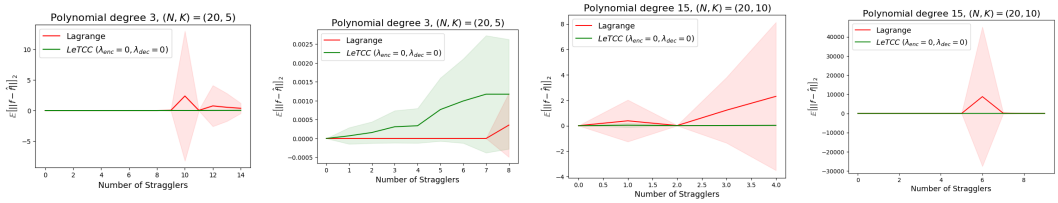
D Comparison with Lagrange Coded Computing

Although the **only** existing coded computing scheme for general functions is Berrut coded computing [29], with which we have compared our proposed scheme, other schemes are designed for specific computations, such as polynomial functions [3] and matrix multiplication [13]. To provide further comparison, we evaluate our proposed scheme against Lagrange coded computing (LCC) [3], which is specifically designed for polynomial computations, as follows:

D.1 Accuracy of function approximation

LCC is applicable only to polynomial computing functions [3]. Additionally, to enable recovery, the number of servers required must be at least $(K-1)\times\deg(f)+S+1$ worker nodes [3, 29]; otherwise, the master node cannot recover any results. Moreover, LCC is designed for computation over finite fields and encounters serious instability when computing over real numbers, particularly when $(K-1)\times\deg(f)$ is around 25 or higher [18, 29].

We compare the proposed framework with LCC in Figure 5. Note that if $N<(K-1)\times\deg(f)+S+1$, LCC cannot operate effectively. To adapt LCC for such cases, we approximate results by fitting a lower-degree polynomial to the available workers' outputs. We run LeTCC and LCC on the same set of input data and a fixed polynomial function for 20 trials, plotting the average performance and corresponding 95% confidence intervals in Figure 5. Figures 5a and 5b illustrate the performances of LCC and LeTCC for a low-degree polynomial and a small number of data points ($\deg(f)=3$ and $K=5$). In contrast, Figures 5c and 5d show performance for a higher-degree polynomial and a larger dataset ($\deg(f)=15$ and $K=10$). As shown in Figures 5a and 5b, LCC achieves exact results for $S\leq 7$. However, at larger values of S, as well as larger polynomial degree (as in Figures 5c and 5d), the proposed approach, without any parameter tuning, outperforms LCC in terms of both computational stability (lower variance) and recovery accuracy.



(a) (b) (c) (d)

Figure 5: Average performance of LeTCC and Lagrange Coded Computing, with a 95% confidence interval. Plots (a) and (d) show the overall performance, while the zoomed-in subplots (b) and (c) highlight the performance for smaller range of stragglers.

D.2 Computational complexity

Encoding and decoding complexities in LCC are $\mathcal{O}(N \cdot \log^2(K) \cdot \log\log(K) \cdot d)$ and $\mathcal{O}((N - S) \cdot \log^2((N - S)) \cdot \log\log((N - S)) \cdot m)$, respectively, where d and m are input and output dimensions of the computing function $f(\cdot)$, respectively [3]. In contrast, as mentioned before, for smoothing splines, the encoding and decoding process, which involves evaluation on new points and calculating the fitted coefficients, have the computational complexity of $\mathcal{O}(K.d)$ and $\mathcal{O}((N-s).m)$. Consequently, the computational complexity of the proposed scheme is less than LCC.

E Sensitivity Analysis

E.1 Sensitivity to number of stragglers

The smoothing parameters for each model show low sensitivity to the number of stragglers (or worker nodes). To find the optimal smoothing parameter, we use cross-validation across different $\frac{S}{N}$ values. The following table presents the optimal smoothing parameters for selected numbers of stragglers for LeNet5 with $(N, K) = (100, 60)$ and RepVGG with $(N, K) = (60, 20)$, respectively. As shown in Table 3, the optimal values of λ_e and λ_d exhibit low sensitivity to the number of stragglers.

Table 3: Optimal smoothing parameters for different number of stragglers for LeNet and RepVGG architectures.

	LeNet5		RepVGG	
(N, K)	$(100, 60)$		$(60, 20)$	
S	λ_e^*	λ_d^*	λ_e^*	λ_d^*
0	10^{-13}	10^{-6}	10^{-6}	10^{-4}
5	10^{-13}	10^{-6}	10^{-6}	10^{-4}
10	10^{-13}	10^{-6}	10^{-5}	10^{-4}
15	10^{-13}	10^{-6}	10^{-5}	10^{-4}
20	10^{-13}	10^{-6}	10^{-5}	10^{-4}
25	10^{-8}	10^{-5}	10^{-5}	10^{-4}
30	10^{-8}	10^{-4}	10^{-5}	10^{-3}
35	10^{-8}	10^{-4}	10^{-5}	10^{-3}

E.2 Sensitivity to smoothing parameters

To assess the performance of the proposed scheme with respect to the smoothing parameters, we vary each parameter individually around its optimal point while holding the other parameter fixed at their optimal value. We then record the average percentage increase in RMSE relative to the RMSE at the

https://doi.org/10.52202/079017-3554

optimal point. Figure 6 presents these results for LeNet with $(N, K, S) = (100, 60, 20)$ (Figures 6a and 6b) and for RepVGG with $(N, K, S) = (60, 20, 35)$ (Figures 6c and 6d).

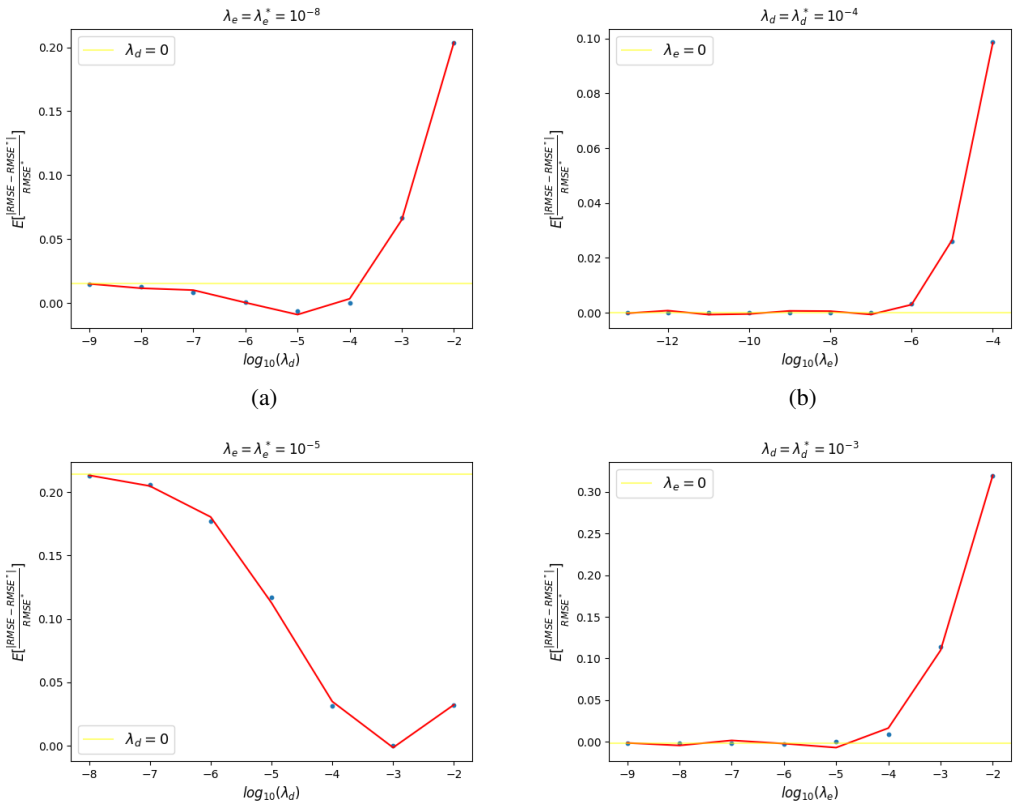


Figure 6: Sensitivity of LeTCC performance with respect to $\log_{10}(\lambda_d)$ and $\log_{10}(\lambda_e)$. The yellow line represents the performance when the variable smoothing parameter is set to zero.

As shown in Figure 6, the presence of more stragglers increases the sensitivity of LeTCC with respect to its smoothing parameter. However, even in a high-straggler regime, the RMSE increases by only around 3% when the smoothing parameter deviates from its optimal value by a scale of 10.

F High-dimensional computing function

Let us consider more general cases where $f = [f_1, \ldots, f_m]$ is a vector-valued function, where each component function $f_j : \mathbb{R} \to \mathbb{R}$ is q_j-Lipschitz continuous. Based on (2), we have:

$$\begin{aligned}
\mathcal{R}(\hat{\mathbf{f}}) &\leqslant \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}, \mathcal{F} \sim F_{S,N}} \left[\frac{2}{K} \sum_{k=1}^{K} \|\mathbf{u}_{\text{dec}}(\alpha_k) - \mathbf{f}(u_{\text{enc}}(\alpha_k))\|_2^2 \right] + \frac{2}{K} \sum_{k=1}^{K} \|\mathbf{f}(u_{\text{enc}}(\alpha_k)) - \mathbf{f}(x_k)\|_2^2 . \\
&\leqslant \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}, \mathcal{F} \sim F_{S,N}} \left[\frac{2}{K} \sum_{k=1}^{K} \sum_{j=1}^{m} \left(u_{\text{dec}_j}(\alpha_k) - f_j(u_{\text{enc}}(\alpha_k)) \right)_2^2 \right] + \frac{2 \sum_{j=1}^{m} q_j^2}{K} \sum_{k=1}^{K} \|u_{\text{enc}}(\alpha_k) - x_k\|_2^2 \\
&= \sum_{j=1}^{m} \mathop{\mathbb{E}}_{\boldsymbol{\epsilon}, \mathcal{F} \sim F_{S,N}} \left[\frac{2}{K} \sum_{k=1}^{K} \left(u_{\text{dec}_j}(\alpha_k) - f_j(u_{\text{enc}}(\alpha_k)) \right)_2^2 \right] + \frac{2 \sum_{j=1}^{m} q_j^2}{K} \sum_{k=1}^{K} \|u_{\text{enc}}(\alpha_k) - x_k\|_2^2
\end{aligned}$$

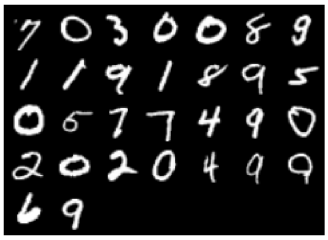
Let us define the following objective for the decoder function:

$$\mathbf{u}_{\text{dec}}^{\star} = \mathop{\text{argmin}}_{\mathbf{u} \in \mathcal{H}^2(\Omega; \mathbb{R}^M)} \frac{1}{|\mathcal{F}|} \sum_{v \in \mathcal{F}} \|\mathbf{u}(\beta_v) - \mathbf{f}(u_{\text{enc}}(\beta_v))\|_2^2 + \sum_{j=1}^{m} \lambda_{\text{d}} \int_{\Omega} \left(u_j''(t) \right)^2 dt. \tag{131}$$

The solution to (131), denoted as $\mathbf{u}^{\star}_{\text{dec}}$, is a vector-valued function, where each component $u_{\text{dec}_j}(\cdot)$ is a smoothing spline function fitted to the data points $\{(\beta_v, f_j(u_{\text{enc}}(\beta_v)))\}_{v\in\mathcal{F}}$. As a result, By defining $q = \sqrt{\sum_{j=1}^{m} q_j^2}$ and scaling up all upper bounds for $\mathcal{L}_{\text{dec}}$ by a factor of m, all previous results and theorems seamlessly extend to high-dimensional computing functions.

G Coded data points

Figures 7b and 7c display coded samples generated by BACC and LeTCC, respectively, derived from the same initial data points depicted in Figure 7a. These samples are presented for the MNIST dataset with parameters $(N, K) = (70, 30)$. From the figures, it is apparent (Specifically in paired ones that are shown with the same color) that while both schemes' coded samples are a weighted combination of multiple initial samples, BACC's coded samples exhibit high-frequency noise. This observation suggests that LeTCC regression functions produce more refined coded samples without any disruptive noise.



(a) Initial inputs

(b) BACC coded samples

(c) LeTCC coded samples

Figure 7: Comparison of coded samples between BACC and LeTCC frameworks. Figure 7a represents the initial data points $\{\mathbf{x}_k\}_{k=1}^{K}$ for $K = 30$. Figures 7b and 7c display $N = 70$ coded samples $\{\tilde{\mathbf{x}}_n\}_{n=1}^{N}$ from BACC and LeTCC, respectively. Samples with clear differences are highlighted with the same color.

https://doi.org/10.52202/079017-3554

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We detailed our contributions clearly in the abstract and the introduction sections of the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:We provided all the details regarding the assumptions, conditions, and limitations in the framework explanations (Section 3), theorems (Section 4), as well as the experiments section (Section 5) in the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our paper includes theoretical results in Section 4. In our theorems, we clearly mentioned all the required assumptions, and a complete (and correct) proof of them is available in appendices (e.g., see Appendix B). Please see Section 3 for a full definition of the problem and introduction to the notations used in the paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We completely explained our proposed framework in Section 3 and we also provided details regarding our empirical evaluations in Section 3 in the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

 https://doi.org/10.52202/079017-3554

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We provided references to all the open datasets that we used in the paper. Regarding the code, we are happy to share it later if required.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so No is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provided full experimental details in the paper (see Section 5).

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The details are provided in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided all the details regarding our experiments in Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We followed the NeurIPS code of ethics in our paper.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our paper is focused on developing a new framework for coded distributed computing and should be categorized as foundational research. We believe this work has no direct societal impact that should be explained in the paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

https://doi.org/10.52202/079017-3554

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This is not applicable to our work and this paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: This paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our paper does not involve these.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our paper does not involve these.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

https://doi.org/10.52202/079017-3554

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.