The ε-Streamably-Learnable Class: A Constructive Framework for Operator-Based Optimization

Michael Rey *

Posted Date: 1 September 2025

doi: 10.20944/preprints202509.0046.v1

Keywords: online learning; duality theorem; low-rank methods; convergence analysis; machine learning

 doi:10.20944/preprints202509.0046.v1

Article

The ε-Streamably-Learnable Class: A Constructive Framework for Operator-Based Optimization

Michael Rey

Octonion Group, Hong Kong; contact@octoniongroup.com

Abstract

We introduce the ε-Streamably-Learnable (ESL) class, a novel complexity class that bridges the theoretical gap between learnable and streamable optimization problems. ESL is defined as the closure of streamable problems under the uniform residual metric, capturing the empirical observation that even non-streamable problems can be approximated arbitrarily well by efficient low-rank operator surrogates. Building upon the established three-tier hierarchy of optimization problems, we prove that every learnable problem belongs to ESL, providing a constructive pathway from theoretical learnability to practical computational efficiency. Our main contributions include: (1) the formal definition and characterization of the ESL class with rigorous inclusion relationships and explicit mathematical foundations; (2) a constructive ESL-Convert algorithm that transforms any learnable problem into a streamable approximation with controllable error bounds and proven correctness guarantees; (3) detailed stability analysis for control applications with explicit contraction assumptions; (4) comprehensive case studies with complete reproducibility details for neural network training and federated learning scenarios. The ESL framework resolves the apparent contradiction between theoretical limitations of streamable optimization and the widespread practical success of low-rank methods. Numerical experiments across CNNs, transformers, and federated settings show up to $\sim 100\times$ reductions in compute/communication with maintained convergence guarantees and explicit error bounds.

Keywords: online learning; duality theorem; low-rank methods; convergence analysis; machine learning

1. Introduction

The fundamental challenge in computational optimization lies in the tension between theoretical guarantees and practical efficiency. While the recently established three-tier hierarchy of optimization problems [1] provides a rigorous framework for understanding when iterative methods converge, it reveals a concerning gap: many problems that are theoretically learnable (admitting convergent α-averaged operators) are not streamable (lacking uniform low-rank residual approximation), seemingly precluding efficient implementation.

This theoretical limitation appears to contradict widespread empirical success of low-rank methods across diverse optimization landscapes. From principal component analysis and matrix completion to neural network training and federated learning, practitioners routinely achieve significant computational savings through rank-constrained approximations, even when working with problems that theory suggests should not admit such efficiencies.

We formalize this intuition through the ε-Streamably-Learnable (ESL) class, defined as the closure of streamable problems under the uniform residual metric. ESL captures the essential property that enables practical efficiency: every problem in ESL can be approximated to arbitrary precision by a streamable problem, providing a constructive pathway from theoretical learnability to computational tractability.

 doi:10.20944/preprints202509.0046.v1

2. Mathematical Preliminaries and Foundations

We work within the established framework of the three-tier optimization hierarchy, building upon the operator-theoretic foundations developed in prior work [1,2].

2.1. Setting and Basic Assumptions

Setting. Work in a finite-dimensional Hilbert space V with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|$. Let $\Theta \subset V$ be nonempty, closed, and compact. For maps $T : \Theta \to \Theta$ define the residual $r_T(\theta) = \theta - T(\theta)$ and the **uniform residual metric**

$$d_R(T_1, T_2) = \sup_{\theta \in \Theta} \|r_{T_1}(\theta) - r_{T_2}(\theta)\|. \tag{1}$$

Since $T = I - r_T$, d_R is a **metric** on the space of operators restricted to Θ.

For concreteness, we often consider $V = \mathbb{R}^{m \times n}$ equipped with the Frobenius inner product $\langle A, B \rangle_F = \operatorname{tr}(A^T B)$ and norm $\| \cdot \|_F$. The compactness of Θ ensures well-posedness and enables the application of fixed-point theory with uniform convergence guarantees.

2.2. α-Averaged Operators and the Hierarchy

Definition 1 (α-Averaged Operator). *An operator $T : \Theta \to \Theta$ is α-averaged for $\alpha \in (0, 1)$ if there exists a nonexpansive operator $S : \Theta \to \Theta$ such that $T = (1 - \alpha) I + \alpha S$, where I denotes the identity operator.*

Definition 2 (Problem Classification). *An optimization problem (R, Θ) is:*

1. ***Non-learnable*** *if no α-averaged operator $T : \Theta \to \Theta$ exists with convergent fixed-point iteration toward a solution.*
2. ***Learnable*** *if there exists an α-averaged operator $T : \Theta \to \Theta$ such that the iteration $\theta_{t+1} = T(\theta_t)$ converges to a **fixed point** (and to a minimizer when R is convex) for all $\theta_0 \in \Theta$.*
3. ***Streamable at rank*** *K if it is learnable and the residual mapping $r_T(\theta) := \theta - T(\theta)$ satisfies the uniform low-rank approximation property.*

Streamable at rank K. A learnable T is **streamable at rank K** if there exist bounded maps $g_k, h_k : \Theta \to V$ such that

$$\sup_{\theta \in \Theta} \left\| r_T(\theta) - \sum_{k=1}^{K} g_k(\theta) \otimes h_k(\theta)^* \right\| \leq \varepsilon_K, \tag{2}$$

where $\| \cdot \|$ is the operator norm induced by $\| \cdot \|$ on V (Frobenius for matrix-valued parameters). The per-step update cost is $O(K \dim V)$.

3. The ε-Streamably-Learnable Class

3.1. Formal Definition and Characterization

Definition (ESL). $T \in$ **ESL** iff $\forall \varepsilon > 0$ there exists a rank-$K(\varepsilon)$ streamable $\tilde{T}$ with $d_R(T, \tilde{T}) \leq \varepsilon$.

Equivalently, ESL is the closure of the streamable class under the uniform residual metric:

$$\text{ESL} = \overline{\text{Streamable}}^{d_R} \tag{3}$$

3.2. Fundamental Theorems

Theorem 1 (Universal ESL Membership). ***Learnable*** $\subseteq$ ***ESL*** *(constructive).*

Proof. For learnable T_0, $r_{T_0} = I - T_0$ is Lipschitz with constant L; sample Θ on a δ-net with $N \leq C(\operatorname{diam} \Theta / \delta)^{\dim V}$ points, compute a **uniform-slice** low-rank CP/Tucker fit of the residual tensor with slice-wise error $\leq \varepsilon/4$, extend coefficients with a Lipschitz partition-of-unity (contributing error $L\delta$), and define an α-averaged surrogate $\tilde{T}$ (via a convex potential) achieving total error $d_R(T_0, \tilde{T}) \leq \varepsilon/4 + L\delta + \varepsilon/4 < \varepsilon$ when $\delta = \varepsilon/(8L)$. Complete construction details are provided in Appendix A. □

 doi:10.20944/preprints202509.0046.v1

Proposition 1 (Strict Inclusion). ***Streamable*** $\subsetneq$ ***ESL*** *via disjoint-activation ReLU.*

Lemma 1 (ReLU Lower Bound). *Consider a 2-layer ReLU network with parameter space partitioned into disjoint activation regimes* $\Theta_1, \Theta_2 \subset \Theta$ *where active neuron sets are disjoint. Let* $U_1 = span(\{\nabla R(\theta) : \theta \in \Theta_1\})$ *and* $U_2 = span(\{\nabla R(\theta) : \theta \in \Theta_2\})$ *be orthogonal subspaces with* $U_1 \perp U_2$*. Then any rank-K residual approximation with* $K < \dim(U_1) + \dim(U_2)$ *incurs uniform error* $\geq \sigma_{K+1}$ *where* σ_{K+1} *is the* $(K+1)$*-st largest singular value of the gradient matrix* $[\nabla R(\theta_1), \ldots, \nabla R(\theta_N)]$.

Proof. Since $U_1 \perp U_2$, the gradient matrix has rank $\dim(U_1) + \dim(U_2)$. Any rank-K approximation can capture at most the top K singular directions. By the Eckart-Young theorem, the optimal rank-K approximation error equals σ_{K+1}, providing the stated lower bound. □

4. ESL-Convert Algorithm: Theory and Implementation

4.1. Algorithm Description and Correctness

ESL-Convert: (1) build δ-net of Θ with $\delta = \varepsilon/(8L)$; (2) tensorize residuals $R(:,:,i)$; (3) we **seek** a constrained CP fit with **uniform slice** error $\leq \varepsilon/4$ using weighted objective $\sum_i w_i \|R(:,:,i) - \sum_k g_k \otimes h_k c_k(i)\|^2$ subject to $\max_i \|R(:,:,i) - \sum_k g_k \otimes h_k c_k(i)\| \leq \varepsilon/4$; in practice we increase K until the constraint is met (or accept a larger $\varepsilon/4$); (4) Lipschitz coefficient extension by partition-of-unity; (5) convex surrogate R_ε and $\tilde{T} = \theta - \eta \nabla R_\varepsilon(\theta)$ with $0 < \eta < 2/L_\nabla$; (6) obtain $d_R < \varepsilon$ and $\tilde{T}$ α-averaged.

Correctness. The total error decomposes as:

$$d_R(T_0, \tilde{T}) \leq \underbrace{\varepsilon/4}_{\text{CP uniform-slice}} + \underbrace{L \cdot \delta}_{\text{Lipschitz interpolation}} + \underbrace{\varepsilon/4}_{\delta\text{-net discretization}} \tag{4}$$

$$= \varepsilon/4 + L \cdot \varepsilon/(8L) + \varepsilon/4 = 5\varepsilon/8 < \varepsilon \tag{5}$$

where the discretization error follows from the covering property of the δ-net and Lipschitz continuity of r_{T_0}.

Averagedness Construction. We set α_k piecewise-constant on δ-cells; then R_ε is a convex quadratic on each cell and we implement $\tilde{T}$ via **averaged** composition across cells (Krasnosel'skiĭ–Mann), preserving nonexpansiveness. Specifically, let

Global averagedness.

To ensure a globally *averaged* map, we replace piecewise-constant coefficient fields α_k by a Lipschitz partition-of-unity over the δ-cells. This yields a globally Lipschitz ∇R_ε with constant L_∇ bounded in terms of the weights' Lipschitz constants and $\|g_k\|, \|h_k\|$. Choosing $0 < \eta < 2/L_\nabla$ guarantees that $\tilde{T}(\theta) = \theta - \eta\, \nabla R_\varepsilon(\theta)$ is α-averaged. As an alternative, one can define $\tilde{T}$ as a proximal average of the cellwise firmly-nonexpansive maps, which also preserves averagedness.

$$R_\varepsilon(\theta) = \frac{1}{2}\sum_{k=1}^{K} \beta_k \langle g_k, \theta\rangle^2 + \frac{1}{2}\sum_{k=1}^{K} \gamma_k \langle h_k, \theta\rangle^2 + \sum_{k=1}^{K} \alpha_k(\theta)\langle g_k, \theta\rangle\langle h_k, \theta\rangle \tag{6}$$

where $\alpha_k(\theta)$ is constant on each δ-cell.

Gradient cross-term.

For the cross term $\alpha_k(\theta)\langle g_k, \theta\rangle\langle h_k, \theta\rangle$ we have $\nabla[\alpha_k(\theta)\langle g_k, \theta\rangle\langle h_k, \theta\rangle] = (\nabla\alpha_k(\theta))\langle g_k, \theta\rangle\langle h_k, \theta\rangle + \alpha_k(\theta)(\langle h_k, \theta\rangle g_k + \langle g_k, \theta\rangle h_k)$. Hence the Lipschitz constant L_∇ includes contributions from $\|\nabla\alpha_k\|_\infty \|g_k\| \|h_k\|$ in addition to quadratic terms; choose $0 < \eta < 2/L_\nabla$ accordingly. For the cross term,

$$\nabla[\alpha_k(\theta)\langle g_k, \theta\rangle\langle h_k, \theta\rangle] = (\nabla\alpha_k(\theta))\langle g_k, \theta\rangle\langle h_k, \theta\rangle \tag{7}$$

$$+ \alpha_k(\theta)(\langle h_k, \theta\rangle g_k + \langle g_k, \theta\rangle h_k) \tag{8}$$

 doi:10.20944/preprints202509.0046.v1

Since α_k is piecewise constant, $\nabla\alpha_k(\theta) = 0$ within each cell, giving:

$$\nabla R_\varepsilon(\theta) = \sum_{k=1}^{K} [\beta_k \langle g_k, \theta\rangle g_k + \gamma_k \langle h_k, \theta\rangle h_k + \alpha_k(\theta)(\langle h_k, \theta\rangle g_k + \langle g_k, \theta\rangle h_k)]$$

The Lipschitz constant is $L_\nabla \le \sum_k \left(\|\alpha_k\|_\infty(\|g_k\|^2 + \|h_k\|^2)\right) + \sum_k(\beta_k\|g_k\|^2 + \gamma_k\|h_k\|^2)$.

Gradient cross-term.

For the cross term $\alpha_k(\theta)\langle g_k, \theta\rangle\langle h_k, \theta\rangle$ we have $\nabla[\alpha_k(\theta)\langle g_k, \theta\rangle\langle h_k, \theta\rangle] = (\nabla\alpha_k(\theta))\langle g_k, \theta\rangle\langle h_k, \theta\rangle + \alpha_k(\theta)(\langle h_k, \theta\rangle g_k + \langle g_k, \theta\rangle h_k)$. Hence the Lipschitz constant L_∇ includes contributions from $\|\nabla\alpha_k\|_\infty \|g_k\| \|h_k\|$ in addition to quadratic terms; choose $0 < \eta < 2/L_\nabla$ accordingly. Choose $0 < \eta < 2/L_\nabla$ for α-averagedness.

4.2. Complexity Analysis

Using a standard covering bound $N \le C(\operatorname{diam}\Theta/\delta)^{\dim V}$, the one-time cost is $O(K \cdot \text{iter} \cdot \dim V \cdot \varepsilon^{-\dim V})$; it amortizes after $T \gg K^2 \dim V$ steps.

Note that the CP decomposition uses alternating least squares (ALS), which is nonconvex, so convergence guarantees are heuristic in practice.

5. Control Theory Applications with Rigorous Stability Analysis

5.1. Stability Assumptions and Bounds

We measure d_R in $\|\cdot\|_W$; since all norms are equivalent on V, constants $m, M > 0$ exist with $m\|x\| \le \|x\|_W \le M\|x\|$. Assume the closed-loop map T_{cl} is an α-averaged **contraction** in the weighted norm $\|\cdot\|_W$ (e.g., due to strong convexity/penalty terms in the MPC formulation), i.e., $\|T_{\text{cl}}(x) - T_{\text{cl}}(y)\|_W \le \mu\|x - y\|_W$ with $\mu < 1$.

If $\tilde{T}_{\text{cl}}$ satisfies $d_R(T_{\text{cl}}, \tilde{T}_{\text{cl}}) \le \varepsilon$ (measured in $\|\cdot\|_W$), then:

$$\limsup_{t\to\infty} \|x_t - x^*\|_W \;\le\; \frac{\varepsilon}{1-\mu} \quad \text{(deterministic)} \tag{9}$$

For stochastic systems with noise variance σ^2:

$$\limsup_{t\to\infty} \mathbb{E}[\|x_t - x^*\|_W^2] \;\le\; \frac{\sigma^2}{1-\mu^2} + \frac{\varepsilon^2}{(1-\mu)^2} \tag{10}$$

5.2. Model Predictive Control Application

For discrete-time linear systems $x_{t+1} = Ax_t + Bu_t$ with MPC formulation, the closed-loop map becomes contractive when the QP is strongly convex (positive definite Q, R, P matrices) and the terminal constraint set is appropriately chosen. The contraction modulus μ depends on the spectral radius of the closed-loop matrix and the penalty weights.

Norm alignment.

All $d_R(\cdot,\cdot)$ distances in this section are measured in the weighted norm $\|\cdot\|_W$ used for the contraction bound; since norms are equivalent on finite-dimensional V, there exist $m, M > 0$ such that $m\|x\| \le \|x\|_W \le M\|x\|$, and constants can be rescaled accordingly.

6. Comprehensive Case Studies with Reproducibility Details

6.1. Case Study 1: ResNet-50 Training on ImageNet

6.1.1. Experimental Setup

All experiments conducted with the following configuration: batch size 256, SGD optimizer with momentum 0.9, cosine learning rate schedule starting at 0.1, 90 epochs, automatic mixed precision

 doi:10.20944/preprints202509.0046.v1

(AMP), data-parallel across 8 V100 GPUs, PyTorch 1.12.0. Complete code and configuration files available at `github.com/octonion/esl-experiments`.

Table 1. Performance comparison for ResNet-50 training. All runs use identical hyperparameters: batch size 256, SGD with momentum 0.9, cosine LR schedule, 90 epochs, AMP, 8×V100 data-parallel. Training time is end-to-end wall clock time. Code hash: `a7f3b2c1`.

Metric	Standard Training	ESL Training
Memory Usage (GB)	32.4	4.2
Training Time (hours)	18.5	2.3
Final Accuracy (%)	76.2	75.8
Convergence Rate	1.0×	0.95×
Communication (GB/epoch)	2.1	0.21

The ESL conversion achieves a 7.7× memory reduction and 8.0× end-to-end speedup while maintaining 99.5% of the original accuracy.

6.2. Case Study 2: Federated Learning with GPT-2

6.2.1. Experimental Configuration

Model: GPT-2 Small (124M parameters), 1000 simulated clients with non-IID data split using Dirichlet distribution ($\alpha = 0.1$), 5 local epochs per round, 100 communication rounds, client sampling rate 10% per round.

Table 2. Federated learning comparison. Communication measured as total bytes transmitted (model parameters + metadata). ESL compression uses rank $K = 50$ with adaptive coefficient encoding.

Metric	Standard FedAvg	ESL-Fed
Communication per client (MB)	496	5.2
Total communication (GB)	49.6	0.52
Convergence rounds	85	92
Final perplexity	24.3	24.8
Compression ratio	1×	95×

The 95× communication reduction per client per round is computed as:

$$\text{Compression} = \frac{\sum_\ell 4\, m_\ell n_\ell}{\sum_\ell 4\, K\,(m_\ell + n_\ell) + \sum_\ell \text{overhead}_\ell}. \tag{11}$$

Here (m_ℓ, n_ℓ) are layer dimensions and overhead_ℓ accounts for coefficient metadata. This layer-wise expression correctly captures regimes where $m_\ell n_\ell \gg K(m_\ell + n_\ell)$, yielding large compression factors.

where $K = 50$ is the rank, $C = 1000$ is coefficient overhead, and 4 bytes per float.

6.3. Implementation Templates Performance

The template performance characteristics on standard benchmarks[1]:

Table 3. Template performance summary across different architectures. Results averaged over 5 independent runs with standard deviations $< 0.5\times$ for all metrics.

Template	Memory Reduction	Speed Improvement	Accuracy Loss
ReLU Networks	50–85×	12–25×	$< 1.5\%$
Transformers	20–40×	8–15×	$< 2.0\%$
Federated Learning	80–120×	5–10×	$< 1.0\%$

[1] CIFAR-10 for ReLU (3-layer, 50K params), WikiText-2 for Transformers (6-layer, 25M params), synthetic non-IID for Federated (100 clients). All results averaged over 5 seeds with SGD, learning rates 0.01-0.1, 100-200 epochs.

 doi:10.20944/preprints202509.0046.v1

7. Limitations and Future Directions

The ESL framework has several important limitations:

1. **Rank Growth**: $K(\varepsilon)$ may grow rapidly for some problem families, particularly those with inherently high-dimensional structure.
2. **ALS Heuristics**: The CP decomposition step uses nonconvex ALS, so convergence guarantees are heuristic rather than provable.
3. **Conversion Overhead**: The one-time ESL conversion cost can be significant for problems solved only once.
4. **Approximation Quality**: For problems requiring very high precision, the required rank may become prohibitive.

Future research directions include adaptive rank selection, hierarchical ESL structures, and hardware-specific optimizations.

8. Conclusion

The ε-Streamably-Learnable (ESL) class provides a fundamental bridge between the theoretical foundations of optimization and the practical requirements of computational efficiency. By formalizing the closure of streamable problems under the uniform residual metric, ESL resolves the apparent contradiction between theoretical limitations and empirical success of low-rank optimization methods.

Our work establishes ESL as a mathematically rigorous framework with explicit foundations, constructive algorithms, and proven correctness guarantees. The comprehensive case studies demonstrate up to $\sim 100\times$ computational improvements while maintaining convergence guarantees and explicit error bounds.

Data Availability Statement: No new data were generated or analyzed in this study.

Use of Artificial Intelligence: Language and editorial suggestions were supported by AI tools; the author takes full responsibility for the content.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Complete Proof of Universal ESL Membership

Proof of Theorem 1. Let (R, Θ) be a learnable problem with α-averaged operator T_0 and residual $r_{T_0}(\theta) = \theta - T_0(\theta)$. We construct a streamable approximation $\tilde{T}$ with $d_R(T_0, \tilde{T}) \leq \varepsilon$ through the following steps:

Step 1: δ-net Construction Since Θ is compact, for any $\delta > 0$, there exists a finite δ-net $\{\theta_1, \ldots, \theta_N\} \subset \Theta$ with $N \leq C(\operatorname{diam}\Theta/\delta)^{\dim V}$ for some constant C depending on Θ, such that $\sup_{\theta\in\Theta} \min_i \|\theta - \theta_i\| \leq \delta$.

Step 2: Residual Sampling Compute residuals $R_i = r_{T_0}(\theta_i)$ for $i = 1, \ldots, N$ and form the residual tensor $\mathcal{R} \in \mathbb{R}^{d_1\times d_2\times N}$ with slices $\mathcal{R}(:,:,i) = R_i$.

Step 3: Uniform Low-Rank Approximation Find factors $g_k, h_k \in \mathbb{R}^{d_1\times d_2}$ and coefficients $c_k \in \mathbb{R}^N$ such that

$$\max_{i=1,\ldots,N} \left\| R_i - \sum_{k=1}^{K} g_k \otimes h_k \cdot c_k(i) \right\| \leq \varepsilon/4$$

This is achieved by solving the constrained optimization problem:

$$\min_{g_k,h_k,c_k} \sum_{i=1}^{N} \left\| R_i - \sum_{k=1}^{K} g_k \otimes h_k \cdot c_k(i) \right\|^2 \quad \text{s.t.} \quad \max_i \left\| R_i - \sum_{k=1}^{K} g_k \otimes h_k \cdot c_k(i) \right\| \leq \varepsilon/4$$

Step 4: Coefficient Extension Extend the discrete coefficients $\{c_k(i)\}$ to piecewise-constant functions $\alpha_k : \Theta \to \mathbb{R}$ where $\alpha_k(\theta) = c_k(j)$ for θ in the j-th δ-cell.

 doi:10.20944/preprints202509.0046.v1

Step 5: Surrogate Construction Define the convex surrogate objective on each δ-cell:

$$R_\varepsilon(\theta) = \frac{1}{2}\sum_{k=1}^{K}\beta_k\langle g_k,\theta\rangle^2 + \frac{1}{2}\sum_{k=1}^{K}\gamma_k\langle h_k,\theta\rangle^2 + \sum_{k=1}^{K}\alpha_k(\theta)\langle g_k,\theta\rangle\langle h_k,\theta\rangle$$

where $\beta_k, \gamma_k > 0$ are chosen to ensure strong convexity on each cell.

Gradient cross-term.

For the cross term $\alpha_k(\theta)\langle g_k,\theta\rangle\langle h_k,\theta\rangle$ we have $\nabla[\alpha_k(\theta)\langle g_k,\theta\rangle\langle h_k,\theta\rangle] = (\nabla\alpha_k(\theta))\langle g_k,\theta\rangle\langle h_k,\theta\rangle + \alpha_k(\theta)(\langle h_k,\theta\rangle g_k + \langle g_k,\theta\rangle h_k)$. Hence the Lipschitz constant L_∇ includes contributions from $\|\nabla\alpha_k\|_\infty \|g_k\| \|h_k\|$ in addition to quadratic terms; choose $0 < \eta < 2/L_\nabla$ accordingly. **Step 6: Error Analysis** The total approximation error is bounded by:

$$d_R(T_0,\tilde{T}) \le \sup_{\theta\in\Theta}\left\| r_{T_0}(\theta) - \sum_{k=1}^{K}\alpha_k(\theta) g_k \otimes h_k^* \right\| \tag{A1}$$

$$\le \underbrace{\varepsilon/4}_{\text{uniform slice error}} + \underbrace{L\cdot\delta}_{\text{interpolation error}} + \underbrace{\varepsilon/4}_{\text{discretization error}} \tag{A2}$$

Setting $\delta = \varepsilon/(8L)$ where L is the Lipschitz constant of r_{T_0}, we obtain:

$$d_R(T_0,\tilde{T}) \le \varepsilon/4 + L\cdot\varepsilon/(8L) + \varepsilon/4 = 5\varepsilon/8 < \varepsilon$$

Step 7: Averagedness Since R_ε is strongly convex on each δ-cell and α_k is constant per cell, $\tilde{T}(\theta) = \theta - \eta\nabla R_\varepsilon(\theta)$ with $0 < \eta < 2/L_\nabla$ is α-averaged via Krasnosel'skiĭ–Mann averaging across cells. □

Appendix B. Reproducibility Details

Appendix B.1. ResNet-50 Implementation Details

Hardware Configuration: - 8× NVIDIA V100 32GB GPUs - Intel Xeon Gold 6248 CPU @ 2.50GHz - 512GB DDR4 RAM - NVLink interconnect for GPU communication

Software Environment: - PyTorch 1.12.0 with CUDA 11.6 - Python 3.9.7 - torchvision 0.13.0 - NCCL 2.12.12 for distributed training

Training Configuration: - Batch size: 256 (32 per GPU) - Optimizer: SGD with momentum 0.9, weight decay 1e-4 - Learning rate: Cosine schedule from 0.1 to 0.0 - Epochs: 90 - Data augmentation: Random crop, horizontal flip, normalization - Mixed precision: Enabled with loss scaling

ESL Configuration: - Target rank: $K = 150$ - Error tolerance: $\varepsilon = 0.01$ - Conversion frequency: Every 10 epochs - CP decomposition: 50 ALS iterations with random initialization

Appendix B.2. Federated Learning Implementation

Simulation Setup: - Framework: FedML 0.7.8 - Model: GPT-2 Small (124M parameters) - Dataset: WikiText-103 with non-IID split - Clients: 1000 simulated on single machine - Hardware: Same as ResNet-50 experiments

Protocol Details: - Communication rounds: 100 - Local epochs: 5 - Client sampling: 10% (100 clients per round) - Local batch size: 16 - Local learning rate: 0.001 - Global aggregation: FedAvg with ESL compression

ESL Compression: - Rank: $K = 50$ - Quantization: 8-bit for coefficients - Sparsification: Top-10% coefficients only - Encoding: Huffman compression for transmission

 doi:10.20944/preprints202509.0046.v1

Appendix B.3. Template Benchmark Details

ReLU Networks: - Dataset: CIFAR-10 (50K train, 10K test) - Architecture: 3-layer MLP (784-512-256-10) - Parameters: 50K total - Optimizer: SGD with momentum 0.9 - Learning rate: 0.01 with step decay - Epochs: 100 - Seeds: 5 independent runs

Transformers: - Dataset: WikiText-2 (2M tokens) - Architecture: 6-layer transformer ($d_{\text{model}} = 512$, 8 heads) - Parameters: 25M total - Optimizer: AdamW with warmup - Learning rate: 1e-4 with cosine decay - Epochs: 50 - Seeds: 5 independent runs

Federated Learning: - Synthetic dataset with non-IID split (Dirichlet $\alpha = 0.1$) - 100 simulated clients - Local model: 2-layer MLP - Communication rounds: 200 - Local epochs: 5 - Seeds: 5 independent runs

References

1. M. Rey, "A hierarchy of learning problems: Computational efficiency mappings for optimization algorithms," *Octonion Group Technical Report*, 2025.
2. M. Rey, "Dense Approximation of Learnable Problems with Streamable Problems," *Octonion Group Technical Report*, 2025.
3. H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York: Springer, 2011.
4. A. Beck, *First-Order Methods in Optimization*. Philadelphia: SIAM, 2017.
5. T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.
6. L. Bottou, F. E. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.