In-memory Training on Analog Devices with Limited Conductance States via Multi-tile Residual Learning

Jindan Li[1], **Zhaoxian Wu**[1], **Gaowen Liu**[2], **Tayfun Gokmen**[3], **Tianyi Chen**[1,4]
[1] Cornell University, [2] Cisco Research, [3] IBM Research, [4] RPI

Abstract

Analog in-memory computing (AIMC) accelerators enable efficient deep learning directly within memory using resistive crossbar arrays, where model parameters are represented by the conductance states of memristive devices. However, effective AIMC-based training typically requires at least 8-bit conductance states to match digital baselines. Realizing such fine-grained states is costly and often requires complex noise mitigation techniques that increase circuit complexity and energy consumption. In practice, many promising memristive devices, such as ReRAM, offer only about 4-bit resolution due to fabrication constraints, leading to substantially degraded training accuracy. To overcome the fabrication constraints, this paper proposes a purely algorithmic framework - *multi-tile residual learning* that sequentially learns on multiple crossbar tiles to compensate for the errors from low-precision weight updates. Our theory shows that the optimality gap shrinks with the number of tiles and achieves a linear convergence rate. Experiments on standard image classification benchmarks demonstrate that our method consistently outperforms state-of-the-art in-memory analog training strategies under limited-state settings, while incurring only moderate hardware overhead, as confirmed by our cost analysis.

1 INTRODUCTION

With the growing adoption of AI across various fields, the demand for *accurate and energy-efficient* training hardware is increasing. In this context, *analog in-memory computing* (AIMC) is an emerging solution that performs matrix vector multiplication (MVM) operations directly on weights stored in memory, offering significant efficiency improvements over conventional von Neumann systems. In AIMC hardware, the parameters (matrices) of deep neural networks (DNN) are represented by the conductance states of *memristive devices* in analog crossbar arrays, while the inputs (vectors) are programmed as voltage signals. Using Kirchhoff's and Ohm's laws, MVM operations between a $D \times D$ matrix and a vector can be completed in $\mathcal{O}(1)$ time on AIMC hardware (Hu et al., 2016), while a dense digital MVM requires $\mathcal{O}(D^2)$ operations. A full MVM on analog hardware can be executed with energy in the range of tens of femtojoules (10^{-15} joules), whereas accessing a 1 kB SRAM block in digital systems typically costs 1 picojoule (10^{-12} joules) per byte (Murmann, 2021). This advantage translates into higher energy efficiency. A typical commercial digital accelerator has plateaued around 10 tera-operations per second per watt (TOPS/W) (Reuther et al., 2022), which can be significantly surpassed by AIMC accelerators. For example, a monolithic 3D AIMC chip achieves more than 210 TOPS/W (Chen et al., 2022), and a 4×4 core array reaches 30 TOPS/W (Jia et al., 2022). However, due to the inherent difficulty in precisely and reliably changing the conductance of the memory elements, in-memory analog training presents significant challenges.

This paper focuses on gradient-based in-memory training on AIMC hardware. The objective of training is to solve the optimization problem, formally defined as:

$$W^* := \arg \min_{W \in \mathbb{R}^{D \times D}} f(W) \tag{1}$$

where $f(\cdot) : \mathbb{R}^{D \times D} \to \mathbb{R}$ is the objective and W is a trainable matrix stored in analog crossbar arrays. In digital accelerators, (1) can be solved by stochastic gradient descent (SGD), whose recursion is given by $W_{t+1} = W_t - \alpha \nabla f(W_t; \xi_t)$. Here, α is the learning rate and ξ_t denotes a sample randomly drawn in iteration t. To implement SGD on AIMC hardware,

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

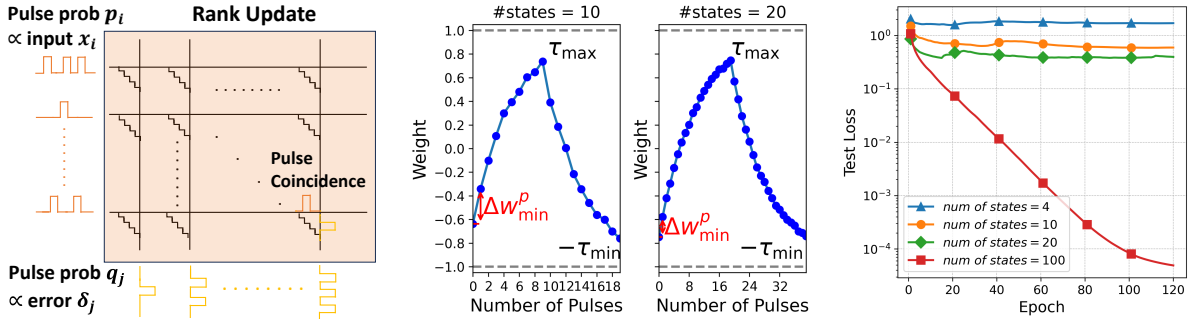


Figure 1: (left) Illustration of rank update via stochastic pulse streams. (middle) Illustration of pulsed weight updates on 10-state and 20-state softbound devices. Due to asymmetric update, the actual weight increment follows $\Delta w^p_{\min} = \Delta w_{\min} \cdot q_+(w)$, where $q_+(\cdot)$ represents the device positive response factor. (right) Training fails to converge under 4-bit conductance states. The experiment is conducted on LeNet-5 (MNIST) using Tiki-Taka.

one needs to update the weights stored in the crossbar array using the *rank update* method (Gokmen and Vlasov, 2016). This approach leverages two $\mathcal{O}(D)$-dimensional vectors, the backpropagation error δ and the input x, to perform in-memory updates directly on the analog array via stochastic pulse streams, as illustrated in Figure 1 (left). Ideally, each pulse adjusts a weight element $w \in W$ by the minimal increment $\pm\Delta w_{\min}$, with the sign determined by the pulse polarity. The resulting weight evolution is illustrated in Figure 1 (middle).

We define the number of device states by dividing the total weight range $w \in [\tau_{\min}, \tau_{\max}]$ by this minimal weight change: $n_{\text{states}} := (\tau_{\max} - \tau_{\min})/\Delta w_{\min}$, where n_{states} determines how many distinct values the weight can stably represent. A smaller n_{states} (larger $\Delta w_{\min}$) amplifying quantization noise ζ whose variance scales with $\Delta w_{\min}$. This noise captures the gap between ideal and actual updates and fundamentally limits training accuracy. Previous studies have shown that successful training on crossbar-based architectures typically requires at least 8-bit distinct conductance levels to achieve competitive accuracy (Li et al., 2018; Chen et al., 2017). However, some devices struggle to provide this level of granularity within a single memory cell. MRAM devices are typically limited to two stable states per cell, whereas ReRAM is usually constrained to 4-bit per cell in practice (detailed survey in Table 5 and Appendix A), which makes it difficult to achieve the multi-bit precision required for effective training. As illustrated in Figure 1 (right), reducing the number of states to 20 or fewer results in a convergence failure. While ECRAM can support thousands of states, it remains hindered by practical challenges, including complex three-terminal design, CMOS incompatibility, and material instability (Kim et al., 2023; Kwak et al., 2025), which lack a scalable fabrication pipeline (Kwak et al., 2024). In contrast, ReRAM remains one of the most manufacturable and scalable options (Stecconi et al., 2024). In practice, its bi-directional update behavior typically involves limited conductance states together with asymmetric non-idealities Xi et al. (2021), which form the primary focus of this paper. *Rather than* pushing for increasingly precise devices, our work advocates *algorithm innovations* to mitigate the limitations of low-state memristive devices, which better align with current fabrication capabilities and offer energy and area efficiency for near-term deployment. Importantly, our goal is not to dismiss high-state devices, but to emphasize the practical and architectural benefits of training with low-state memristive technologies.

1.1 Main results

This work addresses the fundamental challenges of limited precision in gradient-based training on AIMC hardware, which stem from the limited number of conductance states and the asymmetric update. We address these challenges by designing *composite weight representations* that integrate multiple low-precision tiles to represent high-precision weights, and by developing *multi-timescale residual learning* algorithms that enable each tile to dynamically track the residual training error left by preceding tiles. Together, these techniques ensure stable convergence and high training accuracy under low-precision constraints. This motivates our first question:

Q1) *How can high-precision weights be represented using limited-conductance states AIMC devices?*

To construct a high precision weight, we define the composite weight as $\overline{W} = \sum_{n=0}^{N} \gamma^n W^{(n)}$, where $W^{(n)}$ denotes a low precision weight on an AIMC tile n, and $\gamma \in (0, 1)$ controls its scaling. This structure increases the total number of representable values exponentially with the number of tiles, thus significantly enhancing

the effective numeric precision. The composite weight $\overline{W}$ is then used in both forward and backward passes; see the details of circuit implementation in Section 3.3. Given the composite weight $\overline{W}$, it raises another critical question:

Q2) *How to ensure that the composite weight $\overline{W}$ converges effectively under gradient-based training?*

To ensure that the composite weight $\overline{W}$ converges under gradient-based training, we propose a multi-timescale residual learning strategy inspired by the recent advances in *single-timescale stochastic approximation* (STSA) (Shen and Chen, 2022). However, different from STSA, which tracks a drifting optimum at a single timescale, our method employs a multi-timescale scheme in which each analog tile learns on a progressively slower timescale to approximate the residual left by lower-resolution tiles. This recursive refinement ensures that the composite weight $\overline{W}$ approaches the global optimum W^* with an exponentially vanishing residual error.

Our contributions. This work makes the following key contributions:

1. We propose a high-precision in-memory analog training framework termed multi-timescale residual learning, which overcomes the precision bottleneck of limited conductance states, without requiring reset operations and relying solely on an open-loop transfer process between tiles, thus simplifying hardware implementation.

2. We theoretically analyze the non-convergence of single-tile Analog SGD under realistic device constraints and establish both an upper bound and a matching lower bound. Furthermore, we analyze the convergence of our multi-timescale residual learning and show that the error reduces exponentially by increasing the number of tiles.

3. We evaluate the proposed algorithm using IBM AIHWKIT (Rasch et al., 2021) on CIFAR-100, Fashion-MNIST, and other datasets, demonstrating consistent improvements over existing in-memory analog training methods under limited conductance states.

4. We analyze hardware costs including storage, energy, latency, and area on real datasets, showing that our method achieves an accuracy–efficiency trade-off compared to baseline methods.

1.2 Related works

Gradient-based training on AIMC hardware. Gradient-based AIMC training was first explored by

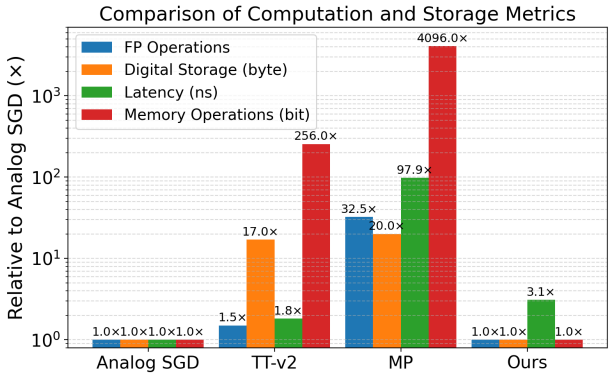


Figure 2: Algorithm comparison of computation and storage for per-sample update with model dimension $D = 32$ and mini-batch size $B = 4$ from the statistics in Table 7. MP incurs substantially higher overhead, which grows more severe as D and B increase.

rank-update methods such as Analog SGD (Gokmen and Vlasov, 2016). Tiki-Taka (TT-v1) mitigates asymmetric updates and noise by accumulating gradients on an auxiliary array and periodically transferring them (Gokmen and Haensch, 2020), while TT-v2 adds digital filtering for improved robustness (Gokmen, 2021). However, these methods often fail to converge on larger models under limited conductance states. Hybrid 3T1C–PCM designs (Ambrogio et al., 2018; Cristiano et al., 2018) improve precision through closed-loop tuning but incur high latency and area overhead (detailed in Appendix A). Another line of *hybrid training paradigms* program high-precision digital gradients into analog weights (MP (Le Gallo et al., 2018)), with momentum-based extensions (Wang et al., 2020; Huang et al., 2020), though these incur substantial digital storage and compute cost, as compared in Figure 2.

Multi-sequence stochastic approximation. Stochastic approximation with multiple coupled sequences Yang et al. (2019); Shen and Chen (2022); Huang et al. (2025) has found broad applications in machine learning such as bilevel learning Lu (2023); Jiang et al. (2024) and reinforcement learning Zeng and Doan (2024a,b). Our analog training on multiple tiles can be naturally viewed as a system of coupled sequences, since the gradient is computed on all tiles jointly, and in turn, each tile's update is driven by this gradient to ensure that the composite weight converges to the global optimum.

Low-precision computing. Existing works have shown how low-precision devices can be combined to achieve high-precision computation in *static settings* such as scientific computing and DNN inference (e.g., bit-slicing schemes (Feinberg et al., 2018; Song et al., 2023, 2024; Le Gallo et al., 2022; Pedretti et al., 2021; Boybat et al., 2018; Mackin et al., 2022)). However,

extending this concept to high-precision training on multiple low-precision tiles is far more challenging, as it must maintain convergence in the presence of asymmetric updates while weights are continually changing under limited-precision gradients. Alternative precision enhancement strategies fall under hardware-aware training, which incorporates quantization and other hardware noise into the digital training process to improve inference accuracy when weights are deployed on non-ideal analog devices (Klachko et al., 2019; He et al., 2019; Büchel et al., 2025), in contrast to our in-memory analog training that updates weights directly on analog hardware.

2 TRAINING DYNAMICS ON LIMITED-PRECISION AIMC HARDWARE

In this section, we analyze how two critical non-idealities in AIMC hardware, *limited conductance states* and *asymmetric updates*, affect the training dynamics. We derive an analog update rule that captures these effects, and show that under this rule, Analog SGD exhibits an asymptotic error determined by both the gradient noise and the quantization noise.

Asymmetric pulse update. Rank-update-based training updates the weights on the crossbar array by simultaneously applying two stochastic pulse streams to its rows and columns, with weight increments occurring at pulse coincidences. Ideally, each pulse coincidence induces a minimal weight change $\Delta w_{\min}$. However, practical updates depend nonlinearly on both the current weight and the pulse polarity, causing deviations from this idealized increment and resulting in asymmetric updates. Specifically, given the weight $W_t \in \mathbb{R}^{D\times D}$ at iteration t, the asymmetric pulse update for an element w_t is modeled as: $w_{t+1} = \begin{cases} w_t + \Delta w_{\min}\cdot q_+(w_t), & \textit{for a positive pulse,} \\ w_t + \Delta w_{\min}\cdot q_-(w_t), & \textit{for a negative pulse,} \end{cases}$ where $q_+(w)$ and $q_-(w)$ denote the device response factors to positive and negative pulses, respectively. Following the decomposition introduced in (Gokmen and Haensch, 2020), we define the symmetric and asymmetric components as $F(w) := \frac{q_-(w)+q_+(w)}{2}$ and $G(w) := \frac{q_-(w)-q_+(w)}{2}$, yielding a compact element-wise update form triggered by each pulse coincidence: $w_{t+1} = w_t + \Delta w_{\min}\odot F(w_t) - |\Delta w_{\min}|\odot G(w_t)$.

Quantization noise from limited conductance states. During the rank update process (see Figure 1 (left)), each weight element w_{ij}, located at column i and row j of the crossbar array, is updated by $\alpha x_i\delta_j$, where x_i is the i-th entry of the input vector x, δ_j is the j-th entry of the backpropagated error vector δ, and α is the learning rate. We implement the update using stochastic pulse streams, where the amplitude of each pulse is generated from a Bernoulli distribution with parameters $p_i \propto x_i$ and $q_j \propto \delta_j$. This scheme guarantees that the expectation of the actual weight change Δw_{ij} is equal to the ideal update $\alpha x_i\delta_j$. However, due to the limited number of conductance states, each pulse induces only a discrete weight increment of magnitude $\Delta w_{\min}$. This discretization introduces a mismatch between the actual update and its ideal target. We capture this discrepancy by defining a stochastic noise term ζ_{ij}, such that $\Delta w_{ij} = \alpha x_i\delta_j + \zeta_{ij}$, and show its statistical properties in the following.

Lemma 1 (Statistical properties of pulse update noise)**.** *Under the stochastic pulse update in (Gokmen and Vlasov, 2016), the random variable ζ_{ij} has the following properties:*

$$\mathbb{E}[\zeta_{ij}] = 0, \quad \textit{and} \quad \mathrm{Var}[\zeta_{ij}] = \Theta(\alpha\cdot\Delta w_{\min}).$$

The proof of Lemma 1 is deferred to Appendix D.1. Since analog crossbar arrays update all weight elements in parallel, the matrix update rule combining asymmetric pulse updates and quantization noise from limited conductance states can be succinctly represented as:

$$W_{t+1} = W_t + \Delta W_t \odot F(W_t) - |\Delta W_t|\odot G(W_t) + \zeta_t \tag{2}$$

where the operations $|\cdot|$ and $\odot$ denote element-wise absolute value and multiplication, respectively. The specific form of ΔW_t depends on the chosen optimization algorithm. By substituting ΔW_t with the gradient used in digital SGD, the update rule for Analog SGD under (2) becomes:

$$\begin{aligned} W_{t+1} = & W_t - \alpha\nabla f(W_t;\xi_t)\odot F(W_t) \\ & - |\alpha\nabla f(W_t;\xi_t)|\odot G(W_t) + \zeta_t. \end{aligned} \tag{3}$$

Based on (3), we establish the upper and lower bounds on the convergence of Analog SGD on a single tile with limited conductance states. Our analysis shows that asymmetric pulse responses and the quantization noise term ζ_t arising from limited conductance states pose fundamental challenges to convergence and lead to a non-negligible asymptotic error.

Theorem 1 (Convergence of Analog SGD, short version)**.** *Under a set of mild assumptions, with σ^2 denoting the variance bound of the gradient noise, if the learning rate is set as $\alpha = \mathcal{O}\big(\sqrt{\frac{1}{T}}\big)$, then it holds that:*

$$E_T \le \mathcal{O}\Big(\sqrt{\frac{\sigma^2}{T}}\Big) + 4\sigma^2 S_T + R_T\Delta w_{\min}$$

where $E_T := \frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|W^* - W_t\|^2]$, $S_T := \frac{1}{T}\sum_{t=0}^{T-1}\frac{\|W_t\|_\infty^2/\tau_{\max}^2}{1-\|W_t\|_\infty^2/\tau_{\max}^2}$, $R_T := \frac{1}{T}\sum_{t=0}^{T-1}\frac{2L}{1-\|W_t\|_\infty^2/\tau_{\max}^2}$.

Theorem 1 suggests that the average squared Euclidean distance between W_t and W^* is upper bounded by the sum of three terms: the first term vanishes at a rate of $\mathcal{O}(\sqrt{\sigma^2/T})$, which also appears in the digital SGD's convergence bound; the second and third terms contribute to the *asymptotic error* of Analog SGD, which does not vanish as the number of iterations T increases. Intuitively, the second term arises from the absolute gradient term $|\alpha\nabla f(W_t,\xi_t)|$ in (3), which introduces variance scaling as $\alpha\sigma^2$. The third term originates from the quantization noise ζ_t, which has variance $\Theta(\alpha\Delta w_{\min})$. In the convergence analysis, after normalizing by the descent coefficient α, these two terms result in residual errors of order σ^2 and $\Delta w_{\min}$, respectively. In contrast, in digital SGD, the variance of sample noise scaling as $\alpha^2\sigma^2$ vanishes under diminishing learning rates. We next present a matching lower bound.

Theorem 2 (Lower bound of the error of Analog SGD, short version)**.** *Under a set of mild assumptions, if the learning rate* $\alpha = \frac{1}{2L}$*, there exists an instance where Analog SGD generates a sequence* $\{W_t\}_{t=0}^{T-1}$ *such that the iterates converge to a neighborhood of the optimal solution* W^**, satisfying:*

$$E_T \geq \Omega(\sigma^2 S_T + R_T\Delta w_{\min}).$$

The full versions of Theorem 1 and 2 with their proofs are deferred to Appendix E. These theoretical insights underscore the importance of addressing quantization noise, which stands as a key obstacle to fully realizing the potential of analog neural network training.

3 MULTI-TILE RESIDUAL LEARNING ON NON-IDEAL HARDWARE

As discussed in Section 2, single-tile training on non-ideal AIMC hardware inevitably results in a non-vanishing error. We propose a multi-timescale residual learning strategy for non-ideal AIMC hardware, where each additional scaled tile iteratively corrects the *asymptotic residual error* left by the preceding lower-precision tiles due to limited conductance states and asymmetric updates.

3.1 Multi-tile residual learning formulation

Denote the weights stored on a single analog tile at iteration t by $W_t^{(0)}$, the optimal weight by W^*, and the non-vanishing error as $E := \lim_{t\to\infty} W^* - W_t^{(0)}$. To mitigate this error, we introduce a second analog tile $W^{(1)}$, scaled by a factor γ, to iteratively compensate for it. As illustrated in Figure 3 (left), rather than directly approximating E, the second tile approximates a scaled target E/γ. Although $W^{(1)}$ still suffers from similar device non-idealities and incurs a non-vanishing error when tracking its target (i.e., $\lim_{t\to\infty} E/\gamma - W_t^{(1)} = E$), the combined output of the two tiles nonetheless converges to a smaller residual:

$$\begin{aligned}&\lim_{t\to\infty}\Big(W^* - (W_t^{(0)} + \gamma W_t^{(1)})\Big)\\ &= \lim_{t\to\infty}(W^* - W_t^{(0)}) - \gamma W_t^{(1)}\\ &= \lim_{t\to\infty} E - \gamma W_t^{(1)} = \gamma E.\end{aligned}$$

This shows that the use of an additional tile reduces the asymptotic residual by a factor of γ. Extending this idea further, we introduce N more analog tiles $W^{(1)},\dots,W^{(N)}$, each tile $W^{(n)}$ is scaled by a geometric factor γ^n. We define the geometric sum of the first n tiles as $\overline{W}^{(n)} := \sum_{n'=0}^{n-1}\gamma^{n'}W^{(n')}, n\in\{1,\dots N\}$, so that the residual left by the first n tiles is $W^* - \overline{W}^{(n)}$. We define the local optimal point for tile $W^{(n)}$ as $P_n^*(\overline{W}^{(n)}) := \arg\min_{P_n} f(\overline{W}^{(n)} + \gamma^n P_n)$. Assuming that $f(\cdot)$ is strongly convex with a unique minimizer W^*, the optimal solution is $P_n^*(\overline{W}^{(n)}) = \gamma^{-n}(W^* - \overline{W}^{(n)})$, with $P_0^* := W^*$. To optimize each tile $W^{(n)}$, we minimize the objective $\|W^{(n)} - P_n^*(\overline{W}^{(n)})\|^2$, so that $\gamma^n W^{(n)}$ approximates the residual left by the first n tiles. Applying this process to all tiles finally yields an exponentially reduced error between the composite weight $\overline{W}$ and the optimal weight W^*. Formally, we solve the multi-layer problem as:

$$W^{(0)} := \arg\min_{U_0}\|U_0 - P_0^*\|^2, \quad P_0^* := W^* \tag{4a}$$

$$W^{(1)} := \arg\min_{U_1}\|U_1 - P_1^*(\overline{W}^{(1)})\|^2, \tag{4b}$$
$$\text{s.t. } P_1^*(\overline{W}^{(1)}) := \arg\min_{P_1} f(\overline{W}^{(1)} + \gamma P_1)$$
$$\cdots\cdots$$

$$W^{(N)} := \arg\min_{U_N}\|U_N - P_N^*(\overline{W}^{(N)})\|^2, \tag{4c}$$
$$\text{s.t. } P_N^*(\overline{W}^{(N)}) := \arg\min_{P_N} f(\overline{W}^{(N)} + \gamma^N P_N)$$

where $U_n, P_n \in \mathbb{R}^{D\times D}$ for $n\in\{0,\dots,N\}$.

3.2 Multi-tile gradient-based update

The optimization problem (4) is challenging because the drifting optimum $P_n^*(\overline{W}^{(n)})$ is an implicit function of $\overline{W}^{(n)}$. To decouple this dependency when optimizing $W^{(n)}$, we freeze tiles $\{W^{(0)},\dots,W^{(n-1)}\}$ to ensure

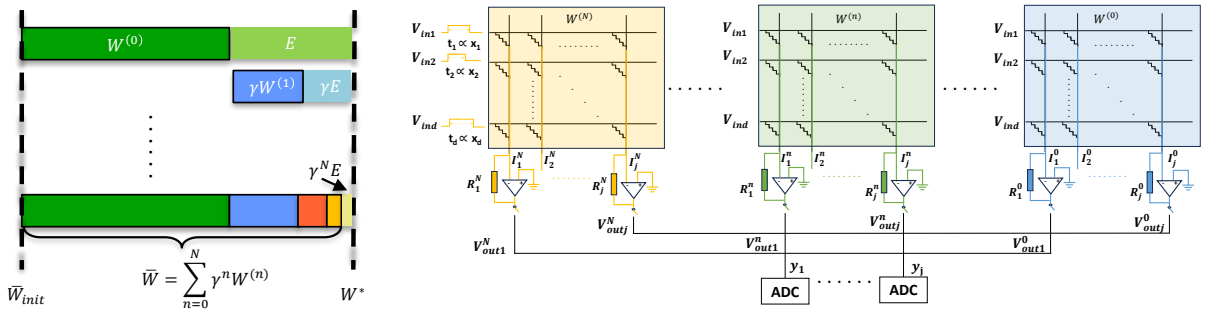


Figure 3: (left) Illustration of residual learning, where $\overline{W}_{init}$ denotes the initial composite weight. (right) Circuit implementation of forward process using composite weight $\overline{W}$.

that $\overline{W}^{(n)}$ remains fixed. To optimize each tile $W^{(n)}$ in the (4), we aim to update via an approximate descent direction of its objective $\|W^{(n)} - P_n^*(\overline{W}^{(n)})\|^2$. The negative gradient of the objective is :

$$-\nabla_{W^{(n)}} \|W^{(n)} - P_n^*(\overline{W}^{(n)})\|^2 = 2(P_n^*(\overline{W}^{(n)}) - W^{(n)}) \tag{5}$$

which implies that $P_n^*(\overline{W}^{(n)}) - W^{(n)}$ is the descent direction to update $W^{(n)}$. For $n = N$, since

$$P_N^*(\overline{W}^{(N)}) - W^{(N)} = \gamma^{-N}(W^* - \overline{W}^{(N)}) - W^{(N)}$$
$$= \gamma^{-N}(W^* - \sum_{n=0}^{N} \gamma^n W^{(n)}) = \gamma^{-N}(W^* - \overline{W})$$

and $\mathbb{E}_\xi[-\nabla f(\overline{W};\xi)] = -\nabla f(\overline{W}) \propto W^* - \overline{W}$ under strong convexity assumptions, we directly use the stochastic gradient $-\nabla f(\overline{W};\xi)$ as the descent direction to update $W^{(N)}$. For $n \in \{0,\dots,N-1\}$, since the optimization on $W^{(n+1)}$ (see (4)) ensures that $W^{(n+1)} \approx P_{n+1}^*(\overline{W}^{(n+1)}) = \gamma^{-1}(P_n^*(\overline{W}^{(n)}) - W^{(n)})$, we use $W^{(n+1)}$ to update $W^{(n)}$. Following the update rule in (2), for tile $W^{(N)}$, the update is given by:

$$W_{t+1}^{(N)} = W_t^{(N)} - \alpha \nabla f(\overline{W}_t;\xi_t) \odot F\big(W_t^{(N)}\big) - \big|\alpha \nabla f(\overline{W}_t;\xi_t)\big| \odot G\big(W_t^{(N)}\big) + \zeta_t. \tag{6}$$

For $W^{(n)}$, $n \in \{0,\dots,N-1\}$, the update is given by:

$$W_{t_n+1}^{(n)} = W_{t_n}^{(n)} + \beta \tilde{W}^{(n+1)} \odot F\big(W_{t_n}^{(n)}\big) - \big|\beta \tilde{W}^{(n+1)}\big| \odot G\big(W_{t_n}^{(n)}\big) + \zeta_{t_n} \tag{7}$$

where β is the learning rate, and the transferred weight is defined as $\tilde{W}^{(n+1)} := W_{t_{n+1}+T_{n+1}-1}^{(n+1)}$. We show in Section 4 that each tile $W^{(n)}$ requires an inner loop of $T_n = \Theta(\gamma^{-1})$ steps to converge to its optimum $P_n^*(\overline{W}^{(n)})$. We thus adopt a multi-timescale training schedule to coordinate these updates, with each tile $W^{(n)}$ maintains a local step counter $t_n = \lfloor (t+1)/\prod_{n'=n+1}^{N} T_{n'} \rfloor$. A detailed algorithm is provided in Algorithm 1.

Remark 1. *The optimization problem in* (4) *resembles the STSA framework (Shen and Chen, 2022), where each sequence tracks a drifting optimum that evolves with the updates of other sequences, denoted as* $y^{n,*}(y^{n-1})$ *in STSA and* $P_n^*(\overline{W}^{(n)})$ *in our setting. However, directly applying STSA to our problem encounters two main difficulties:* ***C1)*** *STSA relies on rapid convergence of each sequence to its drifting optimum via a single update step, but in our composite weight structure, a single-step update on* $W^{(n-1)}$ *causes* $P_n^*(\overline{W}^{(n)})$ *to drift approximately* $\Theta(\gamma^{-1})$ *times faster than a single-step update on* $W^{(n)}$*;* ***C2)*** *STSA considers that* $y^{n,*}(y^{n-1})$ *depends only on sequence* y^{n-1}*, while our scenario involves* $P_n^*(\overline{W}^{(n)})$ *depending on multiple sequences* $\overline{W}^{(n)}$*.*

Remark 2. *Our update dynamics for each tile naturally supports an open-loop transfer process. As shown in* (5)*, what needs to be propagated is only the descent direction of each tile rather than its exact weight value. This eliminates the need for closed-loop tuning, thereby reducing control overhead and highlighting the hardware efficiency of our algorithm.*

3.3 Analog circuit implementation

Figure 3 (right) illustrates how the composite weight $\overline{W}$ is formed in the analog domain by combining low-precision tiles $W^{(n)} \in \mathbb{R}^{D\times D}$, $n \in \{0,\dots,N\}$. For the forward pass $y = x^\top \overline{W}$, each input x_d is encoded by a voltage pulse on the d-th row of each tile $W^{(n)}$, with duration proportional to x_d. By Ohm's and Kirchhoff's laws, each tile produces a current of the j-th column as $I_j^n = \sum_{d=1}^{D} W_{d,j}^{(n)} x_d$. Each I_j^n is fed into an inverting op-amp with feedback resistor R_j^n to apply scaling γ^n, yielding $V_{\text{out}j}^n = -R_j^n I_j^n$. The voltage outputs are summed in hardware to produce the final result y_j as:

$$y_j \propto \sum_{n=0}^{N} V_{\text{out}j}^n = -\sum_{n=0}^{N} R_j^n I_j^n$$

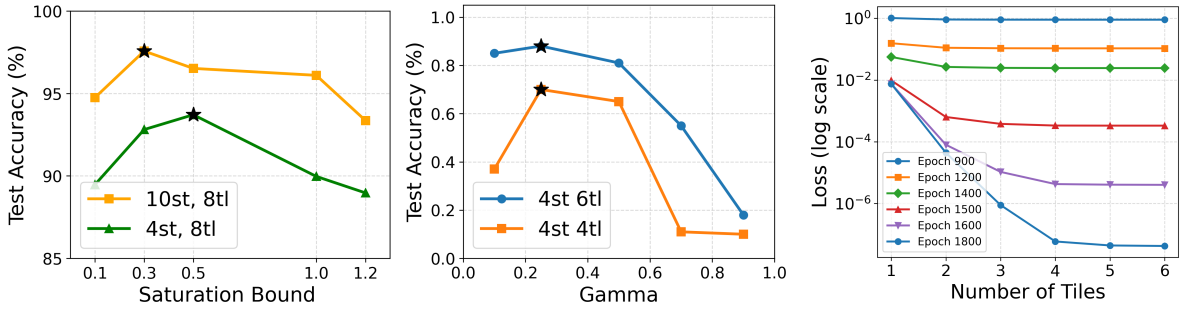


Figure 4: (left) Effect of asymmetry. st: #states, tl: #tiles. (middle) Effect of geometric scaling factor γ. (right) A toy example illustrating that training loss decreases along both the epoch and tile-count dimensions. The results demonstrate that each tile converges successfully, meanwhile more tiles also improves accuracy.

$$= -\sum_{n=0}^{N} R_j^n \sum_{d=1}^{D} W_{d,j}^{(n)} x_d = -\sum_{d=1}^{D} \overline{W}_{d,j} x_d.$$

A similar operation is used during the backward pass, where the output is given by $\delta = \overline{W}^\top \delta'$, with δ' denoting the error signal propagated from the next layer. Table 7 compares the computational complexity and estimated update latency of MP, Analog SGD, TT-v2, and our method. Our algorithm achieves a low update latency, upper bounded by 95.9 ns even with an infinite number of tiles, which is more than 30× faster than MP, while requiring only $\mathcal{O}(2D)$ digital storage for the input x and error δ and $\mathcal{O}(1)$ digital memory operations for reading and writing, which is comparable to the cost of Analog SGD. Please see Appendix H, where we discuss the circuit-level implementation and present a detailed comparison of digital storage, runtime, energy, and area costs across algorithms.

4 STOCHASTIC APPROXIMATION THEORY FOR RESIDUAL LEARNING

In this section, we present a proof sketch for the convergence of our proposed multi-timescale residual learning algorithm. Before analyzing the algorithm, we introduce four assumptions concerning the objective function, the gradient noise, and the device response characteristics.

Assumption 1 (Unbiasness and bounded variance)**.** *The sample ξ_t is independently sampled from a distribution $\mathcal{D}$, $\forall t \in [T]$, and the stochastic gradient is unbiased with bounded variance, i.e., $\mathbb{E}_{\xi_t}[\nabla f(W_t;\xi_t)] = \nabla f(W_t)$ and $\mathbb{E}_{\xi_t}[\|\nabla f(W_t;\xi_t) - \nabla f(W_t)\|^2] \le \sigma^2$.*

Assumption 2 (Smoothness and strong convexity)**.** *$f(W)$ is L-smooth and μ-strongly convex.*

Assumption 3 (Bounded weights)**.** *The weights are bounded as $\|W_t\|_\infty \le W_{\max} \le \tau_{\max}$ for all t.*

Assumption 4 (Response factor and zero shifted symmetric point)**.** ***(Continuity)*** *$q_+(\cdot)$ and $q_-(\cdot)$ are continuous;* ***(Saturation)*** *$q_+(\tau_{\max}) = 0$, $q_-(\tau_{\min}) = 0$;* ***(Positive-definiteness)*** *$q_+(w) > 0$ for all $w < \tau_{\max}$, and $q_-(w) > 0$ for all $w > \tau_{\min}$;* ***(Symmetric point)*** *$G(w) = 0$ if and only if $w = 0$.*

Assumptions 1–2 are standard in convex optimization (Bottou et al., 2018). Assumption 3 assumes that W_t remains within a small region, which is a mild condition that generally holds in practice. Assumption 4 defines the response function class observed in resistive devices (Wu et al., 2025) and adopts the widely used zero-shifted symmetric point in analog training (Gokmen and Haensch, 2020). We begin by presenting Lemmas 2 and 3, which describe how each tile tracks its drifting optimum within an inner loop and serve as the basis for our convergence analysis. Their full proofs are given in Appendix F. Here $P_n^*(\overline{W}_{t_{n-1}}^{(n)}) =: P_n^*$ for convenience.

Lemma 2 (Descent lemma of the main sequence $W^{(N)}$)**.** *Under Assumptions 1–4, the update dynamics* (6) *ensures that after a single inner loop of length T_N, the expected distance between $W^{(N)}$ and its optimum decreases as:*

$$\begin{aligned}&\mathbb{E}\big[\|W_{t+T_N-1}^{(N)} - P_N^*\|^2\big] \\ &\le \big(1-\Theta(\gamma^N)\big)^{T_N} \|W_t^{(N)} - P_N^*\|^2 \\ &\quad + \Theta(\sigma^2\gamma^{-N} + \gamma^{-\frac{4N}{3}}(\sigma\Delta w_{\min})^{\frac{2}{3}}).\end{aligned}$$

Lemma 3 (Descent lemma of the sequences $W^{(n)}$)**.** *Under the same assumptions as Lemma 2, for $n \in \{0,\dots,N-1\}$, the update dynamics* (7) *ensures that:*

$$\begin{aligned}\mathbb{E}\big[\|W_{t_n+T_n-1}^{(n)} - P_n^*\|^2\big] &\le (1-\Theta(\gamma))^{T_n} \|W_{t_n}^{(n)} - P_n^*\|^2 \\ &\quad + \Theta(\gamma^2\|\tilde{W}^{(n+1)} - P_{n+1}^*\|^2 + \Delta w_{\min}).\end{aligned}$$

Lemmas 2 and 3 yield contraction terms $(1-\Theta(\gamma^p))^{T_n}$ with $p = 1$ for $W^{(n)}$ and $p = N$ for $W^{(N)}$. Using $(1-\lambda)^{T_n} \le e^{-\lambda T_n}$, we set $T_n = \Theta(\gamma^{-1})$, $T_N = \Theta(\gamma^{-N})$ so that $(1-\Theta(\gamma^p))^{T_n} = \Theta(\rho)$, where ρ is the contraction rate in the Lyapunov analysis below.

Dataset	TT-v1	TT-v2	MP	Ours (3 tiles)	Ours (4 tiles)	Ours (6 tiles)
Fashion-MNIST (#4)	10.01±0.07	47.51±0.91	75.61±0.69	68.09±0.49	73.35±0.13	75.11 ±0.07
MNIST (#10)	78.65±2.36	95.43±0.17	99.13±0.02	95.07±0.35	97.10±0.17	98.53±0.09

Table 1: Test accuracy on MNIST and Fashion-MNIST with analog LeNet-5 under 10 and 4 states. Compared methods include MP, TT-v1, TT-v2 and different versions of our algorithm.

Dataset	TT-v1	TT-v2	MP	Ours (4 tiles)	Ours (6 tiles)	Ours (8 tiles)
CIFAR-10 (#4)	11.65±0.68	87.43±0.10	93.31±0.04	90.45±0.28	92.02±0.14	90.65±0.09
CIFAR-100 (#4)	9.75±2.47	11.28±0.62	70.25±0.61	63.26±0.45	68.46±0.23	69.63±0.40
CIFAR-10 (#16)	58.00±0.26	84.30±0.09	95.04±0.05	92.98±0.62	93.29±0.64	94.36±0.08
CIFAR-100 (#16)	26.69±0.17	65.17±0.36	73.16±0.07	68.11±0.35	69.62±0.53	72.20±0.11

Table 2: Test accuracy on CIFAR-10 and CIFAR-100 under 4 and 16 conductance states on ResNet-34. Compared methods include MP, TT-v1, TT-v2 and different versions of our algorithm.

To this end, we now analyze the optimization problem in (4) by introducing a Lyapunov sequence as $\mathbb{J}_k := \sum_{n=0}^{N} \|W^{(n)}_{t_n+kT_n-1} - P^*_n(\overline{W}^{(n)}_{t_{n-1}+k})\|^2$, which aggregates the squared distances between each tile and its drifting optimum after completing k inner loops. In particular, when the slowest tile $W^{(0)}$ has completed k inner loops, the fastest tile $W^{(N)}$ will have performed $t = k\prod_{n=0}^{N} T_n = \mathcal{O}(\gamma^{-2N}k)$ gradient updates, which we take as the reference measure for evaluating the overall convergence rate. By balancing the learning rates and inner-loop length across all tiles, we establish a linear convergence rate for the Lyapunov sequence with an asymptotic error induced by device imperfections. This result is formalized in the following theorem.

Theorem 3 (Convergence of residual learning)**.** *Suppose Assumptions 1–4 hold. Let the scaling parameter satisfy $\gamma \in (0, 1/\sqrt{6}]$. For all $n \in \{0, \dots, N-1\}$, set the learning rate $\beta = \Theta(\gamma^2)$ and the inner loop length $T_n \geq \Theta(\gamma^{-1})$, except for $T_0 = \Theta(1)$. For $n = N$, set the learning rate $\alpha = \Theta(1)$ and $T_N \geq \Theta(\gamma^{-N})$. Given $t = \mathcal{O}(\gamma^{-2N}k)$, $\rho \in (0,1)$, the Lyapunov function $\mathbb{J}_k$ is bounded as:*

$$\mathbb{E}[\mathbb{J}_k] \leq \mathcal{O}((1-\rho)^{\gamma^{2N}t})\mathbb{E}[\mathbb{J}_0] + \Theta(\gamma^{-\frac{4N}{3}}(\sigma\Delta w_{\min})^{\frac{2}{3}}).$$

Theorem 3 shows that the Lyapunov function converges linearly up to an asymptotic upper bound $\Theta(\gamma^{-\frac{4N}{3}}(\sigma\Delta w_{\min})^{2/3})$ determined solely by quantization and sample noise. Corollary 1 follows from Theorem 3 by applying the the bound of Lyapunov function as an upper bound on the component $\|W^{(N)}_{t_N+kT_N-1} - P^*_N(\overline{W}^{(N)}_{t_{N-1}+k})\|^2$. Then using the definition of $P^*_N(\overline{W}^{(N)})$ and multiplying both sides by γ^{2N} yields the optimality gap of the composite weight.

Corollary 1 (Optimality gap of residual learning)**.** *Under the same conditions as in Theorem 3, the limit of the composited weight $\overline{W}_t$ satisfies:*

$$\limsup_{t\to\infty} \mathbb{E}\big[\|W^* - \overline{W}_t\|^2\big] \leq \Theta(\gamma^{\frac{2N}{3}}(\sigma\Delta w_{\min})^{\frac{2}{3}}).$$

The proofs of Theorem 3 and Corollary 1 can be found in Appendix F. Intuitively, increasing the value of N reduces the upper bound $\Theta\big(\gamma^{\frac{2N}{3}}(\sigma\Delta w_{\min})^{\frac{2}{3}}\big)$, demonstrating the effectiveness of multi-timescale residual learning under a limited number of conductance states.

5 NUMERICAL SIMULATIONS

In this section, we evaluate our method using the AIHWKIT toolkit on the SoftBounds device class, which models bi-directional memristive devices by capturing saturation effects and limited update precision. We compare against the MP, TT-v1, and TT-v2 baselines. Detailed algorithms are provided in Appendix G.

5.1 Training performance on real dataset

We train an analog LeNet-5 model on the MNIST and Fashion-MNIST datasets for 100 epochs with 4 and 10 conductance states, respectively. We also train the ResNet-18 model on CIFAR-10, and the ResNet-34 model on the CIFAR-10 and CIFAR-100 datasets with 4, 10 or 16 conductance states, covering both an extreme low-precision case and the widely adopted industrial setting. Training is performed for 200 epochs in CIFAR-10 and 400 epochs in CIFAR-100, with `layer3`, `layer4`, and the fully connected layer mapped to analog. The parameter configurations for TT-v1, TT-v2, MP, and our method are provided in Appendix J. As shown in Tables 1, 2 and 3, our residual learning method steadily improves accuracy as the number of tiles increases, surpassing both TT-v1 and TT-v2 with only 3 to 4 tiles, and reaching accuracy comparable to MP while incurring far lower storage and runtime overhead, demonstrating both scalability and robustness across larger networks.

# States	TT-v1	TT-v2	MP	Ours (4 tiles)	Ours (6 tiles)	Ours (8 tiles)
4	53.83±0.14	87.89±0.06	92.77±0.05	87.35±0.08	89.79±0.14	90.45±0.09
10	84.18±0.06	89.17±0.11	93.45±0.08	90.76±0.06	91.07±0.06	90.88±0.05

Table 3: Test accuracy on CIFAR10 under 4 and 10 conductance states using analog ResNet-18. Compared methods include MP, TT-v1, TT-v2 and different versions of our algorithm.

Dataset	TT-v1	TT-v2	MP	Ours (3 tiles)	Ours (5 tiles)	Ours (7 tiles)
CIFAR-10	10.04±0.04	75.65±0.17	87.32±0.10	82.59±0.27	83.97±0.20	84.96±0.08
CIFAR-100	11.27±0.04	34.80±0.16	58.06 ±0.07	42.12±0.47	45.14±0.13	50.82 ±0.14

Table 4: Test accuracy on CIFAR-10 and CIFAR-100 using 80-state devices with `layer2`, `layer3`, `layer4`, and the fully connected layer in ResNet-34 converted to analog. Compared methods include MP, TT-v1, TT-v2 and different versions of our algorithm.

We further conduct experiments on 80-state SoftBounds devices with larger analog deployment in Table 4. We observe that as the proportion of analog layers increases, for example, when further converting `layer2` in ResNet-34 to analog, TT-v1 and TT-v2 degrades severely due to error accumulation across layers even with a higher number of conductance states. In contrast, our method maintains higher accuracy, surpassing TT baselines with only 3–4 tiles and approaching MP performance with 8 tiles. We believe this accuracy collapse is a general challenge in scaling analog training to larger models, and our algorithm offers a practical solution. We also provide additional results with extensions to NLP tasks in Appendix I, to further demonstrate the scalability of our method.

5.2 Ablation studies

Effect of asymmetry. For a given number of conductance states, the degree of asymmetry in an analog device is determined by the saturation bound $\tau_{\max}$ (with $\tau_{\min} = -\tau_{\max}$). We evaluate training accuracy with $\tau_{\max}$ varying in $(0, 1)$ in MNIST and show that our algorithm consistently maintains high accuracy at different levels of asymmetry in Figure 4 (left).

Effect of geometric scaling factor. The geometric scaling factor γ plays a critical role in determining the effectiveness of residual representation across multiple analog tiles. Intuitively, if each tile has a dynamic range $[-\tau_{\max}, \tau_{\max}]$, then γ should be chosen such that $\gamma \cdot (2\tau_{\max}) \approx \Delta w_{\min}$ to ensure that the representable range of the next tile fully lies within the resolution of the previous tile. In practice, device non-idealities such as conductance saturation and asymmetric update dynamics reduce the usable dynamic range of each tile. To account for these effects, we heuristically choose γ slightly larger than $\frac{\Delta w_{\min}}{(2\tau_{\max})} = \frac{1}{n_{states}}$. We perform an ablation study on LeNet-5 using $n_{states} = 4$, and show the corresponding ablation results in Figure 4 (middle), where the peak γ value is consistent with our hypothesis, while the large γ severely degrades accuracy. These results support the need for selection of the scaling factor based on device characteristics.

Effect of number of tiles. To validate the convergence behavior of our residual learning mechanism, we construct a toy example based on a simple least-squares problem of the form $(w - b)^2$. The target output b is quantized to 16-bit resolution, and each tile has 2-bit update granularity. The parameter configurations are provided in Appendix J. Figure 4 (right) presents the log-scaled training loss. This visualization highlights two aspects of the learning dynamics. First, the composite weight converges to the target $w^* = b$. Second, the loss decreases steadily with more tiles, confirming the effectiveness of the multi-tile strategy.

6 CONCLUSIONS AND LIMITATIONS

We propose multi-timescale residual learning, an in-memory analog training framework that enables reliable DNN training under limited conductance states by modeling device non-idealities and proving convergence, achieving strong results on standard image classification tasks. Future work will focus on further extending our framework to enhance its robustness to device-level variability and noise, and to scale the approach to large language models. We acknowledge the lack of real hardware evaluation and plan to validate our method through future chip-level experiments.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation Projects 2401297 and 2532349, by IBM through the IBM-Rensselaer Future of Computing Research Collaboration, by Cisco Research, and by an NVIDIA Academic Grant Program Award.

References

Ambrogio, S., Narayanan, P., Tsai, H., Shelby, R. M., Boybat, I., Di Nolfo, C., Sidler, S., Giordano, M., Bodini, M., Farinha, N. C., et al. (2018). Equivalent-accuracy accelerated neural-network training using analogue memory. *Nature*, 558(7708):60–67.

Apalkov, D., Khvalkovskiy, A., Watts, S., Nikitin, V., Tang, X., Lottis, D., Moon, K., Luo, X., Chen, E., Ong, A., Driskill-Smith, A., and Krounbi, M. (2013). Spin-transfer torque magnetic random access memory (stt-mram). *J. Emerg. Technol. Comput. Syst.*, 9(2).

Bottou, L., Curtis, F. E., and Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311.

Boybat, I., Le Gallo, M., Nandakumar, S., Moraitis, T., Parnell, T., Tuma, T., Rajendran, B., Leblebici, Y., Sebastian, A., and Eleftheriou, E. (2018). Neuromorphic computing with multi-memristive synapses. *Nature communications*, 9(1):2514.

Burr, G., Shelby, R., di Nolfo, C., Jang, J., Shenoy, R., Narayanan, P., Virwani, K., Giacometti, E., Kurdi, B., and Hwang, H. (2014). Experimental demonstration and tolerancing of a large-scale neural network (165,000 synapses), using phase-change memory as the synaptic weight element. In *2014 IEEE International Electron Devices Meeting*, pages 29.5.1–29.5.4.

Burr, G. W., Brightsky, M. J., Sebastian, A., Cheng, H.-Y., Wu, J.-Y., Kim, S., Sosa, N. E., Papandreou, N., Lung, H.-L., Pozidis, H., Eleftheriou, E., and Lam, C. H. (2016). Recent progress in phase-change memory technology. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 6(2):146–162.

Büchel, J., Chalas, I., Acampa, G., Chen, A., Fagbohungbe, O., Tsai, S., Maghraoui, K. E., Gallo, M. L., Rahimi, A., and Sebastian, A. (2025). Analog foundation models. *arXiv preprint arXiv:2505.09663*.

Chang, T., Jo, S.-H., and Lu, W. (2011). Short-term memory to long-term memory transition in a nanoscale memristor. *ACS nano*, 5(9):7669–7676.

Chen, M.-C., Ohshita, S., Amano, S., Kurokawa, Y., Watanabe, S., Imoto, Y., Ando, Y., Hsieh, W.-H., Chang, C.-H., Wu, C.-C., Chuang, S.-S., Yoshida, H., Lu, M.-C., Liao, M.-H., Chang, S.-Z., and Yamazaki, S. (2022). A $>$ 64 multiple states and $>$ 210 tops/w high efficient computing by monolithic si/caac-igzo + super-lattice zro2/al2 o3/zro2 for ultra-low power edge ai application. In *2022 International Electron Devices Meeting (IEDM)*, pages 18.2.1–18.2.4.

Chen, P.-Y., Peng, X., and Yu, S. (2017). Neurosim+: An integrated device-to-algorithm framework for benchmarking synaptic devices and array architectures. In *2017 IEEE International Electron Devices Meeting (IEDM)*, pages 6.1.1–6.1.4.

Cristiano, G., Giordano, M., Ambrogio, S., Romero, L. P., Cheng, C., Narayanan, P., Tsai, H., Shelby, R. M., and Burr, G. W. (2018). Perspective on training fully connected networks with resistive memories: Device requirements for multiple conductances of varying significance. *Journal of Applied Physics*, 124(15).

Feinberg, B., Vengalam, U. K. R., Whitehair, N., Wang, S., and Ipek, E. (2018). Enabling scientific computing on memristive accelerators. In *2018 ACM/IEEE 45th Annual International Symposium on Computer Architecture (ISCA)*, pages 367–382.

Fuller, E. J., Keene, S. T., Melianas, A., Wang, Z., Agarwal, S., Li, Y., Tuchman, Y., James, C. D., Marinella, M. J., Yang, J. J., et al. (2019). Parallel programming of an ionic floating-gate memory array for scalable neuromorphic computing. *Science*, 364(6440):570–574.

Gokmen, T. (2021). Enabling training of neural networks on noisy hardware. *Frontiers in Artificial Intelligence*, 4:699148.

Gokmen, T. and Haensch, W. (2020). Algorithm for training neural networks on resistive device arrays. *Frontiers in neuroscience*, 14:103.

Gokmen, T. and Vlasov, Y. (2016). Acceleration of deep neural network training with resistive cross-point devices: Design considerations. *Frontiers in neuroscience*, 10:333.

Gong, N., Rasch, M. J., Seo, S.-C., Gasasira, A., Solomon, P., Bragaglia, V., Consiglio, S., Higuchi, H., Park, C., Brew, K., et al. (2022). Deep learning acceleration in 14nm cmos compatible reram array: device, material and algorithm co-optimization. In *2022 International Electron Devices Meeting (IEDM)*, pages 33–7. IEEE.

He, Z., Lin, J., Ewetz, R., Yuan, J.-S., and Fan, D. (2019). Noise injection adaption: End-to-end reram crossbar non-ideal effect adaption for neural network mapping. In *Proceedings of the 56th Annual Design Automation Conference 2019*.

Hu, M., Strachan, J. P., Li, Z., Grafals, E. M., Davila, N., Graves, C., Lam, S., Ge, N., Yang, J. J., and Williams, R. S. (2016). Dot-product engine for neuromorphic computing: programming 1t1m crossbar to accelerate matrix-vector multiplication. In *Proceedings of the 53rd Annual Design Automation Conference*.

Huang, S., Sun, X., Peng, X., Jiang, H., and Yu, S. (2020). Overcoming challenges for achieving high in-situ training accuracy with emerging memories. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1025–1030.

Huang, Y., Wu, Z., Ma, S., and Ling, Q. (2025). Single-timescale multi-sequence stochastic approximation without fixed point smoothness: Theories and applications. *IEEE Transactions on Signal Processing*, 73:1939–1953.

Jain, S., Tsai, H., Chen, C.-T., Muralidhar, R., Boybat, I., Frank, M. M., Woźniak, S., Stanisavljevic, M., Adusumilli, P., Narayanan, P., Hosokawa, K., Ishii, M., Kumar, A., Narayanan, V., and Burr, G. W. (2023). A heterogeneous and programmable compute-in-memory accelerator architecture for analog-ai using dense 2-d mesh. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 31(1):114–127.

Jia, H., Ozatay, M., Tang, Y., Valavi, H., Pathak, R., Lee, J., and Verma, N. (2022). Scalable and programmable neural network inference accelerator based on in-memory computing. *IEEE J. Solid-State Circuits*, 57(1):198–211.

Jiang, H., Han, L., Lin, P., Wang, Z., Jang, M. H., Wu, Q., Barnell, M., Yang, J. J., Xin, H. L., and Xia, Q. (2016). Sub-10 nm ta channel responsible for superior performance of a hfo2 memristor. *Scientific reports*, 6(1):28525.

Jiang, L., Xiao, Q., Tenorio, V. M., Real-Rojas, F., Marques, A. G., and Chen, T. (2024). A primal-dual-assisted penalty approach to bilevel optimization with coupled constraints. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*.

Joshi, V., Le Gallo, M., Haefeli, S., Boybat, I., Nandakumar, S. R., Piveteau, C., Dazzi, M., Rajendran, B., Sebastian, A., and Eleftheriou, E. (2020). Accurate deep neural network inference using computational phase-change memory. *Nature communications*, 11(1):2473.

Kim, H., Seo, J., Cho, S., Jeon, S., Woo, J., and Lee, D. (2023). Three-dimensional vertical structural electrochemical random access memory for high-density integrated synapse device. *Scientific Reports*, 13(1):14325.

Kim, S., Todorov, T., Onen, M., Gokmen, T., Bishop, D., Solomon, P., Lee, K.-T., Copel, M., Farmer, D. B., Ott, J. A., Ando, T., Miyazoe, H., Narayanan, V., and Rozen, J. (2019). Metal-oxide based, cmos-compatible ecram for deep learning accelerator. In *2019 IEEE International Electron Devices Meeting (IEDM)*, pages 35.7.1–35.7.4.

Klachko, M., Mahmoodi, M. R., and Strukov, D. (2019). Improving noise tolerance of mixed-signal neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Kwak, H., Choi, J., Han, S., Kim, E. H., Kim, C., Solomon, P., Lee, J., Kim, D., Shin, B., Lee, D., et al. (2025). Unveiling ecram switching mechanisms using variable temperature hall measurements for accelerated ai computation. *Nature Communications*, 16(1):2715.

Kwak, H., Kim, N., Jeon, S., Kim, S., and Woo, J. (2024). Electrochemical random-access memory: recent advances in materials, devices, and systems towards neuromorphic computing. *Nano convergence*, 11(1):9.

Le Gallo, M., Nandakumar, S., Ciric, L., Boybat, I., Khaddam-Aljameh, R., Mackin, C., and Sebastian, A. (2022). Precision of bit slicing with in-memory computing based on analog phase-change memory crossbars. *Neuromorphic Computing and Engineering*, 2(1):014009.

Le Gallo, M., Sebastian, A., Mathis, R., Manica, M., Giefers, H., Tuma, T., Bekas, C., Curioni, A., and Eleftheriou, E. (2018). Mixed-precision in-memory computing. *Nature Electronics*, 1(4):246–253.

Li, Y., Kim, S., Sun, X., Solomon, P., Gokmen, T., Tsai, H., Koswatta, S., Ren, Z., Mo, R., Yeh, C. C., Haensch, W., and Leobandung, E. (2018). Capacitor-based cross-point array for analog neural network with record symmetry and linearity. In *2018 IEEE Symposium on VLSI Technology*, pages 25–26.

Lu, S. (2023). Bilevel optimization with coupled decision-dependent distributions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22758–22789. PMLR.

Mackin, C., Rasch, M. J., Chen, A., Timcheck, J., Bruce, R. L., Li, N., Narayanan, P., Ambrogio, S., Le Gallo, M., Nandakumar, S., et al. (2022). Optimised weight programming for analogue memory-based deep neural networks. *Nature communications*, 13(1):3765.

Murmann, B. (2021). Mixed-signal computing for deep neural network inference. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 29(1):3–13.

Nandakumar, S., Le Gallo, M., Piveteau, C., Joshi, V., Mariani, G., Boybat, I., Karunaratne, G., Khaddam-Aljameh, R., Egger, U., Petropoulos, A., et al. (2020). Mixed-precision deep learning based on

computational memory. *Frontiers in neuroscience*, 14:406.

Nesterov, Y. et al. (2018). *Lectures on convex optimization*, volume 137. Springer.

Park, S., Sheri, A., Kim, J., Noh, J., Jang, J., Jeon, M., Lee, B., Lee, B. R., Lee, B. H., and Hwang, H. (2013). Neuromorphic speech systems using advanced reram-based synapse. In *2013 IEEE International Electron Devices Meeting*, pages 25.6.1–25.6.4.

Pedretti, G., Ambrosi, E., and Ielmini, D. (2021). Conductance variations and their impact on the precision of in-memory computing with resistive switching memory (rram). In *2021 IEEE International Reliability Physics Symposium (IRPS)*, pages 1–8. IEEE.

Rasch, M. J., Carta, F., Fagbohungbe, O., and Gokmen, T. (2023). Fast offset corrected in-memory training. *arXiv preprint arXiv:2303.04721*.

Rasch, M. J., Carta, F., Fagbohungbe, O., and Gokmen, T. (2024). Fast and robust analog in-memory deep neural network training. *Nat. Commun.*, 15(1):7133.

Rasch, M. J., Moreda, D., Gokmen, T., Le Gallo, M., Carta, F., Goldberg, C., El Maghraoui, K., Sebastian, A., and Narayanan, V. (2021). A flexible and fast pytorch toolkit for simulating training and inference on analog crossbar arrays. In *2021 IEEE 3rd international conference on artificial intelligence circuits and systems (AICAS)*, pages 1–4. IEEE.

Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S., and Kepner, J. (2022). Ai and ml accelerator survey and trends. In *2022 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–10.

Rzeszut, P., Chęciński, J., Brzozowski, I., Ziętek, S., Skowroński, W., and Stobiecki, T. (2022). Multistate mram cells for hardware neuromorphic computing. *Scientific reports*, 12(1):7178.

Shen, H. and Chen, T. (2022). A single-timescale analysis for stochastic approximation with multiple coupled sequences. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*.

Song, L., Chen, F., Li, H., and Chen, Y. (2023). Refloat: Low-cost floating-point processing in reram for accelerating iterative linear solvers. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*.

Song, W., Rao, M., Li, Y., Li, C., Zhuo, Y., Cai, F., Wu, M., Yin, W., Li, Z., Wei, Q., et al. (2024). Programming memristor arrays with arbitrarily high precision for analog computing. *Science*, 383(6685):903–910.

Stathopoulos, S., Khiat, A., Trapatseli, M., Cortese, S., Serb, A., Valov, I., and Prodromakis, T. (2017). Multibit memory operation of metal-oxide bi-layer memristors. *Scientific reports*, 7(1):17532.

Stecconi, T., Bragaglia, V., Rasch, M. J., Carta, F., Horst, F., Falcone, D. F., Ten Kate, S. C., Gong, N., Ando, T., Olziersky, A., et al. (2024). Analog resistive switching devices for training deep neural networks with the novel tiki-taka algorithm. *Nano Letters*, 24(3):866–872.

Stuecheli, J. et al. (2013). Next generation power microprocessor. HotChips.

Tang, J., Bishop, D., Kim, S., Copel, M., Gokmen, T., Todorov, T., Shin, S., Lee, K.-T., Solomon, P., Chan, K., Haensch, W., and Rozen, J. (2018). Ecram as scalable synaptic cell for high-speed, low-power neuromorphic computing. In *2018 IEEE International Electron Devices Meeting (IEDM)*, pages 13.1.1–13.1.4.

Vasilopoulos, A., Büchel, J., Kersting, B., Lammie, C., Brew, K., Choi, S., Philip, T., Saulnier, N., Narayanan, V., Le Gallo, M., and Sebastian, A. (2023). Exploiting the state dependency of conductance variations in memristive devices for accurate in-memory computing. *IEEE Transactions on Electron Devices*, 70(12):6279–6285.

Wang, Y., Wu, S., Tian, L., and Shi, L. (2020). Ssm: a high-performance scheme for in situ training of imprecise memristor neural networks. *Neurocomputing*, 407:270–280.

Woo, J., Moon, K., Song, J., Lee, S., Kwak, M., Park, J., and Hwang, H. (2016). Improved synaptic behavior under identical pulses using alox/hfo2 bilayer rram array for neuromorphic systems. *IEEE Electron Device Letters*, 37(8):994–997.

Wu, Z., Gokmen, T., Rasch, M. J., and Chen, T. (2024). Towards exact gradient-based training on analog in-memory computing. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*.

Wu, Z., Xiao, Q., Gokmen, T., Fagbohungbe, O., and Chen, T. (2025). Analog in-memory training on general non-ideal resistive elements: The impact of response functions. *arXiv preprint arXiv:2502.06309*.

Xi, Y., Gao, B., Tang, J., Chen, A., Chang, M.-F., Hu, X. S., Spiegel, J. V. D., Qian, H., and Wu, H. (2021). In-memory learning with analog resistive switching memory: A review and perspective. *Proceedings of the IEEE*, 109(1):14–42.

Xu, J., Liu, H., Duan, Z., Liao, X., Jin, H., Yang, X., Li, H., Liu, C., Mao, F., and Zhang, Y. (2024a). Reharvest: An adc resource-harvesting crossbar architecture for reram-based dnn accelerators. *ACM Trans. Archit. Code Optim.*, 21(3).

Xu, J., Liu, H., Peng, X., Duan, Z., Liao, X., and Jin, H. (2024b). A cascaded reram-based crossbar architecture for transformer neural network acceleration. *ACM Trans. Des. Autom. Electron. Syst.*, 30(1).

Yang, S., Wang, M., and Fang, E. X. (2019). Multilevel stochastic gradient methods for nested composition optimization. *SIAM Journal on Optimization*, 29(1):616–659.

Zeng, S. and Doan, T. (2024a). Fast two-time-scale stochastic gradient method with applications in reinforcement learning. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 5166–5212.

Zeng, S. and Doan, T. T. (2024b). Accelerated multi-time-scale stochastic approximation: Optimal complexity and applications in reinforcement learning and multi-agent games. *arXiv preprint arXiv:2409.07767*.

Zhang, J., Wang, Z., and Verma, N. (2017). In-memory computation of a machine-learning classifier in a standard 6t sram array. *IEEE J. Solid-State Circuits*, 52(4):915–924.

CHECKLIST

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Detailed in Section 4.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] Detailed in Section 3.3 and Appendix H.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Detailed in Appendix J.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] Detailed in Section 4.
   (b) Complete proofs of all theoretical results. [Yes] Detailed in Appendix E and F.
   (c) Clear explanations of any assumptions. [Yes] Detailed in Section 4.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] Detailed in Appendix J.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Detailed in Appendix J.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] Detailed in Appendix J.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] Detailed in Appendix J.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] We cite IBM AIHWKIT (Rasch et al., 2021) in the main text.
   (b) The license information of the assets, if applicable. [Yes] Detailed in Appendix J.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes] Detailed in Appendix J.
   (d) Information about consent from data providers/curators. [Yes] Detailed in Appendix J.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Supplementary Materials

A LITERATURE REVIEW

This section briefly reviews the literature related to this paper, as a complement to Section 1.2.

Gradient-based training on AIMC hardware. Since the introduction of *rank-update-based* Analog SGD (Gokmen and Vlasov, 2016), various techniques have been proposed to improve its robustness under non-ideal device behavior. To mitigate asymmetric updates and noise accumulation in Analog SGD, TT-v1 uses an auxiliary array to accumulate a moving average of gradients, periodically transferring them to the main array (Gokmen and Haensch, 2020). TT-v2 enhances this approach with digital filtering to improve robustness against noise and low conductance (Gokmen, 2021). However, despite these refinements, rank-update methods still struggle to converge reliably when scaling to larger models under limited conductance states. Some in-memory analog training approaches employ hybrid 3T1C–PCM synapses, which rely on closed-loop tuning (CLT) with about 20 write–verify retries to transfer gradients from the linear 3T1C capacitor to the non-ideal PCM device (Ambrogio et al., 2018; Cristiano et al., 2018). While precise, CLT introduces substantial latency, energy, and control overhead, and the bulky 3T1C CMOS cells also limit array density compared to compact memristive devices, which has motivated a shift toward more compact, modestly linear NVMs. Moreover, the volatile nature of CMOS capacitors means that the effective number of valid states is ultimately determined by the non-volatile PCM, preventing extension to arbitrary precision beyond the intrinsic limits of memristive devices. Our approach avoids the use of auxiliary CMOS components and iterative CLT, and instead employs fully asymmetric and limited-precision memristor-based synapses, which are trained in-memory through a simple *open-loop transfer* mechanism between tiles (i.e., without the need for verification following each pulse) without requiring reset operations after transfer compared with (Ambrogio et al., 2018; Cristiano et al., 2018), meanwhile can achieve high-precision training. Meanwhile, another class of in-memory training approaches known as *hybrid training paradigms* has emerged. The MP approach (Le Gallo et al., 2018), for example, computes gradients digitally and directly programs these high-precision gradients into low-precision analog weights, enabling high-accuracy training even with very few conductance states. Subsequent extensions (Wang et al., 2020; Huang et al., 2020) incorporate momentum to further suppress gradient noise. However, these methods often incur higher storage, memory access, and computational costs.

Low-precision computing. Existing works have mainly demonstrated how low-precision devices can support high-precision computing in scientific computing and DNN inference. In scientific computing, algorithmic techniques have been proposed to achieve high-precision arithmetic on binary devices Feinberg et al. (2018); Song et al. (2023), and residual programming strategies sequentially approximate the residuals of previously programmed tiles Song et al. (2024). The residual concept has also inspired precision-enhancing strategies in DNN inference, where multiple cells are used to encode high-precision weights Le Gallo et al. (2022); Pedretti et al. (2021); Boybat et al. (2018). Varying the significance of these devices has been shown to yield further accuracy improvements Mackin et al. (2022). Unlike these approaches that operate on static weights, our method tackles the more challenging setting of training, where weights dynamically evolve under asymmetric and low-precision updates across multiple coupled tiles, and must still converge jointly toward the optimal solution. Alternative precision enhancement strategies include incorporating hardware specific noises into the training process to improve inference accuracy Klachko et al. (2019); He et al. (2019). Other approaches include logarithmic weight-to-conductance mappings that bias encoding toward more stable device states Vasilopoulos et al. (2023), and inference Zhang et al. (2017) through weighted majority

Device Name	**# States**	**Mature**
Capacitor (Li et al., 2018)	400	✓
ECRAM (Tang et al., 2018)	1000	✗
ECRAM (MO) (Kim et al., 2019)	7100	✗
PCM (Nandakumar et al., 2020)	200	✓
RERAM (OM) (Gong et al., 2022)	21	✓
RERAM (HfO_2) (Gong et al., 2022)	4	✓
RERAM (AlO_x/HfO_2) (Woo et al., 2016)	40	✓
RERAM (PCMO) (Park et al., 2013)	50	✓
RERAM (HfO_2) (Jiang et al., 2016)	26	✓

Table 5: Comparison of representative analog memory devices used for DNN training. Here, *Mature* indicates whether the device technology has been demonstrated with stable fabrication (Joshi et al., 2020).

voting after offline learning of binary weights and column-specific scaling factors.

Memristor devices. To provide an intuitive overview of conductance granularity in current analog memory devices, we survey recent reports on ReRAM (Stathopoulos et al., 2017; Chang et al., 2011), PCM (Burr et al., 2014, 2016), MRAM (Apalkov et al., 2013; Rzeszut et al., 2022), and ECRAM (Fuller et al., 2019), and summarize their reported numbers of accessible conductance states in Table 5. In this work, we focus on the algorithmic implications of limited conductance granularity under bidirectional updates, a regime that is practically relevant to several analog training device classes and is frequently discussed in the context of ReRAM-based training hardware. Based on the device reports summarized in Table 5, experimentally demonstrated analog memory devices often expose well below ideal high-precision conductance control, with many practical demonstrations often clustering around a few-bit regime. This motivates studying training methods that remain effective without relying on unrealistically fine conductance resolution. As illustrated in Table 4, where we conducted experiments on ResNet-34 with a larger portion of the model converted to analog and using 80 states, accuracy collapses for TT-v1 and remains far below our method even when TT-v2 is given the same device with only three tiles. These results suggest that a residual multi-tile mechanism can play an important role in scaling analog training to larger models under low-state device constraints.

Contribution relative to prior works. Our work extends Wu et al. (2024, 2025) by introducing a new setting that accounts for limited conductance state non-idealities and by addressing it through a new algorithm called multi-timescale residual learning. Both prior studies focus on asymmetric non-idealities: they model the analog update dynamics and establish convergence analyzes for Analog SGD and the Tiki-Taka algorithm. Specifically, Wu et al. (2024) considers only an asymmetric linear device, while Wu et al. (2025) extends the analysis to general asymmetric devices to demonstrate the scalability of the approach. Building on their foundation, we incorporate another widespread device non-ideality, limited conductance states, by modeling a quantization noise term that introduces a new non-vanishing error component in Analog SGD. To mitigate this error, we generalize the two-tile residual learning scheme in Wu et al. (2025) to a multi-tile regime, where each tile approximates the residual left by lower-resolution tiles, leading to an exponentially reduced error floor as the number of tiles increases. The key challenge is that this residual keeps drifting as lower-resolution tiles are updated, which we resolve by a multi-timescale learning strategy that freezes the lower-resolution tiles while the current tile runs a sufficiently long inner loop to track its quasi-stationary residual.

B MAPPING COEFFICIENT SETTING

In AIMC, each *logical weight* W is physically represented by mapping the difference between two *physical conductance* values: a main conductance C_{main} and a reference conductance C_{ref}. Specifically, the mapping takes the form

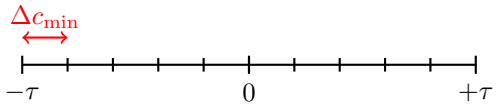
$$W = \kappa C = \kappa(C_{\text{main}} - C_{\text{ref}}) \tag{8}$$

where κ is a fixed scaling constant that determines the logical weight range based on the physical conductance range of the device.

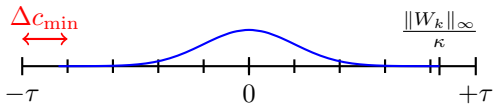
This representation scheme allows the hardware to represent both positive and negative weights using non-negative conductance values, which are physically realizable. Before proceeding with the analysis, we clarify a slight abuse of notation used in the main text. In our notation, we do not explicitly distinguish between the physical conductance C of the memristive crossbar array and the corresponding logical weight W, which are related by a fixed mapping constant. However, it is important to note that the device-level response functions $q_{\pm}(\cdot)$ as well as symmetric and asymmetric components $F(\cdot)$ and $G(\cdot)$ are defined over the conductance domain. In the following theoretical analysis, we will reintroduce this mapping explicitly when necessary to ensure mathematical correctness. In fact, the conductance-update rule is derived using the same approach as the update rule presented in (2) of the main text:

$$C_{t+1} = C_t + \Delta C_t \odot F(C_t) - |\Delta C_t| \odot G(C_t) + \frac{\zeta_t}{\kappa} \tag{9}$$

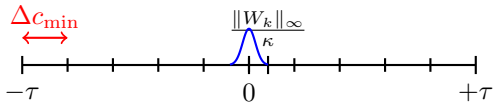
where ζ_t is a stochastic quantization noise term introduced by the finite weight resolution $\Delta w_{\min}$, with $\mathbb{E}[\zeta_t] = 0, \text{Var}[\zeta_t] = \Theta(\alpha \cdot \Delta w_{\min})$. Here α is the learning rate. See details in Lemma 1. We rewrite (2) in its exact form



(a) A 10-state memristive device with range $[-\tau, +\tau]$ and increment $\Delta c_{\min}$.



(b) With $\kappa = \mathcal{O}(1)$, weights span most of the conductance range.



(c) With $\kappa = \mathcal{O}(\gamma^{-N/4})$, weights lie well inside the range.

Figure 5: Comparison of dynamic ranges and weight distributions under different κ.

by multiply κ on both sides of (9):

$$W_{t+1} = W_t + \Delta W_t \odot F\left(\frac{W_t}{\kappa}\right) - |\Delta W_t| \odot G\left(\frac{W_t}{\kappa}\right) + \zeta_t. \tag{10}$$

All theoretical guarantees in Section E.2 and F are proved using (10); the shorthand (2) is retained elsewhere to keep the notation compact. Furthermore, although the logical weight $W \in \mathbb{R}^{D\times D}$ in the main text, the proof part focuses on a vector-valued $W \in \mathbb{R}^D$ while retaining the same uppercase notation to distinguish it from scalar elements w. This simplification is justified because, in analog updates—whether during gradient updates or transfer updates—each column of the weight matrix behaves identically. Specifically, during gradient updates, all elements in a column are updated in parallel, and during transfer updates, updates are applied column-wise from one tile to another. Therefore, the vector setting captures the essential behavior without loss of generality. Table 6 summarizes the notations that appear in both the main text and the proofs, especially those where conductance and weight representations may be easily confused.

In section F, the mapping constant κ is set as $\mathcal{O}(\gamma^{-\frac{N}{4}})$ to make Theorem 3 hold. This setting is feasible because increasing the mapping coefficient κ narrows the physical conductance range used to represent the same logical weight values. Figure 5 illustrates that when the logical weights are concentrated well within the conductance boundaries, the impact of non-ideal device behaviors such as saturation, nonlinearities, or non-monotonicity is significantly reduced. This makes the training process more robust.

C NOTATIONS

In this section, we define a series of notations that will be used in the analysis.

Pseodo-inverse of diagonal matrix or vector. For a given diagonal matrix $U \in \mathbb{R}^{D\times D}$ with its i-th diagonal element $[U]_i$, we define the pseudo-inverse of a diagonal matrix U as $U^\dagger$, which is also a diagonal matrix with its i-th diagonal element

$$[U^\dagger]_i := \begin{cases} 1/[U]_i, & [U]_i \neq 0, \\ 0, & [U]_i = 0. \end{cases} \tag{11}$$

By definition, the pseudo-inverse satisfies $UU^\dagger V = U^\dagger U V$ for any diagonal matrix $U \in \mathbb{R}^D$ and any matrix $V \in \mathbb{R}^D$. With a slight abuse of notation, we also define the pseudo-inverse of a vector $W \in \mathbb{R}^D$ as $W^\dagger := \mathrm{diag}(W)^\dagger$.

Weighted norm. For a given weight vector $S \in \mathbb{R}^D_+$, the weighted norm $\|\cdot\|_S$ of vector $W \in \mathbb{R}^D$ is defined by

$$\|W\|_S := \sqrt{\sum_{d=1}^{D} [S]_d [W]_d^2} = W^\top \mathrm{diag}(S) W \tag{12}$$

Symbol	Meaning
Conductance domain	
$[-\tau, \tau]$	Physical conductance range
C_t	Matrix of physical conductance at step t
c_t	Scalar conductance value of a single element at step t
$\Delta c_{\min}$	Minimum conductance increment from one pulse at $c = 0$
$F(c), G(c)$	Symmetric and asymmetric components of conductance
$q_+(c), q_-(c)$	Upward and downward pulse response factors at conductance c
$U(c, \Delta c)$	Approximate analog update over conductance
$U_p^{\mathrm{BL}}(c, s)$	Pulse-based update using bit-length pulses in conductance domain
Weight domain	
$[\tau_{\min}, \tau_{\max}]$	Logical weight range used in training
W_t	Matrix of logical weights at step t
w_t	Scalar logical weight of a single element at step t
$\|W_t\|_\infty$	Maximum absolute value among all elements in W_t
$W_{\max}$	Upper bound of $\|W_t\|_\infty$ for all t, lies in $[\|W_t\|_\infty, \tau_{\max})$
$\Delta w_{\min}$	Minimum weight increment from one pulse at $w = 0$
$F(w), G(w)$	Symmetric and asymmetric components evaluated as $F(w/\kappa), G(w/\kappa)$
$q_+(w), q_-(w)$	Pulse response factors evaluated as $q_\pm(w) := q_\pm(w/\kappa)$
$U(w, \Delta w)$	Approximate analog update over logical weights
$U_p^{\mathrm{BL}}(w, s)$	Pulse-based update using bit length in weight domain
Shared / Mapping	
κ	Mapping constant: $W = \kappa C$

Table 6: Notations in the conductance and weight domains

where $\mathrm{diag}(S) \in \mathbb{R}_+^{D\times D}$ rearranges the vector S into a diagonal matrix.

Lemma 4. *$\|W\|_S$ has the following properties: (1) $\|W\|_S = \|W \odot \sqrt{S}\|$; (2) $\|W\|_S \le \|W\|\sqrt{\|S\|_\infty}$; (3) $\|W\|_S \ge \|W\|\sqrt{\min\{[S]_i : i \in \mathcal{I}\}}$.*

D USEFUL LEMMAS

D.1 Lemma 1: Pulse update error

Lemma 1 (Statistical properties of pulse update noise)**.** *Under the stochastic pulse update in (Gokmen and Vlasov, 2016), the random variable ζ_{ij} has the following properties:*

$$\mathbb{E}[\zeta_{ij}] = 0, \quad \textit{and} \quad \mathrm{Var}[\zeta_{ij}] = \Theta(\alpha \cdot \Delta w_{\min}).$$

Proof of Lemma 1. Each weight update Δw_{ij} is the sum of BL independent Bernoulli trials , with BL large enough to satisfy $\frac{\alpha|x_i\delta_j|}{BL\cdot\Delta w_{\min}} \le 1$:

$$\Delta w_{ij} = \sum_{t=1}^{BL} Z_t \tag{13}$$

where:

$$Z_t = \begin{cases} \Delta w_{\min} \cdot \mathrm{sign}(x_i\delta_j), & \text{with probability } p := \frac{|\alpha x_i \delta_j|}{BL\cdot\Delta w_{\min}}, \\ 0, & \text{with probability } 1-p. \end{cases} \tag{14}$$

Then:

$$\mathbb{E}[Z_t] = \Delta w_{\min} \cdot \operatorname{sign}(x_i \delta_j) \cdot p = \frac{\alpha x_i \delta_j}{BL}, \quad \mathbb{E}[Z_t^2] = \Delta w_{\min}^2 \cdot p. \tag{15}$$

The equality holds for $\operatorname{sign}(x_i \delta_j) \cdot |x_i \delta_j| = x_i \delta_j$. So the variance of a single trial is:

$$\operatorname{Var}[Z_t] = \mathbb{E}[Z_t^2] - \mathbb{E}[Z_t]^2 = \Delta w_{\min}^2 p(1-p). \tag{16}$$

Then summing over BL trials:

$$\mathbb{E}[\Delta w_{ij}] = BL \cdot \mathbb{E}[Z_t] = \alpha x_i \delta_j, \quad \operatorname{Var}[\Delta w_{ij}] = BL \cdot \Delta w_{\min}^2 \cdot p \cdot (1-p). \tag{17}$$

Thus,

$$\mathbb{E}[\zeta_{ij}] = \mathbb{E}[\Delta w_{ij}] - \alpha x_i \delta_j = 0. \tag{18}$$

Moreover, substituting $p = \dfrac{|\alpha x_i \delta_j|}{BL \cdot \Delta w_{\min}}$ into $\operatorname{Var}[\Delta w_{ij}]$, we get:

$$\operatorname{Var}[\Delta w_{ij}] = BL \cdot \Delta w_{\min}^2 \cdot \frac{|\alpha x_i \delta_j|}{BL \cdot \Delta w_{\min}} \cdot \left(1 - \frac{|\alpha x_i \delta_j|}{BL \cdot \Delta w_{\min}}\right). \tag{19}$$

Thus,

$$\operatorname{Var}[\zeta_{ij}] = \operatorname{Var}[\Delta w_{ij}] = \alpha |x_i \delta_j| \cdot \Delta w_{\min} \cdot \left(1 - \frac{\alpha |x_i \delta_j|}{BL \cdot \Delta w_{\min}}\right) = \Theta(\alpha \cdot \Delta w_{\min}). \tag{20}$$

□

D.2 Lemma 5: Lipschitz continuity of analog update

Lemma 5. *Under Assumption 4, the analog increment defined in* (2) *is Lipschitz continuous with respect to* ΔW *in terms of any weighted norm* $\|\cdot\|_S$*, i.e., for any* $W, \Delta W, \Delta W' \in \mathbb{R}^D$ *and* $S \in \mathbb{R}_+^D$*, it holds*

$$\begin{aligned} &\|\Delta W \odot F(W) - |\Delta W| \odot G(W) - (\Delta W' \odot F(W) - |\Delta W'| \odot G(W))\|_S \\ &\leq F_{\max} \|\Delta W - \Delta W'\|_S. \end{aligned} \tag{21}$$

The proof of Lemma 5 can be found in (Wu et al., 2025, Section C).

Lemma 6. *Under Assumption 4, the following statements about the response factors are valid; (1) the symmetric part* $F(\cdot)$ *is upper bounded by a constant* $F_{\max} > 0$*, i.e.* $F(w) \leq F_{\max}, \forall w \in \mathbb{R}$*; (2) The following inequality holds*

$$-F(w) \leq G(w) \leq F(w) \tag{22}$$

where $G(w) = -F(w)$ *and* $G(w) = F(w)$ *hold only when* $w = \tau_{\min}$ *and* $w = \tau_{\max}$*, respectively.*

D.3 Lemma 7: Element-wise product error

Lemma 7. *Let* U, V, Q *be vectors indexed by* $\mathcal{I}$*. Then the following inequality holds*

$$\langle U, V \odot Q \rangle \geq C_+ \langle U, V \rangle - C_- \langle |U|, |V| \rangle \tag{23}$$

where the constant C_+ *and* C_- *are defined by*

$$C_+ := \frac{1}{2} \left(\max_{i \in \mathcal{I}} \{[Q]_i\} + \min_{i \in \mathcal{I}} \{[Q]_i\} \right), \tag{24}$$

$$C_- := \frac{1}{2} \left(\max_{i \in \mathcal{I}} \{[Q]_i\} - \min_{i \in \mathcal{I}} \{[Q]_i\} \right). \tag{25}$$

The proof of Lemma 7 can be found in (Wu et al., 2025, Section C).

E PROOF OF ANALOG STOCHASTIC GRADIENT DESCENT CONVERGENCE

E.1 Convergence of Analog SGD

In this section, we derive the convergence guarantee of *Analog SGD* under the hardware-constrained update rule in (10). In this section, we derive the convergence guarantee of *Analog SGD* under the hardware-constrained update rule in (10).

Theorem 4 (Convergence of Analog SGD, long version of Theorem 1)**.** *Suppose Assumptions 1, 3, 4 hold. Let* $\alpha = \mathcal{O}(\sqrt{\frac{2(f(W_0)-f^*)}{\sigma^2 T}})$*. Then it holds that:*

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|W^* - W_t\|^2] \leq \mathcal{O}\left(R_T\sqrt{\frac{2(f(W_0)-f^*)\sigma^2}{T}}\right) + 4\sigma^2 S_T + R_T \Delta w_{\min}$$

where S_T *and* R_T *denote the amplification factor given by :*

$$S_T := \frac{1}{T}\sum_{t=0}^{T-1}\frac{\|W_t\|_\infty^2/\tau_{\max}^2}{1-\|W_t\|_\infty^2/\tau_{\max}^2}, \qquad R_T := \frac{1}{T}\sum_{t=0}^{T-1}\frac{2L}{1-\|W_t\|_\infty^2/\tau_{\max}^2}. \tag{26}$$

Proof of Theorem 4. The L-smooth assumption (Assumption 2) implies that:

$$\mathbb{E}_{\xi_t,\zeta_t}[f(W_{t+1})] \leq f(W_t) + \underbrace{\mathbb{E}_{\xi_t,\zeta_t}[\langle \nabla f(W_t), W_{t+1}-W_t\rangle]}_{(a)} + \underbrace{\frac{L}{2}\mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1}-W_t\|^2]}_{(b)}. \tag{27}$$

The term (a) in (27) can be bounded by Assumption 1 that $\mathbb{E}_{\xi_t,\zeta_t}[\nabla f(W_t;\xi_t)] = \nabla f(W_t)$ and $2\langle U, V\rangle = \|U+V\|^2 - \|U\|^2 - \|V\|^2$:

$$\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}[\langle \nabla f(W_t), W_{t+1}-W_t\rangle] \qquad (28)\\
&= \alpha_t \mathbb{E}_{\xi_t,\zeta_t}\left[\left\langle \nabla f(W_t)\odot\sqrt{F(\frac{W_t}{\kappa})}, \frac{W_{t+1}-W_t}{\alpha_t\sqrt{F(\frac{W_t}{\kappa})}} + \sqrt{F(\frac{W_t}{\kappa})}\odot(\nabla f(W_t;\xi_t)-\nabla f(W_t)) - \frac{\zeta_t}{\alpha_t\sqrt{F(\frac{W_t}{\kappa})}}\right\rangle\right]\\
&= -\frac{\alpha_t}{2}\left\|\sqrt{F(\frac{W_t}{\kappa})}\odot\nabla f(W_t)\right\|^2\\
&\quad - \frac{1}{2\alpha_t}\mathbb{E}_{\xi_t,\zeta_t}\left[\left\|\frac{W_{t+1}-W_t}{\sqrt{F(\frac{W_t}{\kappa})}} + \alpha_t\sqrt{F(\frac{W_t}{\kappa})}\odot(\nabla f(W_t;\xi_t)-\nabla f(W_t)) - \frac{\zeta_t}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|^2\right]\\
&\quad + \frac{1}{2\alpha_t}\mathbb{E}_{\xi_t}\left[\left\|\frac{W_{t+1}-W_t}{\sqrt{F(\frac{W_t}{\kappa})}} + \alpha_t\sqrt{F(\frac{W_t}{\kappa})}\odot\nabla f(W_t;\xi_t) - \frac{\zeta_t}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|^2\right].
\end{aligned}$$

The second term in (28) can be bounded by

$$\begin{aligned}
&\frac{1}{2\alpha_t}\mathbb{E}_{\xi_t,\zeta_t}\left[\left\|\frac{W_{t+1}-W_t}{\sqrt{F(\frac{W_t}{\kappa})}} + \alpha_t\sqrt{F(\frac{W_t}{\kappa})}\odot(\nabla f(W_t;\xi_t)-\nabla f(W_t)) - \frac{\zeta_t}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|^2\right] \qquad (29)\\
&= \frac{1}{2\alpha}\mathbb{E}_{\xi_t,\zeta_t}\left[\left\|\frac{W_{t+1}-W_t+\alpha(\nabla f(W_t;\xi_t)-\nabla f(W_t))\odot F(\frac{W_t}{\kappa}) - \zeta_t}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|^2\right]
\end{aligned}$$

$$\geq \frac{1}{2\alpha F_{\max}} \mathbb{E}_{\xi_t,\zeta_t}\left[\left\| W_{t+1} - W_t + \alpha(\nabla f(W_t;\xi_t) - \nabla f(W_t)) \odot F(\frac{W_t}{\kappa}) - \zeta_t \right\|^2\right].$$

The last inequality holds by defining a constant $F_{\max}$ such that $\|F(W)\|_\infty \leq F_{\max}$. The third term in (28) can be bounded by variance decomposition and bounded variance assumption (Assumption 1)

$$\begin{aligned}
&\frac{1}{2\alpha_t}\mathbb{E}_{\xi_t}\left[\left\| \frac{W_{t+1}-W_t}{\sqrt{F(\frac{W_t}{\kappa})}} + \alpha_t\sqrt{F(\frac{W_t}{\kappa})} \odot \nabla f(W_t;\xi_t) - \frac{\zeta_t}{\sqrt{F(\frac{W_t}{\kappa})}} \right\|^2\right] \\
&= \frac{\alpha_t}{2}\mathbb{E}_{\xi_t}\left[\left\| |\nabla f(W_t;\xi_t)| \odot \frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}} \right\|^2\right] \\
&\leq \frac{\alpha_t}{2}\left\| |\nabla f(W_t)| \odot \frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}} \right\|^2 + \frac{\alpha_t\sigma^2}{2}\left\| \frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}} \right\|_\infty^2.
\end{aligned} \tag{30}$$

Define the saturation vector the saturation vector $H(W_t) \in \mathbb{R}^D$ as:

$$\begin{aligned}
H(W_t) &:= F\Big(\frac{W_t}{\kappa}\Big)^{\odot 2} - G\Big(\frac{W_t}{\kappa}\Big)^{\odot 2} \\
&= \Big(F\Big(\frac{W_t}{\kappa}\Big) + G\Big(\frac{W_t}{\kappa}\Big)\Big) \odot \Big(F\Big(\frac{W_t}{\kappa}\Big) - G\Big(\frac{W_t}{\kappa}\Big)\Big) \\
&= q_+\Big(\frac{W_t}{\kappa}\Big) \odot q_-\Big(\frac{W_t}{\kappa}\Big).
\end{aligned} \tag{31}$$

Note that the first term in the right-hand side (RHS) of (27) and the second term in the RHS of (30) can be bounded by

$$\begin{aligned}
&-\frac{\alpha_t}{2}\left\| \nabla f(W_t) \odot \sqrt{F(\frac{W_t}{\kappa})} \right\|^2 + \frac{\alpha_t}{2}\left\| |\nabla f(\frac{W_t}{\kappa})| \odot \frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}} \right\|^2 \\
&= -\frac{\alpha_t}{2}\sum_{d\in[D]}\left([\nabla f(W_t)]_d^2\left([F(\frac{W_t}{\kappa})]_d - \frac{[G(\frac{W_t}{\kappa})]_d^2}{[F(\frac{W_t}{\kappa})]_d}\right)\right) \\
&= -\frac{\alpha_t}{2}\sum_{d\in[D]}\left([\nabla f(W_t)]_d^2\left(\frac{[F(\frac{W_t}{\kappa})]_d^2 - [G(\frac{W_t}{\kappa})]_d^2}{[F(\frac{W_t}{\kappa})]_d}\right)\right) \\
&\leq -\frac{\alpha_t}{2F_{\max}}\sum_{d\in[D]}\left([\nabla f(W_t)]_d^2\left([F(\frac{W_t}{\kappa})]_d^2 - [G(\frac{W_t}{\kappa})]_d^2\right)\right) \\
&= -\frac{\alpha_t}{2F_{\max}}\|\nabla f(W_t)\|_{H(W_t)}^2 \leq 0.
\end{aligned} \tag{32}$$

Plugging (29) to (32) into (28), we bound the term (a) by

$$\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}[\langle \nabla f(W_t), W_{t+1} - W_t\rangle] \\
&= -\frac{\alpha_t}{2F_{\max}}\|\nabla f(W_t)\|_{H(W_t)}^2 + \frac{\alpha_t\sigma^2}{2}\left\| \frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}} \right\|_\infty^2 \\
&\quad -\frac{1}{2\alpha F_{\max}}\mathbb{E}_{\xi_t,\zeta_t}\left[\left\| W_{t+1} - W_t + \alpha(\nabla f(W_t;\xi_t) - \nabla f(W_t)) \odot F(\frac{W_t}{\kappa}) - \zeta_t \right\|^2\right].
\end{aligned} \tag{33}$$

The term (b) in (27) can be bounded by $\mathbb{E}_{\xi_t}[\|\nabla f(W_t;\xi_t)-\nabla f(W_t)\|^2]\le\sigma^2$:

$$
\begin{aligned}
&\frac{L}{2}\mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1}-W_t\|^2] \\
&\le L\mathbb{E}_{\xi_t,\zeta_t}\left[\left\|W_{t+1}-W_t+\alpha_t(\nabla f(W_t;\xi_t)-\nabla f(W_t))\odot F(\frac{W_t}{\kappa})-\zeta_t\right\|^2\right] \\
&\quad + L\mathbb{E}_{\xi_t}\left[\left\|\alpha_t(\nabla f(W_t;\xi_t)-\nabla f(W_t))\odot F(\frac{W_t}{\kappa})+\zeta_t\right\|^2\right] \\
&\le L\mathbb{E}_{\xi_t,\zeta_t}\left[\left\|W_{t+1}-W_t+\alpha_t(\nabla f(W_t;\xi_t)-\nabla f(W_t))\odot F(\frac{W_t}{\kappa})-\zeta_t\right\|^2\right] \\
&\quad + 2\alpha_t^2F_{\max}^2L\sigma^2+2LF_{\max}^2\cdot\Theta(\alpha_t\Delta w_{\min}).
\end{aligned}
\tag{34}
$$

Substituting (33) and (34) back into (27), we have

$$
\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}[f(W_{t+1})] \\
&\le f(W_t)-\frac{\alpha_t}{2F_{\max}}\|\nabla f(W_t)\|_{H(W_t)}^2+2\alpha_t^2LF_{\max}^2\sigma^2+\frac{\alpha_t\sigma^2}{2}\left\|\frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|_\infty^2+2LF_{\max}^2\cdot\Theta(\alpha_t\Delta w_{\min}) \\
&-\frac{1}{F_{\max}}\left(\frac{1}{2\alpha_t}-LF_{\max}\right)\mathbb{E}_{\xi_t,\zeta_t}\left[\left\|W_{t+1}-W_t+\alpha(\nabla f(W_t;\xi_t)-\nabla f(W_t))\odot F(\frac{W_t}{\kappa})-\zeta_t\right\|^2\right] \\
&\le f(W_t)-\frac{\alpha_t}{2F_{\max}}\|\nabla f(W_t)\|_{H(W_t)}^2+2\alpha_t^2LF_{\max}^2\sigma^2+\frac{\alpha_t\sigma^2}{2}\left\|\frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|_\infty^2+2LF_{\max}^2\cdot\Theta(\alpha_t\Delta w_{\min}).
\end{aligned}
\tag{35}
$$

The last inequality holds when $\alpha_t\le\frac{1}{2LF_{\max}}$. Taking average over t, we get:

$$
\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|\nabla f(W_t)\|_{H(W_t)}^2] \\
&\le\frac{2F_{\max}(f(W_0)-f(W_T))}{\alpha_tT}+4LF_{\max}^3\alpha_t\sigma^2+\sigma^2F_{\max}\frac{1}{T}\sum_{t=0}^{T-1}\left\|\frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|_\infty^2+4LF_{\max}^3\Theta(\Delta w_{\min}) \\
&\le\mathcal{O}\left(F_{\max}^2\sqrt{\frac{8L(f(W_0)-f^*)\sigma^2}{T}}\right)+S_TF_{\max}\sigma^2+4LF_{\max}^3\Theta(\Delta w_{\min}).
\end{aligned}
\tag{36}
$$

The second inequality holds by choosing $\alpha=\mathcal{O}(\sqrt{\frac{f(W_0)-f^*}{2LF_{\max}^2\sigma^2T}})$. S_T denotes the amplification factors given by:

$$
S_T:=\frac{1}{T}\sum_{t=0}^{T-1}\left\|\frac{G(\frac{W_t}{\kappa})}{\sqrt{F(\frac{W_t}{\kappa})}}\right\|_\infty^2. \tag{37}
$$

Adopting the bounded saturation (Wu et al., 2025, Assumption 3), we assume there exists a non-zero constant $H_{\min}>0$ such that $\min\{H(W_t)\}\ge H_{\min}$ for all t. Since $\mu(W-W^*)\le\nabla f(W)$, we multiply both sides by μ and normalize by the constant $H_{\min}$. This yields the upper bound for Analog SGD on general response factors:

$$
\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|W^*-W_t\|^2]\le\mathcal{O}\left(\frac{F_{\max}^2}{H_{\min}}\sqrt{\frac{8L(f(W_0)-f^*)\sigma^2}{T}}\right)+\frac{S_TF_{\max}}{H_{\min}}\sigma^2+4L\frac{F_{\max}^3}{H_{\min}}\Theta(\Delta w_{\min}). \tag{38}
$$

For concrete illustration, we now analyze the asymmetric linear device (ALD) model, in which the pulse response factors are defined as $q_+(c) = 1 - \frac{c-c^\diamond}{\tau}$ and $q_-(c) = 1 + \frac{c-c^\diamond}{\tau}$ (Rasch et al., 2023). Here, $c^\diamond$ denotes the symmetric conductance point, at which the upward and downward pulse responses are equal, i.e., $q_+(c^\diamond) = q_-(c^\diamond)$; in other words, a positive and a negative pulse induce the same magnitude of conductance change at this point. By Assumption 4, we set $c^\diamond = 0$ due to zero-shifting of the conductance domain. Under these settings, the symmetric and asymmetric components simplify to $F(c) = 1$ and $G(c) = \frac{|c|}{\tau}$. Correspondingly, in the weight domain, we have $F(\frac{w}{\kappa}) = 1$ and $G(\frac{w}{\kappa}) = \frac{|w/\kappa|}{\tau} = \frac{|w|}{\tau_{\max}}$, where $\tau_{\max} = \kappa\tau$. Substituting the matrix-form expressions $F(\frac{W}{\kappa}) = 1$ and $G(\frac{W}{\kappa}) = \frac{|W|}{\tau_{\max}}$ into (10), the Analog SGD recursion on ALD reduces to:

$$W_{t+1} = W_t - \alpha_t \nabla f(W_t; \xi_t) - \frac{\alpha_t}{\tau_{\max}} |\nabla f(W_t; \xi_t)| \odot W_t + \zeta_t, \tag{39}$$

We can naturally get that $F_{\max} = 1$, $H_{\min,t} = 1 - \frac{\|W_t\|_\infty^2}{\tau_{\max}^2}$. Rearranging (35) as

$$\begin{aligned} &\mathbb{E}_{\xi_t,\zeta_t}[f(W_{t+1})] \\ &\leq f(W_t) - \frac{\alpha_t}{2(1 - \frac{\|W_t\|_\infty^2}{\tau_{\max}^2})} \|\nabla f(W_t)\|^2 + 2\alpha_t^2 L\sigma^2 + \frac{\alpha_t \sigma^2 \|W_t\|_\infty^2}{2\tau_{\max}^2} + 2L \cdot \Theta(\alpha_t \Delta w_{\min}). \end{aligned} \tag{40}$$

Divide both sides of (40) by $1 - \|W_t\|_\infty^2/\tau_{\max}^2 > 0$ and average over t:

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(W_t)\|^2] &\leq \frac{2R_T(f(W_0) - \mathbb{E}[f(W_T)])}{\alpha_t T} + R_T \alpha_t \sigma^2 + 4\sigma^2 S_T + R_T \Theta(\Delta w_{\min}) \\ &\leq \frac{2R_T(f(W_0) - f^*)}{\alpha_t T} + R_T \alpha_t \sigma^2 + 4\sigma^2 S_T + R_T \Theta(\Delta w_{\min}). \end{aligned} \tag{41}$$

Here, S_T and R_T denote the amplification factors given by :

$$S_T := \frac{1}{T}\sum_{t=0}^{T-1} \frac{\|W_t\|_\infty^2/\tau_{\max}^2}{1 - \|W_t\|_\infty^2/\tau_{\max}^2}, \qquad R_T := \frac{1}{T}\sum_{t=0}^{T-1} \frac{2L}{1 - \|W_t\|_\infty^2/\tau_{\max}^2}. \tag{42}$$

Choosing $\alpha = \mathcal{O}(\sqrt{\frac{2(f(W_0)-f^*)}{\sigma^2 T}})$, when $T \to \infty$, it satisfies that $\alpha \leq \frac{1}{2L}$ and (41) becomes:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f(W_t)\|^2] \leq \mathcal{O}\left(R_T \sqrt{\frac{2(f(W_0) - f^*)\sigma^2}{T}}\right) + 4\sigma^2 S_T + R_T \Delta w_{\min}. \tag{43}$$

Since $\mu(W - W^*) \leq \nabla f(W)$, multiplying μ on both sides, we get:

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|W^* - W_t\|^2] \leq \mathcal{O}\left(R_T \sqrt{\frac{2(f(W_0) - f^*)\sigma^2}{T}}\right) + 4\sigma^2 S_T + R_T \Delta w_{\min}. \tag{44}$$

which completes the proof. □

E.2 Lower bound of asymptotic error for Analog SGD

Under the convergence of Analog SGD with hardware-constrained update rule in Theorem 4, we derive a lower bound on the asymptotic error floor that arises when training with a single analog tile on non-ideal AIMC hardware.

Theorem 5 (Asymptotic error bound under quantization, long version of Theorem 2)**.** *Suppose Assumptions 1-4 hold, $\alpha = \frac{1}{2L}$, there exists an instance where Analog SGD generates a sequence $\{W_t\}_{t=0}^{T-1}$ such that:*

$$\frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}[\|W^* - W_t\|^2] \geq \Omega(\sigma^2 S_T + R_T \Delta w_{\min})$$

This section provides the lower bound of Analog SGD on noisy asymmetric linear devices with limited conductance states under Assumptions 1–4. The proof is completed based on the following example from (Wu et al., 2024, Section G.2).

(Example) Consider an example where all the coordinates are identical, i.e., $W_t = w_t\mathbf{1}$ for some $w_t \in \mathbb{R}$ where $\mathbf{1} \in \mathbb{R}^D$ is the all-one vector. Define $W^* = w^*\mathbf{1}$ where $w^* \in \mathbb{R}$ is a constant scalar and a quadratic function $f(W) := \frac{L}{2}\|W - W^*\|^2$ whose minimum is W^*. Initialize the weight on $W_0 = W^*$. Furthermore, consider the sample noise ξ_t defined as $\xi_t = \varepsilon_t\mathbf{1}$, where random variable $\varepsilon_t \in \mathbb{R}$ is sampled by:

$$\varepsilon_t = \begin{cases} \varepsilon_t^+ := \frac{\sigma}{\sqrt{D}}\sqrt{\frac{1-p_t}{p_t}}, & \text{w.p. } p_t, \\ \varepsilon_t^- := -\frac{\sigma}{\sqrt{D}}\sqrt{\frac{p_t}{1-p_t}}, & \text{w.p. } 1-p_t, \end{cases} \qquad \text{with } p_t = \frac{1}{2}\left(1 - \frac{w_t}{\tau_{\max}}\right). \tag{45}$$

As a reminder, it is always valid that $|w_t| = \|W_t\|_\infty \le \tau_{\max}$ (see (Wu et al., 2024, Theorem 5)) and $0 \le p_t \le 1$. Therefore, the noise distribution is well-defined. Furthermore, without loss of generality, we assume $|w^*| \le \frac{\tau_{\max}}{4}$ and $\sigma \le \frac{\tau_{\max}L\sqrt{D}}{4\sqrt{3}}$. We define the objective $f(w;\varepsilon_t) := \frac{L}{2}\left(w - w^* + \frac{\varepsilon_t}{L}\right)^2$, whose minimum is $w^*_{\xi_t} = w^* - \frac{\varepsilon_t}{L}$. The noise ε_t satisfies (Wu et al., 2024, Assumption 7).

Proof of Theorem 5. Consider the example constructed above. Before deriving the lower bound, we demonstrate Assumption 1–2 hold. It is obvious that $\nabla f(W) = L(W - W^*)$ satisfies Assumption 2. In addition, Assumption 1 could be verified by noticing (45) implies $\mathbb{E}_{\xi_t}[\xi_t] = 0$ and $\mathbb{E}_{\xi_t}[\|\xi_t\|^2] \le \sigma^2$. Assumption 3 is verified by (Wu et al., 2024, Lemma 2). We now proceed to derive the lower bound. Manipulating the recursion (39), we have the following result:

$$\begin{aligned} &\mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1} - W^*\|^2] \\ &= \mathbb{E}_{\xi_t,\zeta_t}[\|W_t - \alpha_t\nabla f(W_t;\xi_t) - \frac{\alpha_t}{\tau_{\max}}|\nabla f(W_t;\xi_t)| \odot W_t - W^* + \zeta_t\|^2] \\ &= \mathbb{E}_{\xi_t}[\|W_t - \alpha_t\nabla f(W_t;\xi_t) - \frac{\alpha_t}{\tau_{\max}}|\nabla f(W_t;\xi_t)| \odot W_t - W^*\|^2] + \Theta(\alpha_t\Delta w_{\min}) \\ &= \mathbb{E}_{\xi_t}[\|W_t - \alpha_t\nabla f(W_t;\xi_t) - W^*\|^2] + \mathbb{E}_{\xi_t}[\|\frac{\alpha_t}{\tau_{\max}}|\nabla f(W_t;\xi_t)| \odot W_t\|^2] \\ &\quad - 2\mathbb{E}_{\xi_t}[\langle W_t - \alpha_t\nabla f(W_t;\xi_t) - W^*, \frac{\alpha_t}{\tau_{\max}}|\nabla f(W_t;\xi_t)| \odot W_t\rangle] + \Theta(\alpha_t\Delta w_{\min}). \end{aligned} \tag{46}$$

The second equality holds for $\|U+V\|^2 = \|U\|^2 + \|V\|^2 + 2\langle U, V\rangle$ with $\mathbb{E}_{\zeta_t}[2\langle U, V\rangle] = \mathbb{E}_{\zeta_t}[\Theta(\zeta_t)] = 0$ here, and $\mathbb{E}_{\zeta_t}[\zeta_t^2] = \Theta(\alpha_t\Delta w_{\min})$. The first term on the right-hand side (RHS) of (46) can be bounded as:

$$\begin{aligned} &\mathbb{E}_{\xi_t}\left[\|W_t - \alpha_t\nabla f(W_t;\xi_t) - W^*\|^2\right] \\ &= \|W_t - W^*\|^2 - 2\alpha_t\mathbb{E}_{\xi_t}\left[\langle W_t - W^*, \nabla f(W_t;\xi_t)\rangle\right] + \alpha_t^2\mathbb{E}_{\xi_t}\left[\|\nabla f(W_t;\xi_t)\|^2\right] \\ &\ge (1 - 2\alpha_t L)\|W_t - W^*\|^2 + \alpha_t^2\mathbb{E}_{\xi_t}\left[\|\nabla f(W_t;\xi_t)\|^2\right]. \end{aligned} \tag{47}$$

Here the second equality uses the unbiasedness of the stochastic gradient, i.e., $\mathbb{E}_{\xi_t}[\nabla f(W_t;\xi_t)] = \nabla f(W_t)$, and the inequality follows from the Lipschitz condition $\langle W_t - W^*, \nabla f(W_t)\rangle \le L\|W_t - W^*\|^2$. The second term in the RHS of (46) can be bounded as:

$$\begin{aligned} &\mathbb{E}_{\xi_t}[\|\frac{\alpha_t}{\tau_{\max}}|\nabla f(W_t;\xi_t)| \odot W_t\|^2] \\ &= \frac{\alpha_t^2\|W_t\|_\infty^2}{\tau_{\max}^2}\mathbb{E}_{\xi_t}[\||\nabla f(W_t;\xi_t)|\|^2] = \frac{\alpha_t^2\|W_t\|_\infty^2}{\tau_{\max}^2}\mathbb{E}_{\xi_t}[\|\nabla f(W_t;\xi_t)\|^2] \end{aligned} \tag{48}$$

where the first equality uses $W_t = w_t\mathbf{1}$. From (45), we have $\mathbb{E}_{\xi_t}[\|\nabla f(W_t;\xi_t)\|^2] = \|\nabla f(W_t)\|^2 + \mathbb{E}_{\xi_t}[\|\xi_t\|^2] = L^2\|W_t - W^*\|^2 + \sigma^2$, substituting the equation into (47) and (48) yields:

$$\mathbb{E}_{\xi_t}[\|W_t - \alpha_t\nabla f(W_t;\xi_t) - W^*\|^2] + \mathbb{E}_{\xi_t}[\|\frac{\alpha_t}{\tau_{\max}}|\nabla f(W_t;\xi_t)| \odot W_t\|^2] \tag{49}$$

$$\geq \big(1 - 2\alpha_t L + \alpha_t^2 L^2 (1 + \frac{\|W_t\|_\infty^2}{\tau_{\max}^2})\big)\|W_t - W^*\|^2 + \alpha_t^2 \sigma^2 (1 + \frac{\|W_t\|_\infty^2}{\tau_{\max}^2}).$$

The third term in the RHS of (46) can be bounded as:

$$\begin{aligned}
&- 2\mathbb{E}_{\xi_t}[\langle W_t - \alpha_t \nabla f(W_t;\xi_t) - W^*, \frac{\alpha_t}{\tau_{\max}} |\nabla f(W_t;\xi_t)| \odot W_t \rangle] \\
&= -2\mathbb{E}_{\xi_t}[\langle W_t - W^*, \frac{\alpha_t W_t}{\tau_{\max}} \odot |\nabla f(W_t;\xi_t)| \rangle] + \frac{\alpha_t^2 W_t}{\tau_{\max}} \mathbb{E}_{\xi_t}[\langle \nabla f(W_t;\xi_t), |\nabla f(W_t;\xi_t)| \rangle] \\
&= -2\sum_{i=1}^{D}[\langle w_t - w^*, \frac{\alpha_t w_t}{\tau_{\max}} (p_t([\nabla f(W_t)]_i + \varepsilon_t^+) - (1-p_t)([\nabla f(W_t)]_i + \varepsilon_t^-)) \rangle] \\
&\quad + \frac{2\alpha_t^2 w_t}{\tau_{\max}} \sum_{i=1}^{D} \big[p_t([\nabla f(W_t)]_i + \varepsilon_t^+)^2 - (1-p_t)([\nabla f(W_t)]_i + \varepsilon_t^-)^2\big] \\
&\geq -\frac{2\alpha_t L \|W_t\|_\infty^2}{\tau_{\max}^2} \|W_t - W^*\|^2 + \frac{2\alpha_t^2 \|W_t\|_\infty^2}{\tau_{\max}^2} (-L^2 \|W_t - W^*\|^2 + \sigma^2) \\
&\quad - \frac{\alpha_t (w_t - w^*) w_t \sigma \sqrt{D}}{\tau_{\max}} \sqrt{(1 - \|W_t\|_\infty^2/\tau_{\max}^2)} + \frac{2\alpha_t^2 L (w_t - w^*) w_t \sigma \sqrt{D}}{\tau_{\max}} \sqrt{(1 - \|W_t\|_\infty^2/\tau_{\max}^2)} \\
&= -2(1 + \alpha_t L) \frac{\alpha_t L \|W_t\|_\infty^2}{\tau_{\max}^2} \|W_t - W^*\|^2 + \frac{2\alpha_t^2 \|W_t\|_\infty^2 \sigma^2}{\tau_{\max}^2}
\end{aligned} \tag{50}$$

where the first equation uses $W_t = w_t \mathbf{1}$, the second equality holds for (Wu et al., 2024, Lemma 4), which shows that $\nabla f(W_t) + \varepsilon_t^+ \geq 0$ and $\nabla f(W_t) + \varepsilon_t^- \leq 0$, the last equation holds by setting $\alpha_t = \frac{1}{2L}$, the third equation holds by simplifying the second term as:

$$\begin{aligned}
&\sum_{i=1}^{D}[\langle w_t - w^*, \frac{\alpha_t w_t}{\tau_{\max}} (p_t([\nabla f(W_t)]_i + \varepsilon_t^+) - (1-p_t)([\nabla f(W_t)]_i + \varepsilon_t^-)) \rangle] \\
&= \sum_{i=1}^{D}[\langle w_t - w^*, \frac{\alpha_t w_t}{\tau_{\max}} (p_t([\nabla f(W_t)]_i + \frac{\sigma}{\sqrt{D}} \sqrt{\frac{1-p_t}{p_t}}) - (1-p_t)([\nabla f(W_t)]_i - \frac{\sigma}{\sqrt{D}} \sqrt{\frac{p_t}{1-p_t}})) \rangle] \\
&= \sum_{i=1}^{D}[\langle w_t - w^*, \frac{\alpha_t w_t}{\tau_{\max}} (2p_t - 1)[\nabla f(W_t)]_i + \frac{\alpha_t w_t \sigma}{\tau_{\max}\sqrt{D}} \sqrt{(1-p_t)p_t} \rangle] \\
&= \langle W_t - W^*, -\frac{\alpha_t \|W_t\|_\infty^2}{\tau_{\max}} \nabla f(W_t) \rangle + \frac{\alpha_t (w_t - w^*) w_t \sigma \sqrt{D}}{2\tau_{\max}} \sqrt{(1 - \|W_t\|_\infty^2/\tau_{\max}^2)} \\
&\leq \frac{\alpha_t^2 L \|W_t\|_\infty^2}{\tau_{\max}^2} \|W_t - W^*\|^2 + \frac{\alpha_t (w_t - w^*) w_t \sigma \sqrt{D}}{2\tau_{\max}} \sqrt{(1 - \|W_t\|_\infty^2/\tau_{\max}^2)}.
\end{aligned} \tag{51}$$

The inequality holds for $\langle W_t - W^*,\ -\nabla f(W_t) \rangle = -\langle W_t - W^*,\ \nabla f(W_t) \rangle \leq \|W_t - W^*\| \cdot \|\nabla f(W_t)\| \leq L\|W_t - W^*\|^2$. Simplifying the second term as:

$$\begin{aligned}
&\sum_{i=1}^{D} \big[p_t([\nabla f(W_t)]_i + \varepsilon_t^+)^2 - (1-p_t)([\nabla f(W_t)]_i + \varepsilon_t^-)^2\big] \\
&= \sum_{i=1}^{D} \big[p_t\big([\nabla f(W_t)]_i^2 + 2[\nabla f(W_t)]_i \frac{\sigma}{\sqrt{D}} \sqrt{\frac{1-p_t}{p_t}} + (\frac{\sigma}{\sqrt{D}} \sqrt{\frac{1-p_t}{p_t}})^2\big) \\
&\quad - (1-p_t)\big([\nabla f(W_t)]_i^2 + 2[\nabla f(W_t)]_i (-\frac{\sigma}{\sqrt{D}} \sqrt{\frac{p_t}{1-p_t}}) + (-\frac{\sigma}{\sqrt{D}} \sqrt{\frac{p_t}{1-p_t}})^2\big)\big] \\
&= \sum_{i=1}^{D} \big[(1-2p_t)(-[\nabla f(W_t)]_i^2 + \frac{\sigma^2}{D}) + [\nabla f(W_t)]_i \frac{\sigma}{\sqrt{D}} \sqrt{(1 - \|W_t\|_\infty^2/\tau_{\max}^2)}\big]
\end{aligned} \tag{52}$$

$$= -\frac{L^2\|W_t\|_\infty^2}{\tau_{\max}}\|W_t - W^*\|^2 + \frac{\|W_t\|_\infty\sigma^2}{\tau_{\max}} + L\sigma\sqrt{D}(w_t - w^*)\sqrt{(1 - \|W_t\|_\infty^2/\tau_{\max}^2)}.$$

Substituting (49) and (50) into (46), we get:

$$\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1} - W^*\|^2] \qquad (53)\\
&\geq \big(1 - 2\alpha_t L + \alpha_t^2 L^2(1 + \frac{\|W_t\|_\infty^2}{\tau_{\max}^2}) - 2(1+\alpha_t L)\frac{\alpha_t L\|W_t\|_\infty^2}{\tau_{\max}^2}\big)\|W_t - W^*\|^2\\
&\quad + \alpha_t^2\sigma^2(1 + \frac{\|W_t\|_\infty^2}{\tau_{\max}^2}) + \frac{2\alpha_t^2\|W_t\|_\infty^2\sigma^2}{\tau_{\max}^2} + \Theta(\alpha_t\Delta w_{\min}).\\
&= \big(1 - 2\alpha_t L(1 - \frac{\alpha_t L}{2})(1 - \frac{\|W_t\|_\infty^2}{\tau_{\max}^2})\big)\|W_t - W^*\|^2 + \alpha_t^2\sigma^2(1 + \frac{3\|W_t\|_\infty^2}{\tau_{\max}^2}) + \Theta(\alpha_t\Delta w_{\min}).
\end{aligned}$$

Rearranging (53) as:

$$\begin{aligned}
&\|W_t - W^*\|^2 \qquad (54)\\
&\geq \frac{\|W_t - W^*\|^2 - \mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1} - W^*\|^2]}{2\alpha_t L(1 - \alpha_t L/2)(1 - \|W_t\|_\infty^2/\tau_{\max}^2)} + \frac{(1 + 3\|W_t\|_\infty^2/\tau_{\max}^2)\alpha_t\sigma^2}{2L(1 - \alpha_t L/2)(1 - \|W_t\|_\infty^2/\tau_{\max}^2)}\\
&\quad + \frac{\Theta(\Delta w_{\min})}{2L(1 - \alpha_t L/2)(1 - \|W_t\|_\infty^2/\tau_{\max}^2)}\\
&\geq \frac{\|W_t\|_\infty^2/\tau_{\max}^2\sigma^2}{L^2(1 - \|W_t\|_\infty^2/\tau_{\max}^2)} + \frac{2\Theta(\Delta w_{\min})}{3L(1 - \|W_t\|_\infty^2/\tau_{\max}^2)}.
\end{aligned}$$

The inequality holds since $\alpha_t = \frac{1}{2L}$, and $\|W_t - W^*\|^2 \geq \mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1} - W^*\|^2]$ from (Wu et al., 2024, Theorem 8). Taking the expectation over all ξ_t, ζ_t and take the average of (54) for t from 0 to $T-1$, we obtain:

$$\begin{aligned}
&\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}[\|W_t - W^*\|^2] \qquad (55)\\
&\geq \sigma^2 \cdot \frac{1}{T}\sum_{t=0}^{T-1}\frac{\|W_t\|_\infty^2/\tau_{\max}^2}{L^2(1 - \|W_t\|_\infty^2/\tau_{\max}^2)} + \frac{1}{T}\sum_{t=0}^{T-1}\frac{2\Theta(\Delta w_{\min})}{3L(1 - \|W_t\|_\infty^2/\tau_{\max}^2)}\\
&= \Omega(\sigma^2 S_T + R_T\Delta w_{\min}).
\end{aligned}$$

The proof of Theorem 2 is thus completed. □

F PROOF OF THEOREM 3 AND COROLLARY 1: CONVERGENCE OF RESIDUAL LEARNING

This section provides the convergence of residual learning algorithm under Assumptions 1–4. To formalize the analysis, we first clarify the use of tile-specific update indices. In the main text, we define each gradient update as one training step, denoted by the global counter t. Since each tile $W^{(n)}$ is updated only once every T_{n+1} updates of tile $W^{(n+1)}$. As a result, $W^{(n)}$ is not updated at every global step, exhibiting an inherently asynchronous update pattern. We introduce a local update counter t_n for each tile $W^{(n)}$, which tracks the number of its own update steps. These local counters are related to the global counter t by the following approximate relation:

$$t_n = \left\lfloor \frac{t+1}{\prod_{n'=n+1}^{N} T_{n'}} \right\rfloor.$$

As shown in Figure 6, the update schedule exhibits a nested timing hierarchy where inner tiles are updated less frequently.

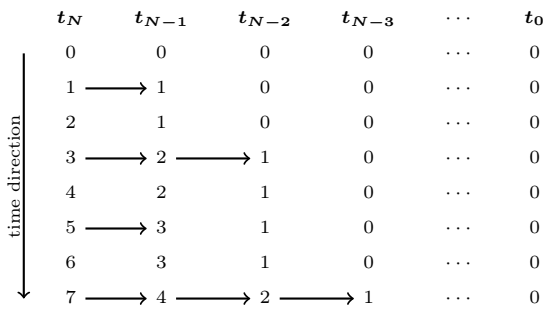


Figure 6: Local index evolution with $T_n = 2$. Arrows show transfers from $W^{(n+1)}$ to $W^{(n)}$.

F.1 Main proof

Theorem 6 (Convergence of residual learning, long version of Theorem 3)**.** *Suppose Assumptions 1–4 hold. Let the scaling parameter satisfy $\gamma \in (0, H_{\min}/\sqrt{6}F_{\max}^2]$, and set the mapping constant as $\kappa = (\sigma L_G W_{\max})^{\frac{1}{2}}(\gamma^N \Delta w_{\min})^{-\frac{1}{4}}$. For all $n \in \{0, \dots, N-1\}$, set the learning rate $\beta = \Theta(\gamma^2)$. Choose*

$$T_n \geq \Theta(\gamma^{-1}),$$

for $n \in \{0, \dots, N-1\}$, except that $T_0 = \Theta(1)$. For $n = N$, set $\alpha = \Theta(1)$ and choose

$$T_N \geq \Theta(\gamma^{-N}).$$

Define the Lyapunov sequence as:

$$\mathbb{J}_k := \sum_{n=0}^{N} \|W^{(n)}_{t_n + kT_n - 1} - P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})\|^2.$$

Since $t = \prod_{n=0}^{N} T_n k = \mathcal{O}(\gamma^{-2N})k$ is the total number of gradient evaluations, with $\rho \in (0, \frac{2}{3})$, the Lyapunov function $\mathbb{J}_k$ satisfies:

$$\mathbb{E}[\mathbb{J}_k] \leq \mathcal{O}((1-\rho)^{\gamma^{2N} t})\mathbb{E}[\mathbb{J}_0] + \Theta\big(\gamma^{-\frac{4N}{3}}(\sigma \Delta w_{\min})^{\frac{2}{3}}\big).$$

Proof of Theorem 6. We begin by presenting two lemmas essential for establishing the convergence proof.

One inner loop contraction.

Lemma 8 establishes that tile $W^{(N)}$ undergoes a descent in expected distance to its local stationary point $P_N^*(\overline{W}^{(N)})$ one inner loop with T_N updates. The update dynamic is defined as:

$$W_{t+1}^{(N)} = W_t^{(N)} - \alpha \nabla f(\overline{W}_t; \xi_t) \odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big) - |\alpha \nabla f(\overline{W}_t; \xi_t)| \odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big) + \zeta_t. \tag{56}$$

Lemma 8 (Descent lemma of the main sequence $W^{(N)}$, long version of Lemma 2)**.** *Suppose Assumptions 1 –4 hold, the learning rate satisfies $\alpha \leq \frac{C_{k,+}}{4\gamma^N(\mu+L)F_{\max}^2}$, the mapping constant is set as $\kappa = (\sigma L_G W_{\max})^{\frac{1}{2}}(\gamma^N \Delta w_{\min})^{-\frac{1}{4}}$. Denote $\mathbb{E}_{\boldsymbol{\xi_N}, \boldsymbol{\zeta_N}} := \mathbb{E}_{\xi_{t:t+T_N-1}, \zeta_{t:t+T_N-1}}$. It holds that:*

$$\begin{aligned} &\mathbb{E}_{\boldsymbol{\xi_N}, \boldsymbol{\zeta_N}}\left[\|W^{(N)}_{t+(k+1)T_N-1} - P_N^*(\overline{W}^{(N)}_{t_{N-1}+k})\|^2\right] \\ &\leq \left(1 - \frac{\alpha\mu L\gamma^N}{4(\mu+L)}\right)^{T_N} \|W^{(N)}_{t+kT_N} - P_N^*(\overline{W}^{(N)}_{t_{N-1}+k})\|^2 + \frac{8(\mu+L)\alpha}{\gamma^N \mu L} F_{\max}^2 \sigma^2 + \gamma^{-\frac{4N}{3}}\Theta((\sigma\Delta w_{\min})^{\frac{2}{3}}). \end{aligned} \tag{57}$$

Lemma 9 establishes that a single update of tile $W^{(n)}$ leads to a descent in its expected distance to the local stationary point $P_n^*(\overline{W}^{(n)})$ after one inner loop with T_n updates. The update dynamic is defined as:

$$W^{(n)}_{t_n+1} = W^{(n)}_{t_n} + \beta W^{(n+1)}_{t_{n+1}+T_{n+1}-1} \odot F\Big(\frac{W^{(n)}_{t_n}}{\kappa}\Big) - |\beta W^{(n+1)}_{t_{n+1}+T_{n+1}-1}| \odot G\Big(\frac{W^{(n)}_{t_n}}{\kappa}\Big) + \zeta_{t_n}. \tag{58}$$

Lemma 9 (Descent lemma of lower level sequences $W^{(n)}$, long version of Lemma 3)**.** *Following the same assumptions of Lemma 8, for $n \in \{0, \dots, N-1\}$, the learning rate satisfies that $\beta \le \frac{F_{\max}^3 \gamma}{3H_{\min}}$. Denote $\mathbb{E}_{\boldsymbol{\zeta_n}} := \mathbb{E}_{\zeta_{t_n+kT_n:t_n+(k+1)T_n-1}}$. It holds that:*

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta_n}}[\|W^{(n)}_{t_n+(k+1)T_n-1} - P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})\|^2] \\
\le& \left(1 - \frac{\beta H_{\min}}{2\gamma F_{\max}}\right)^{T_n} \|W^{(n)}_{t_n+kT_n} - P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})\|^2 \\
&+ \frac{6F_{\max}^4\gamma^2}{H_{\min}^2}\|W^{(n+1)}_{t_{n+1}+(k+1)T_{n+1}-1} - P_{n+1}^*(\overline{W}^{(n+1)}_{t_n+k})\|^2 + \frac{2F_{\max}\Theta(\Delta w_{\min})}{H_{\min}} \\
&- (\frac{2\gamma^2}{H_{\min}} - \frac{6\beta\gamma F_{\max}}{H_{\min}})\Big\|P_{n+1}^*(\overline{W}^{(n+1)}_{t_n+k}) \odot F\Big(\frac{W^{(n)}_{t_n}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}^{(n+1)}_{t_n+k})| \odot G\Big(\frac{W^{(n)}_{t_n}}{\kappa}\Big)\Big\|^2.
\end{aligned} \tag{59}$$

The proof of Lemma 8 and 9 are deferred to Section F.2 and F.3. We then proceed to prove the convergence of the algorithm with the result of Lemma 8 and 9 . Define a Lyapunov function as:

$$\mathbb{J}_k := \sum_{n=0}^{N} \|W^{(n)}_{t_n+kT_n-1} - P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})\|^2. \tag{60}$$

Bounding the drifting optimality gap.

To derive the recursion of the Lyapunov function, we require an additional inequality to characterize the *drift optimality*, when $n \in [1, N]$. Observe that between one time step increment on t_{n-1}, only the component $W^{(n-1)}$ of the stationary point $P_n^*(\overline{W}^{(n)}) = \gamma^{-n}(W^* - \overline{W}^{(n)})$ is updated due to the structure of the inner-loop algorithm we obtain:

$$\begin{aligned}
&\mathbb{E}_{\zeta_{t_n}}[\|P_n^*(\overline{W}^{(n)}_{t_{n-1}+k+1}) - P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})\|^2] \\
=&\frac{1}{\gamma}\mathbb{E}_{\zeta_{t_n}}\Big[\Big\|\beta\Big(W^{(n)}_{t_n+kT_n-1} \odot F\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big) - |W^{(n)}_{t_n+kT_n-1}| \odot G\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big)\Big) + \zeta_{t_n}\Big\|^2\Big] \\
\le&\frac{3\beta^2}{\gamma}\Big\|P_n^*(\overline{W}^{(n)}_{t_{n-1}+k}) \odot F\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big) - |P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})| \odot G\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big)\Big\|^2 + \frac{3\beta^2}{\gamma}\Big\|W^{(n)}_{t_n+kT_n-1} \\
&\odot F\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big) - |W^{(n)}_{t_n+kT_n-1}| \odot G\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big) - \Big(P_n^*(\overline{W}^{(n)}_{t_{n-1}+k}) \odot F\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big) \\
&- |P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})| \odot G\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big)\Big)\Big\|^2 + \frac{\mathbb{E}_{\zeta_{t_n}}[\zeta_{t_n}^2]}{\gamma} \\
\le& \frac{3\beta^2}{\gamma}\Big\|P_n^*(\overline{W}^{(n)}_{t_{n-1}+k}) \odot F\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big) - |P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})| \odot G\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big)\Big\|^2 \\
&+ \frac{3\beta^2}{\gamma}\|W^{(n)}_{t_n+kT_n-1} - P_n^*(\overline{W}^{(n)}_{t_{n-1}+k})\|^2 + \frac{3\beta}{\gamma}\Theta(\Delta w_{\min}).
\end{aligned} \tag{61}$$

Inequality (61) is obtained by Lemma 1, 5 and Cauchy-Schwarz inequality. Therefore, the *drift optimality* can be bounded by substituting (61) with $\|U+V\|^2 \le 2\|U\|^2 + 2\|V\|^2$:

$$\mathbb{E}_{\zeta_{t_n}}[\|W^{(n)}_{t_n+(k+1)T_n-1} - P_n^*(\overline{W}^{(n)}_{t_{n-1}+k+1})\|^2] \tag{62}$$

$$\leq 2\|W^{(n)}_{t_n+(k+1)T_n-1} - P^*_n(\overline{W}^{(n)}_{t_{n-1}+k})\|^2 + 2\mathbb{E}_{\zeta_{t_n}}[\|P^*_n(\overline{W}^{(n)}_{t_{n-1}+k+1}) - P^*_n(\overline{W}^{(n)}_{t_{n-1}+k})\|^2]$$
$$\leq (2+\frac{6\beta^2}{\gamma})\|W^{(n)}_{t_n+(k+1)T_n-1} - P^*_n(\overline{W}^{(n)}_{t_{n-1}+k})\|^2 + \frac{6\beta^2}{\gamma}\Big\|P^*_n(\overline{W}^{(n)}_{t_{n-1}+k})\odot F\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big)$$
$$- |P^*_n(\overline{W}^{(n)}_{t_{n-1}+k})|\odot G\Big(\frac{W^{(n-1)}_{t_{n-1}+k}}{\kappa}\Big)\Big\|^2 + \frac{6\beta}{\gamma}\Theta(\Delta w_{\min}).$$

Establishing convergence.

Define $\mathcal{F}_k$ as the σ-algebra generated by all random variables up to time k, including $\{W^{(n)}_{t_n}\}_{t\leq k, n\in[0,N]}$. We write conditional expectations $\mathbb{E}[\,\cdot\mid\mathcal{F}_k]$ compactly as $\mathbb{E}_k[\,\cdot\,]$ for brevity. For notational consistancy, we denote $W^* =: P^*_{0,t_{-1}+k}$ for all k. Expanding $\mathbb{J}_{k+1}$ with (62), (57) and (59), we get:

$$\mathbb{E}_k[\mathbb{J}_{k+1}] - \mathbb{J}_k \tag{63}$$
$$= \sum_{n'=0}^{N}\|W^{(n')}_{t'_n+(k+1)T'_n-1} - P^*_N(\overline{W}^{(n')}_{t_{n'-1}+k+1})\|^2 - \sum_{n'=0}^{N}\|W^{(n')}_{t'_n+kT'_n-1} - P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\|^2$$
$$\leq \sum_{n'=1}^{N}\Bigg((2+\frac{6\beta^2}{\gamma})\|W^{(n')}_{t_{n'}+(k+1)T'_n-1} - P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\|^2 + \frac{6\beta^2}{\gamma}\Big\|P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\odot F\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big)$$
$$- |P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})|\odot G\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big)\Big\|^2 + \frac{6\beta}{\gamma}\Theta(\Delta w_{\min})\Bigg) + \|W^{(0)}_{t_0+(k+1)T_0-1} - P^*_{0,t_{-1}+k}\|^2$$
$$- \sum_{n'=0}^{N}\|W^{(n')}_{t'_n+kT'_n-1} - P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\|^2$$
$$\leq (2+\frac{6\beta^2}{\gamma})\Big((1-\frac{\alpha\mu L\gamma^N}{4(\mu+L)})^{T_N}\|W^{(N)}_{t+kT_N-1} - P^*_N(\overline{W}^{(N)}_{t_{N-1}+k})\|^2 + \frac{8(\mu+L)\alpha}{\gamma^N\mu L}F^2_{\max}\sigma^2 + \gamma^{-\frac{4N}{3}}$$
$$\Theta((\sigma\Delta w_{\min})^{\frac{2}{3}})\Big) + \frac{6\beta^2}{\gamma}\Big\|P^*_N(\overline{W}^{(N)}_{t_{N-1}+k})\odot F\Big(\frac{W^{(N-1)}_{t_{N-1}+k}}{\kappa}\Big) - |P^*_N(\overline{W}^{(N)}_{t_{N-1}+k})|\odot G\Big(\frac{W^{(N-1)}_{t_{N-1}+k}}{\kappa}\Big)\Big\|^2$$
$$+ \frac{6\beta}{\gamma}\Theta(\Delta w_{\min}) + \sum_{n'=1}^{N-1}\Bigg(\Big(2+\frac{6\beta^2}{\gamma}\Big)\Big((-\frac{2\gamma^2}{H_{\min}} + \frac{6\beta\gamma F_{\max}}{H_{\min}})\Big\|P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\odot F\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big)$$
$$- |P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})|\odot G\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big)\Big\|^2 + \big(1-\frac{\beta H_{\min}}{2\gamma F_{\max}}\big)^{T_{n'}}\|W^{(n')}_{t_{n'}+kT'_n-1} - P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\|^2$$
$$+ \frac{6F^4_{\max}\gamma^2}{H^2_{\min}}\|W^{(n'+1)}_{t_{n'+1}+kT_{n'+1}-1} - P^*_{n'}(\overline{W}^{(n'+1)}_{t_{n'}+k})\|^2 + \frac{2F_{\max}\Theta(\Delta w_{\min})}{H_{\min}}\Big) + \frac{6\beta^2}{\gamma}\Big\|P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})$$
$$\odot F\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big) - |P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})|\odot G\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big)\Big\|^2 + \frac{6\beta}{\gamma}\Theta(\Delta w_{\min})\Bigg)$$
$$+ \left(1-\frac{\beta H_{\min}}{2\gamma F_{\max}}\right)^{T_0}\|W^{(0)}_{t_0+kT_0} - P^*_{0,t_{-1}+k}\|^2 + \frac{6F^4_{\max}\gamma^2}{H^2_{\min}}\|W^{(1)}_{t_1+(k+1)T_1-1} - P^*_1(\overline{W}^{(1)}_{t_0+k})\|^2$$
$$+ \frac{2F_{\max}\Theta(\Delta w_{\min})}{H_{\min}} - \sum_{n'=0}^{N}\|W^{(n')}_{t'_n+kT_n-1} - P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\|^2$$
$$\leq \sum_{n'=1}^{N}\underbrace{\Big((2+\frac{6\beta^2}{\gamma})(-\frac{2\gamma^2}{H_{\min}} + \frac{6\beta\gamma F_{\max}}{H_{\min}}) + \frac{6\beta^2}{\gamma}\Big)}_{(A)}\Big\|P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\odot F\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big) - |P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})|$$

$$\odot\, G\Big(\frac{W^{(n'-1)}_{t_{n'-1}+k}}{\kappa}\Big)\Big\|^2 + \sum_{n'=1}^{N-1} \underbrace{\Big((2+\frac{6\beta^2}{\gamma})(1-\frac{\beta H_{\min}}{2\gamma F_{\max}})^{T_{n'}} + \frac{6F_{\max}^4\gamma^2}{H_{\min}^2} - 1\Big)}_{(B)} \|W^{(n')}_{t_{n'}+kT'_n-1} - P^*_{n'}(\overline{W}^{(n')}_{t_{n'-1}+k})\|^2$$

$$+ \underbrace{\Big((2+\frac{6\beta^2}{\gamma})(1-\frac{\alpha\mu L\gamma^N}{4(\mu+L)})^{T_N} + \frac{6F_{\max}^4\gamma^2}{H_{\min}^2} - 1\Big)}_{(C)} \|W^{(N)}_{t+kT_N-1} - P^*_N(\overline{W}^{(N)}_{t_{N-1}+k})\|^2 + (2+\frac{6\beta^2}{\gamma})$$

$$(\frac{8(\mu+L)\alpha}{\gamma^N\mu L}F_{\max}^2\sigma^2 + \gamma^{-\frac{4N}{3}}\Theta((\sigma\Delta w_{\min})^{\frac{2}{3}})) + \sum_{n'=0}^{N-1}\Big((2+\frac{6\beta^2}{\gamma})\frac{2F_{\max}\Theta(\Delta w_{\min})}{H_{\min}} + \frac{6\beta}{\gamma}\Theta(\Delta w_{\min})\Big)$$

$$+ \Big((1-\frac{\beta H_{\min}}{2\gamma F_{\max}})^{T_0} - 1\Big)\|W^{(0)}_{t_0+kT_0-1} - P^*_{0,t_{-1}+k}\|^2.$$

The coefficient $\Big((1-\frac{\beta H_{\min}}{2\gamma F_{\max}})^{T_{n'}} - 1\Big)$ for the last term is negative when $T_0 \geq 1$. To achieve descent on Lyapunov function, we choose

$$\beta = \Theta(\gamma^2), \quad \alpha = \Theta(1), \quad \gamma \leq \sqrt{\frac{1-\frac{3}{2}\rho}{6}}\frac{H_{\min}}{F_{\max}^2}, \quad T_n \geq \Theta\big(\gamma^{-1}\big), n \in \{1,\dots,N-1\}, \quad T_N \geq \Theta\big(\gamma^{-N}\big)$$

to satisfy the following conditions, where $\rho \in (0, \frac{2}{3})$ is a constant:

$$A := (2+\frac{6\beta^2}{\gamma})(-\frac{2\gamma^2}{H_{\min}} + \frac{6\beta\gamma F_{\max}}{H_{\min}}) + \frac{6\beta^2}{\gamma} \leq 0, \tag{64}$$

$$B := (2+\frac{6\beta^2}{\gamma})\big(1-\frac{\beta H_{\min}}{2\gamma F_{\max}}\big)^{T_n} + \frac{6F_{\max}^4\gamma^2}{H_{\min}^2} - 1 \leq -\rho, \quad n \in [1, N-1], \tag{65}$$

$$C := (2+\frac{6\beta^2}{\gamma})\big(1-\frac{\alpha\mu L\gamma^N}{4(\mu+L)}\big)^{T_N} + \frac{6F_{\max}^4\gamma^2}{H_{\min}^2} - 1 \leq -\rho. \tag{66}$$

Let $\beta = c_\beta\gamma^2$, the inequality (64) becomes:

$$\big(2+6c_\beta^2\gamma^3\big)\big(-2\gamma^2 + 6c_\beta F_{\max}\gamma^3\big) + 6c_\beta^2\gamma^3 H_{\min} \leq 0 \tag{67}$$

which holds when $\gamma \in (0, \frac{H_{\min}}{\sqrt{6}F_{\max}^2}]$ and $c_\beta \in (0, \frac{1}{5F_{\max}}]$. For (65) and (66), we first give the upper bound of γ as $\gamma \leq \frac{\sqrt{1-\frac{3}{2}\rho}H_{\min}}{\sqrt{6}F_{\max}^2}$ in order to satisfy $\frac{6F_{\max}^4\gamma^2}{H_{\min}^2} - 1 \leq -\frac{3}{2}\rho$. We then bound the inner loops count T_n, $n \in \{0,\dots,N-1\}$ by defining :

$$\lambda_1 = \frac{\beta H_{\min}}{2\gamma F_{\max}} = \frac{c_\beta H_{\min}}{2F_{\max}}\gamma =: c_1\gamma = \Theta(\gamma), \qquad A_1 = 2 + \frac{6\beta^2}{\gamma} = 2 + 6c_\beta^2\gamma^3 = \mathcal{O}(1). \tag{68}$$

Using $(1-\lambda_1)^{T_n} \leq e^{-\lambda_1 T_n}$, inequality (65) must satisfy $A_1 e^{-\lambda_1 T_n} \leq \rho/2$, which yields $T_n \geq \frac{1}{\lambda_1}\ln\Big(\frac{2A_1}{\rho}\Big) = \Theta\left(\gamma^{-1}\right)$. Similarly, let $\lambda_2 = \frac{\alpha\mu L\gamma^N}{4(\mu+L)} =: c_2\gamma^N = \Theta(\gamma^N)$, to satisfy $A_1 e^{-\lambda_2 T_N} \leq \rho/2$, we bound T_N as $T_N \geq \frac{1}{\lambda_2}\ln\Big(\frac{2A_1}{\rho}\Big) = \Theta\left(\gamma^{-N}\right)$. Once inequalities (64), (65), and (66) are satisfied, the Lyapunov descent inequality (63) simplifies to:

$$\mathbb{E}_k[\mathbb{J}_{k+1}] \leq (1-\rho)\mathbb{J}_k + \gamma^{-\frac{4N}{3}}\Theta\big((\sigma\Delta w_{\min})^{\frac{2}{3}}\big). \tag{69}$$

Taking full expectation over $\mathcal{F}_k$ on both sides, the recurrence becomes:

$$\mathbb{E}[\mathbb{J}_{k+1}] \leq (1-\rho)\mathbb{E}[\mathbb{J}_k] + \gamma^{-\frac{4N}{3}}\Theta\big((\sigma\Delta w_{\min})^{\frac{2}{3}}\big). \tag{70}$$

Unrolling this inequality over k steps, we get:

$$\mathbb{E}[\mathbb{J}_k] \le (1-\rho)^k \mathbb{E}[\mathbb{J}_0] + \gamma^{-\frac{4N}{3}} \Theta\big((\sigma \Delta w_{\min})^{\frac{2}{3}}\big) \sum_{t=0}^{K-1} (1-\rho)^t \tag{71}$$

$$\le (1-\rho)^k \mathbb{E}[\mathbb{J}_0] + \frac{\gamma^{-\frac{4N}{3}} \Theta\big((\sigma \Delta w_{\min})^{\frac{2}{3}}\big)}{\rho}.$$

Since $\rho \in (0,1) = \Theta(1)$, we get:

$$\mathbb{E}[\mathbb{J}_k] \le (1-\rho)^k \mathbb{E}[\mathbb{J}_0] + \Theta\big(\gamma^{-\frac{4N}{3}} (\sigma \Delta w_{\min})^{\frac{2}{3}}\big). \tag{72}$$

Since $W^{(N)}$ dominates the compute cost, one outer iteration (i.e. one update of $\mathbb{J}_k$) consumes $t = \prod_{n=0}^{N} T_n k = \mathcal{O}(\gamma^{-2N}) \cdot k$ gradient evaluations. Therefore the averaged Lyapunov function, as a function of the total gradient evaluations, obeys:

$$\mathbb{E}[\mathbb{J}_k] \le \mathcal{O}((1-\rho)^{\gamma^{2N} t}) \mathbb{E}[\mathbb{J}_0] + \Theta\big(\gamma^{-\frac{4N}{3}} (\sigma \Delta w_{\min})^{\frac{2}{3}}\big)$$

which completes the proof. □

Corollary 1 (Optimality gap of residual learning)**.** *Under the same conditions as in Theorem 3, the limit of the composited weight $\overline{W}_t$ satisfies:*

$$\limsup_{t \to \infty} \mathbb{E}\big[\|W^* - \overline{W}_t\|^2\big] \le \Theta(\gamma^{\frac{2N}{3}} (\sigma \Delta w_{\min})^{\frac{2}{3}}).$$

Proof of Corollary 1. From (71), taking the limit as $k \to \infty$, we obtain:

$$\limsup_{k \to \infty} \mathbb{E}[\mathbb{J}_k] \le \Theta(\gamma^{-\frac{4N}{3}} (\sigma \Delta w_{\min})^{\frac{2}{3}}). \tag{73}$$

Once (73) holds for $\mathbb{J}_k$, each component of $\mathbb{J}_k$ also satisfies:

$$\limsup_{t_n \to \infty} \mathbb{E}\big[\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2\big] \le \Theta\big(\gamma^{-\frac{4N}{3}} (\sigma \Delta w_{\min})^{\frac{2}{3}}\big). \tag{74}$$

Substituting $P_n^*(\overline{W}^{(n)}) = \gamma^{-n}(W^* - \overline{W}^{(n)})$ into (74), we derive the bound on the scaled residual:

$$\limsup_{t_n \to \infty} \mathbb{E}\big[\|W^* - \overline{W}_{t_n}^{(n+1)}\|^2\big] \le \Theta\big(\gamma^{2n-\frac{4N}{3}} (\sigma \Delta w_{\min})^{\frac{2}{3}}\big), \quad n \in [0, N]. \tag{75}$$

In particular, when $n = N$, we obtain the desired result as:

$$\limsup_{t \to \infty} \mathbb{E}\big[\|W^* - \overline{W}_t\|^2\big] \le \Theta\big(\gamma^{\frac{2N}{3}} (\sigma \Delta w_{\min})^{\frac{2}{3}}\big). \tag{76}$$

This demonstrates that increasing the number of tiles leads to an error decay rate proportional to $\gamma^{\frac{2}{3}}$, thereby completing the proof. □

F.2 Proof of Lemma 8: Descent of the main sequence $W_t^{(N)}$

Lemma 8 (Descent lemma of the main sequence $W^{(N)}$, long version of Lemma 2)**.** *Suppose Assumptions 1 –4 hold, the learning rate satisfies $\alpha \le \frac{C_{k,+}}{4\gamma^N(\mu+L)F_{\max}^2}$, the mapping constant is set as $\kappa = (\sigma L_G W_{\max})^{\frac{1}{2}} (\gamma^N \Delta w_{\min})^{-\frac{1}{4}}$. Denote $\mathbb{E}_{\boldsymbol{\xi_N}, \boldsymbol{\zeta_N}} := \mathbb{E}_{\xi_{t:t+T_N-1}, \zeta_{t:t+T_N-1}}$. It holds that:*

$$\mathbb{E}_{\boldsymbol{\xi_N}, \boldsymbol{\zeta_N}} \left[\|W_{t+(k+1)T_N-1}^{(N)} - P_N^*(\overline{W}_{t_{N-1}+k}^{(N)})\|^2\right] \tag{57}$$

$$\le \left(1 - \frac{\alpha\mu L\gamma^N}{4(\mu+L)}\right)^{T_N} \|W_{t+kT_N}^{(N)} - P_N^*(\overline{W}_{t_{N-1}+k}^{(N)})\|^2 + \frac{8(\mu+L)\alpha}{\gamma^N \mu L} F_{\max}^2 \sigma^2 + \gamma^{-\frac{4N}{3}} \Theta((\sigma \Delta w_{\min})^{\frac{2}{3}}).$$

Proof of Lemma 8. The proof begins from manipulating the norm $\|W_{t+1}^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2$

$$\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1}^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2] \\
&= \|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 + 2\mathbb{E}_{\xi_t,\zeta_t}[\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), W_{t+1}^{(N)} - W_t^{(N)}\rangle] \\
&\quad + \mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1}^{(N)} - W_t^{(N)}\|^2].
\end{aligned} \tag{77}$$

To bound the second term, we first apply the update dynamics given in (56) to obtain the following equality:

$$\begin{aligned}
&2\mathbb{E}_{\xi_t,\zeta_t}[\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), W_{t+1}^{(N)} - W_t^{(N)}\rangle] \\
=& -2\mathbb{E}_{\xi_t,\zeta_t}\Big[\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \alpha\nabla f(\overline{W}_t;\xi_t)\odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big) + |\alpha\nabla f(\overline{W}_t;\xi_t)| \\
&\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big) - \zeta_t\Big\rangle\Big] \\
=& -2\alpha\mathbb{E}_{\xi_t}\Big[\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t;\xi_t)\odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle\Big] \\
&-2\alpha\mathbb{E}_{\xi_t}\Big[\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), |\nabla f(\overline{W}_t;\xi_t)|\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle\Big] \\
=& -2\alpha\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t)\odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle \\
&+2\alpha\mathbb{E}_{\xi_t}\Big[\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), (|\nabla f(\overline{W}_t)| - |\nabla f(\overline{W}_t;\xi_t)|)\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle\Big] \\
&-2\alpha\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), |\nabla f(\overline{W}_t)|\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle \\
\le& \underbrace{-2\alpha\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t)\odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big) + |\nabla f(\overline{W}_t)|\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle}_{(T1)} \\
&\underbrace{+2\alpha\,\mathbb{E}_{\xi_t}\Big[\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), (|\nabla f(\overline{W}_t)| - |\nabla f(\overline{W}_t;\xi_t)|)\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle\Big]}_{(T2)}
\end{aligned} \tag{78}$$

Upper bound of the first term $(T1)$**.** With Lemma 7, the second term in the RHS of (77) can be bounded by:

$$\begin{aligned}
&-2\alpha\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t)\odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big) + |\nabla f(\overline{W}_t)|\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle \\
=& -2\alpha\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t)\odot q_s\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle \\
\le& -2\alpha C_{k,+}\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t)\rangle + 2\alpha C_{k,-}\langle |W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})|, |\nabla f(\overline{W}_t)|\rangle
\end{aligned} \tag{79}$$

where $C_{k,+}$ and $C_{k,-}$ are defined as:

$$C_{k,+} := \frac{1}{2}\left(\max_{i\in\mathcal{I}}\{q_s([\frac{W_t^{(N)}}{\kappa}]_i)\} + \min_{i\in\mathcal{I}}\{q_s([\frac{W_t^{(N)}}{\kappa}]_i)\}\right), \tag{80}$$

$$C_{k,-} := \frac{1}{2}\left(\max_{i\in\mathcal{I}}\{q_s([\frac{W_t^{(N)}}{\kappa}]_i)\} - \min_{i\in\mathcal{I}}\{q_s([\frac{W_t^{(N)}}{\kappa}]_i)\}\right). \tag{81}$$

In the inequality above, the first term can be bounded by the strong convexity of f. Let $\varphi(W^{(N)}) := f(\overline{W}^{(N)} + \gamma^N W^{(N)})$ which is $\gamma^{2N}L$-smooth and $\gamma^{2N}\mu$-strongly convex. It can be verified that $\varphi(W_t^{(N)})$ has gradient

$\nabla\varphi(W_t^{(N)}) = \gamma^N \nabla f(\overline{W}_{t_{N-1}}^{(N)} + \gamma^N W_t^{(N)}) = \gamma^N \nabla f(\overline{W}_{t_N})$ and optimal point $P^*(\overline{W}_{t_{N-1}}^{(N)})$. Leveraging Theorem 2.1.9 in (Nesterov et al., 2018), we have:

$$
\begin{aligned}
& 2\alpha C_{k,+}\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t)\rangle \\
=& \frac{2\alpha C_{k,+}}{\gamma^N}\langle \nabla\varphi(W_t^{(N)}), W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\rangle \\
=& \frac{2\alpha C_{k,+}}{\gamma^N}\langle \nabla\varphi(W_t^{(N)}) - \nabla\varphi(P_N^*(\overline{W}_{t_{N-1}}^{(N)})), W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\rangle \\
\geq & \frac{2\alpha C_{k,+}}{\gamma^N}\left(\frac{\gamma^{2N}\mu\cdot\gamma^{2N}L}{\gamma^{2N}\mu+\gamma^{2N}L}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 + \frac{1}{\gamma^{2N}\mu+\gamma^{2N}L}\|\nabla\varphi(W_t^{(N)})\|^2\right) \\
=& \frac{2\alpha C_{k,+}\mu L\gamma^N}{\mu+L}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 + \frac{2\alpha C_{k,+}}{\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2.
\end{aligned} \tag{82}
$$

The second term in the RHS of (79) can be bounded by the following inequality:

$$
\begin{aligned}
& 2\alpha C_{k,-}\Big\langle |W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})|, |\nabla f(\overline{W}_t)|\Big\rangle \\
\leq & \frac{\alpha C_{k,-}^2\gamma^N(\mu+L)}{C_{k,+}}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 + \frac{\alpha C_{k,+}}{\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2.
\end{aligned} \tag{83}
$$

Therefore, (79) becomes:

$$
\begin{aligned}
& -2\alpha\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), \nabla f(\overline{W}_t)\odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big) + |\nabla f(\overline{W}_t)|\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\rangle \\
\leq & -\gamma^N\left(\frac{2\alpha\mu LC_{k,+}}{\mu+L} - \frac{\alpha C_{k,-}^2(\mu+L)}{C_{k,+}}\right)\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 - \frac{\alpha C_{k,+}}{\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2.
\end{aligned} \tag{84}
$$

Upper bound of the second term $(T2)$**.** Leveraging the Young's inequality, we have:

$$
\begin{aligned}
& 2\alpha\mathbb{E}_{\xi_t}\Big[\Big\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), (|\nabla f(\overline{W}_t)| - |\nabla f(\overline{W}_t;\xi_t)|)\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\rangle\Big] \\
\leq & \frac{\alpha\mu LC_{k,+}\gamma^N}{(\mu+L)}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 \\
& + \frac{\alpha(\mu+L)}{\mu LC_{k,+}\gamma^N}\mathbb{E}_{\xi_t}\Big[\Big\|(|\nabla f(\overline{W}_t)| - |\nabla f(\overline{W}_t;\xi_t)|)\odot G\Big(\frac{1}{\kappa}W_t^{(N)}\Big)\Big\|^2\Big] \\
\overset{(a)}{\leq} & \frac{\alpha\mu LC_{k,+}\gamma^N}{(\mu+L)}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 \\
& + \frac{\alpha(\mu+L)}{\mu LC_{k,+}\gamma^N}\mathbb{E}_{\xi_t}\Big[\Big\|(|\nabla f(\overline{W}_t) - \nabla f(\overline{W}_t;\xi_t)|)\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\|^2\Big] \\
\overset{(b)}{=} & \frac{\alpha\mu LC_{k,+}\gamma^N}{(\mu+L)}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 + \frac{\alpha(\mu+L)\sigma^2}{\mu LC_{k,+}\gamma^N}\Big\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\|_\infty^2
\end{aligned} \tag{85}
$$

where (a) applies $||x|-|y|| \leq |x-y|$ for any $x, y \in \mathbb{R}$, (b) uses the bounded variance assumption (see Assumption 1). Combining the upper bound of $(T1)$ and $(T2)$, we bound (78) by:

$$
\begin{aligned}
& 2\mathbb{E}_{\xi_t}[\langle W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)}), W_{t+1}^{(N)} - W_t^{(N)}\rangle] \\
\leq & -\gamma^N\left(\frac{\alpha\mu LC_{k,+}}{\mu+L} - \frac{\alpha C_{k,-}^2(\mu+L)}{C_{k,+}}\right)\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 - \frac{\kappa^2\alpha C_{k,+}}{\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2 \\
& + \frac{\alpha(\mu+L)\sigma^2}{\mu LC_{k,+}\gamma^N}\left\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\right\|_\infty^2
\end{aligned} \tag{86}
$$

$$\leq -\frac{\gamma^N \alpha\mu L C_{k,+}}{2(\mu+L)}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 - \frac{\alpha C_{k,+}}{\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2 + \frac{\alpha(\mu+L)\sigma^2}{\mu L C_{k,+}\gamma^N}\left\|G(\frac{W_t^{(N)}}{\kappa})\right\|_\infty^2$$

where the last inequality holds for $C_{k,-} \ll C_{k,+}$, which is sufficiently close to 0, and the following inequality holds:

$$(\mu+L)\frac{C_{k,-}^2}{C_{k,+}^2} \leq \frac{\mu L}{2(\mu+L)}. \tag{87}$$

Furthermore, the last term in the RHS of (77) can be bounded by the Lipschitz continuity of analog update (see Lemma 5) and the bounded variance assumption (see Assumption 1) as:

$$\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}\Big[\Big\|W_{t+1}^{(N)} - W_t^{(N)}\|^2\Big] \qquad (88)\\
&= \mathbb{E}_{\xi_t,\zeta_t}[\|\alpha\nabla f(\overline{W}_t;\xi_t)\odot F\Big(\frac{W_t^{(N)}}{\kappa}\Big) - \alpha|\nabla f(\overline{W}_t;\xi_t)|\odot G\Big(\frac{W_t^{(N)}}{\kappa}\Big) + \zeta_t\Big\|^2\Big]\\
&\leq 2\alpha^2 F_{\max}^2\mathbb{E}_{\xi_t}[\|\nabla f(\overline{W}_t;\xi_t)\|^2] + 2\alpha\Theta(\Delta w_{\min})\\
&\leq 2\alpha^2 F_{\max}^2\|\nabla f(\overline{W}_t)\|^2 + 2\alpha^2 F_{\max}^2\sigma^2 + 2\alpha\Theta(\Delta w_{\min})\\
&\leq \frac{\alpha C_{k,+}}{2\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2 + 2\alpha^2 F_{\max}^2\sigma^2 + 2\alpha\Theta(\Delta w_{\min})
\end{aligned}$$

where the first inequality holds by $\|U+V\|^2 \leq 2\|U\|^2 + 2\|V\|^2$ and $\mathbb{E}_{\zeta_t}[\zeta_t^2] = \alpha\Theta(\Delta w_{\min})$, the last inequality holds if $\alpha \leq \frac{C_{k,+}}{4\gamma^N(\mu+L)F_{\max}^2}$. Plugging inequality (86) and (88) above into (77) yields:

$$\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1}^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2] \qquad (89)\\
\leq\ &\left(1 - \frac{\alpha\mu L C_{k,+}\gamma^N}{2(\mu+L)}\right)\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 - \frac{\alpha C_{k,+}}{2\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2\\
&+ \frac{\alpha(\mu+L)\sigma^2}{\mu L C_{k,+}\gamma^N}\Big\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\|_\infty^2 + 2\alpha^2 F_{\max}^2\sigma^2 + 2\alpha\Theta(\Delta w_{\min}).
\end{aligned}$$

From the definition of $C_{k,+}$, when the saturation degree of $W_t^{(N)}$ is properly limited, we have $C_{k,+} \geq \frac{1}{2}$ since $\alpha\gamma^n$ is sufficiently small. Therefore, we have:

$$\begin{aligned}
&\mathbb{E}_{\xi_t,\zeta_t}[\|W_{t+1}^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2] \qquad (90)\\
\leq\ &\left(1 - \frac{\mu L\alpha\gamma^N}{4(\mu+L)}\right)\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 - \frac{\alpha}{4\gamma^N(\mu+L)}\|\nabla f(\overline{W}_t)\|^2\\
&+ \frac{2\alpha(\mu+L)\sigma^2}{\mu L\gamma^N}\Big\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\|_\infty^2 + 2\alpha^2 F_{\max}^2\sigma^2 + 2\alpha\Theta(\Delta w_{\min}).
\end{aligned}$$

Under Assumption 4, which indicates $G(0) = 0$ and the Lipschitz continuity of the response functions, we can directly bound the term $\Big\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\|_\infty^2$ in (89) as:

$$\Big\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\|_\infty^2 \leq \Big\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big)\Big\|^2 = \Big\|G\Big(\frac{W_t^{(N)}}{\kappa}\Big) - G(0)\Big\|^2 \leq \frac{L_G^2}{\kappa^2}\|W_t^{(N)}\|_\infty^2 \tag{91}$$

where $L_G \geq 0$ is a Lipschitz constant. Perform T_N iterations using the recursive process in (89), and denote the expectation over the noise sequence $\mathbb{E}_{\xi_{t:t+T_N-1},\zeta_{t:t+T_N-1}}$ as $\mathbb{E}_{\boldsymbol{\xi_N},\boldsymbol{\zeta_N}}$, we obtain the following upper bound:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi_N},\boldsymbol{\zeta_N}}\left[\|W_{t+T_N-1}^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2\right] \qquad (92)\\
\leq &\left(1 - \frac{\alpha\mu L\gamma^N}{4(\mu+L)}\right)^{T_N}\|W_t^{(N)} - P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2 + \sum_{i=0}^{T_N-1}\left(1 - \frac{\alpha\mu L\gamma^N}{4(\mu+L)}\right)^i\Big(2\alpha^2 F_{\max}^2\sigma^2
\end{aligned}$$

$$
\begin{aligned}
&+\frac{2\alpha(\mu+L)\sigma^2L_G^2\|W_t^{(N)}\|_\infty^2}{\mu L\gamma^N\kappa^2}+2\alpha\Theta(\Delta w_{\min})\Big)\\
\le&\left(1-\frac{\alpha\mu L\gamma^N}{4(\mu+L)}\right)^{T_N}\|W_t^{(N)}-P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2+\frac{8(\mu+L)\alpha}{\gamma^N\mu L}F_{\max}^2\sigma^2+\frac{8(\mu+L)^2\sigma^2L_G^2W_{\max}^2}{\gamma^{2N}\mu^2L^2\kappa^2}\\
&+\frac{8(\mu+L)\kappa}{\gamma^N\mu L}\Theta(\Delta c_{\min})\\
\le&\left(1-\frac{\alpha\mu L\gamma^N}{4(\mu+L)}\right)^{T_N}\|W_t^{(N)}-P_N^*(\overline{W}_{t_{N-1}}^{(N)})\|^2+\frac{8(\mu+L)\alpha}{\gamma^N\mu L}F_{\max}^2\sigma^2+\gamma^{-\frac{4N}{3}}\Theta((\sigma\Delta w_{\min})^{\frac{2}{3}}).
\end{aligned}
$$

The second inequality holds for $\sum_{i=0}^{T_N-1}\left(1-\frac{\alpha\mu L\gamma^N}{8(\mu+L)}\right)^i\le\frac{8(\mu+L)}{\alpha\mu L\gamma^N}$, and we define $W_{\max}\in[\|W_t^{(N)}\|_\infty,\tau_{\max})$ for all t. The last inequality holds by choosing the mapping constant as:

$$
\kappa=(\sigma L_GW_{\max})^{\frac{2}{3}}(\gamma^N\Delta c_{\min})^{-\frac{1}{3}}=(\sigma L_GW_{\max})^{\frac{2}{3}}(\frac{\gamma^N\Delta w_{\min}}{\kappa})^{-\frac{1}{3}}. \tag{93}
$$

The second equality holds by substituting (8). Rearranging (93), we get:

$$
\kappa=(\sigma L_GW_{\max})^{\frac{1}{2}}(\gamma^N\Delta w_{\min})^{-\frac{1}{4}} \tag{94}
$$

When $k\neq0$, (92) can be written as the general case:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\xi_N},\boldsymbol{\zeta_N}}\left[\|W_{t+(k+1)T_N-1}^{(N)}-P_N^*(\overline{W}_{t_{N-1}+k}^{(N)})\|^2\right] \qquad (95)\\
\le&\left(1-\frac{\alpha\mu L\gamma^N}{4(\mu+L)}\right)^{T_N}\|W_{t+kT_N}^{(N)}-P_N^*(\overline{W}_{t_{N-1}+k}^{(N)})\|^2+\frac{8(\mu+L)\alpha}{\gamma^N\mu L}F_{\max}^2\sigma^2+\gamma^{-\frac{4N}{3}}\Theta((\sigma\Delta w_{\min})^{\frac{2}{3}})
\end{aligned}
$$

which completes the proof. □

F.3 Proof of Lemma 9: Descent of lower level sequence $W_{t_n}^{(n)}$

Lemma 9 (Descent lemma of lower level sequences $W^{(n)}$, long version of Lemma 3)**.** *Following the same assumptions of Lemma 8, for $n\in\{0,\dots,N-1\}$, the learning rate satisfies that $\beta\le\frac{F_{\max}^3\gamma}{3H_{\min}}$. Denote $\mathbb{E}_{\boldsymbol{\zeta_n}}:=\mathbb{E}_{\zeta_{t_n+kT_n:t_n+(k+1)T_n-1}}$. It holds that:*

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta_n}}[\|W_{t_n+(k+1)T_n-1}^{(n)}-P_n^*(\overline{W}_{t_{n-1}+k}^{(n)})\|^2] \qquad (59)\\
\le&\left(1-\frac{\beta H_{\min}}{2\gamma F_{\max}}\right)^{T_n}\|W_{t_n+kT_n}^{(n)}-P_n^*(\overline{W}_{t_{n-1}+k}^{(n)})\|^2\\
&+\frac{6F_{\max}^4\gamma^2}{H_{\min}^2}\|W_{t_{n+1}+(k+1)T_{n+1}-1}^{(n+1)}-P_{n+1}^*(\overline{W}_{t_n+k}^{(n+1)})\|^2+\frac{2F_{\max}\Theta(\Delta w_{\min})}{H_{\min}}\\
&-(\frac{2\gamma^2}{H_{\min}}-\frac{6\beta\gamma F_{\max}}{H_{\min}})\Big\|P_{n+1}^*(\overline{W}_{t_n+k}^{(n+1)})\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)-|P_{n+1}^*(\overline{W}_{t_n+k}^{(n+1)})|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big\|^2.
\end{aligned}
$$

Proof of Lemma 9. The proof begins from manipulating the norm $\|W_{t_n+1}^{(n)}-P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2$:

$$
\begin{aligned}
&\mathbb{E}_{\zeta_{t_n}}[\|W_{t_n+1}^{(n)}-P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2] \qquad (96)\\
=&\|W_{t_n}^{(n)}-P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2+2\mathbb{E}_{\zeta_{t_n}}[\langle W_{t_n}^n-P_n^*(\overline{W}_{t_{n-1}}^{(n)}),W_{t_n+1}^{(n)}-W_{t_n}^{(n)}\rangle]+\mathbb{E}_{\zeta_{t_n}}[\|W_{t_n+1}^{(n)}-W_{t_n}^{(n)}\|^2].
\end{aligned}
$$

Substituting update dynamic (58), we bound the second term of (96) as following:

$$
2\mathbb{E}_{\zeta_{t_n}}[\langle W_{t_n}^{(n)}-P_n^*(\overline{W}_{t_{n-1}}^{(n)}),W_{t_n+1}^{(n)}-W_{t_n}^{(n)}\rangle] \tag{97}
$$

$$
\begin{aligned}
&= 2\mathbb{E}_{\zeta_{t_n}}\Big[\Big\langle W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}), \beta\Big(W_{t_{n+1}+T-1}^{(n+1)} \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) + |W_{t_{n+1}+T-1}^{(n+1)}| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big) + \zeta_{t_n}\Big\rangle\Big] \\
&\le 2\beta\Big\langle W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}), P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big\rangle \\
&\quad + 2\beta\mathbb{E}_{\xi_{t:t+\mathcal{T}_n-1}}\Big[\Big\langle W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}), W_{t_{n+1}+T-1}^{(n+1)} \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |W_{t_{n+1}+T-1}^{(n+1)}| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) \\
&\quad - \Big(P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big\rangle\Big].
\end{aligned}
$$

The last inequality holds by Young's inequality. The first term in the RHS of (97) can be bounded by:

$$
\begin{aligned}
&2\beta\Big\langle W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}), P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big\rangle \\
&= 2\beta\Bigg\langle (W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}, \frac{P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}{\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}}\Bigg\rangle \\
&\overset{(a)}{=} -\frac{2\beta}{\gamma}\Bigg\langle (W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}, (W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}\Bigg\rangle \\
&\quad + \frac{2\beta}{\gamma}\Bigg\langle (W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}, |W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})| \odot \frac{G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}{\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}}\Bigg\rangle \\
&\overset{(b)}{=} -\frac{\beta}{\gamma}\Bigg\|(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}\Bigg\|^2 + \frac{\beta}{\gamma}\Bigg\||W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})| \odot \frac{G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}{\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}}\Bigg\|^2 \\
&\quad - \frac{\beta}{\gamma}\Bigg\|(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)} + |W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})| \odot \frac{G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}{\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}}\Bigg\|^2 \\
&\overset{(c)}{\le} -\frac{\beta}{\gamma F_{\max}}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2_{H(W_{t_n}^{(n)})} \\
&\quad - \frac{\beta}{\gamma}\Bigg\|(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)} + |W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})| \odot \frac{G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}{\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}}\Bigg\|^2 \\
&\overset{(d)}{\le} -\frac{\beta}{F_{\max}\gamma}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2_{H(W_{t_n}^{(n)})} \\
&\quad - \frac{\beta\gamma}{F_{\max}}\Bigg\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Bigg\|^2
\end{aligned} \tag{98}
$$

where (a) holds by $P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) = \frac{W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})}{\gamma}$, (b) leverages the equality $2\langle U, V\rangle = \|U\|^2 + \|V\|^2 - \|U-V\|^2$ for any $U, V \in \mathbb{R}^D$, (c) is the saturation vector $H(W_t)$ defined in (31). Thus:

$$
-\frac{\beta}{\gamma}\|(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})) \odot \sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}\|^2 + \frac{\beta}{\gamma}\Bigg\||W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})| \odot \frac{G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}{\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}}\Bigg\|^2
$$

$$
\begin{aligned}
&= -\frac{\beta}{\gamma}\sum_{d\in[D]}\left([(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}))]_d^2\left([F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d - \frac{[G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d^2}{[F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d}\right)\right)\\
&= -\frac{\beta}{\gamma}\sum_{d\in[D]}\left([(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}))]_d^2\left(\frac{[F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d^2 - [G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d^2}{[F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d}\right)\right)\\
&\le -\frac{\beta}{\gamma F_{\max}}\sum_{d\in[D]}\left([(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}))]_d^2\left([F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d^2 - [G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)]_d^2\right)\right)\\
&= -\frac{\beta}{\gamma F_{\max}}\|(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}))\|^2_{H(W_{t_n}^{(n)})}. \qquad (99)
\end{aligned}
$$

(d) comes from :

$$
\begin{aligned}
&-\frac{\beta}{\gamma}\Big\|(W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}))\odot\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)} + |W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})|\odot\frac{G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}{\sqrt{F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)}}\Big\|^2\\
=\ &-\beta\gamma\Big\|\Big(F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big)^{(-\frac{1}{2})}\odot\Big(\frac{W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})}{\gamma}\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\\
&+\Big|\frac{W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})}{\gamma}\Big|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big)\Big\|^2\\
\le\ &-\frac{\beta\gamma}{F_{\max}}\left\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\right\|^2. \qquad (100)
\end{aligned}
$$

The second term in the RHS of (97) is bounded by Lemma 5 as:

$$
\begin{aligned}
&2\mathbb{E}_{\zeta_{t_n}}\Big[\Big\langle W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}), W_{t_{n+1}+T-1}^{(n+1)}\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |W_{t_{n+1}+T-1}^{(n+1)}|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\\
&\quad - \Big(P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big)\Big\rangle\Big] \qquad (101)\\
\le\ &\frac{\beta}{2F_{\max}\gamma}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2_{H(W_{t_n}^{(n)})} + 2\beta F_{\max}\gamma\Big\|W_{t_{n+1}+T-1}^{(n+1)}\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |W_{t_{n+1}+T-1}^{(n+1)}|\\
&\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - \Big(P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big)\Big\|^2_{H(W_{t_n}^{(n)})^\dagger}\\
\le\ &\frac{\beta}{2F_{\max}\gamma}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2_{H(W_{t_n}^{(n)})} + 2\beta F_{\max}^3\gamma\|W_{t_{n+1}+T-1}^{(n+1)} - P^*(\overline{W}_{t_n}^{(n+1)})\|^2_{H(W_{t_n}^{(n)})^\dagger}.
\end{aligned}
$$

Plugging inequality (98) and (101) above into (97) yields:

$$
\begin{aligned}
&2\mathbb{E}_{\zeta_{t_n}}[\langle W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}), W_{t_n+1}^{(n)} - W_{t_n}^{(n)}\rangle] \qquad (102)\\
&\le 2\beta F_{\max}^3\gamma\|W_{t_{n+1}+T-1}^{(n+1)} - P^*(\overline{W}_{t_n}^{(n+1)})\|^2_{H(W_{t_n}^{(n)})^\dagger} - \frac{\beta\gamma}{F_{\max}}\Big\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\\
&\quad - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big\|^2 - \frac{\beta}{2\gamma F_{\max}}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2_{H(W_{t_n}^{(n)})}.
\end{aligned}
$$

We assume there exists a non-zero constant $H_{\min} > 0$ such that $\min\{H(W_{t_n}^{(n)})\} \ge H_{\min}$ for all t_n and n. Under this condition, we have the following inequalities:

$$
-\frac{\beta}{2F_{\max}\gamma}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2_{H(W_{t_n}^{(n)})} \le -\frac{\beta H_{\min}}{2F_{\max}\gamma}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2, \qquad (103)
$$

$$2\beta F_{\max}^3\gamma \left\|W_{t_{n+1}+T-1}^{(n+1)} - P^*(\overline{W}_{t_n}^{(n+1)})\right\|^2_{H(W_{t_n}^{(n)})^\dagger} \leq \frac{2\beta F_{\max}^3\gamma}{H_{\min}} \left\|W_{t_{n+1}+T-1}^{(n+1)} - P^*(\overline{W}_{t_n}^{(n+1)})\right\|^2.$$

Plugging inequality (103) above into (102) yields:

$$\begin{aligned}
&2\mathbb{E}_{\zeta_{t_n}}[\langle W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)}), W_{t_n+1}^{(n)} - W_{t_n}^{(n)}\rangle] \\
\leq& -\frac{\beta H_{\min}}{2\gamma F_{\max}}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2 + \frac{2\beta F_{\max}^3\gamma}{H_{\min}}\|W_{t_{n+1}+T-1}^{(n+1)} - P^*(\overline{W}_{t_n}^{(n+1)})\|^2 \\
&- \frac{\beta\gamma}{F_{\max}}\left\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\right\|^2.
\end{aligned} \tag{104}$$

The third term in the RHS of (96) is bounded by Lemma 5 as:

$$\begin{aligned}
&\mathbb{E}_{\zeta_{t_n}}[\|W_{t_n+1}^{(n)} - W_{t_n}^{(n)}\|^2] \\
=&\mathbb{E}_{\zeta_{t_n}}\left[\left\|\beta\Big(W_{t_{n+1}+T-1}^{(n+1)} \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |W_{t_{n+1}+T-1}^{(n+1)}| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big) + \zeta_{t_n}\right\|^2\right] \\
\leq& 3\beta^2\left\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\right\|^2 + 3\beta\Theta(\Delta w_{\min}) \\
&+ 3\beta^2\left\|W_{t_{n+1}+T-1}^{(n+1)} \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |W_{t_{n+1}+T-1}^{(n+1)}| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - \Big(P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\right. \\
&\left. - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big)\right\|^2 \\
\leq& 3\beta^2\left\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\right\|^2 + 3\beta\Theta(\Delta w_{\min}) \\
&+ 3\beta^2\mathbb{E}_{\xi_{t:t+\mathcal{T}_n-1}}[\|W_{t_{n+1}+T-1}^{(n+1)} - P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\|^2].
\end{aligned} \tag{105}$$

The second inequality holds by Cauchy-Schwarz inequality. Plugging inequality (104) and (105) above into (96) yields:

$$\begin{aligned}
&\mathbb{E}_{\zeta_{t_n}}[\|W_{t_n+1}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2] \\
=& (1 - \frac{\beta H_{\min}}{2\gamma F_{\max}})\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2 + \frac{2\beta F_{\max}^3\gamma}{H_{\min}}\|W_{t_{n+1}+T-1}^{(n+1)} - P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\|^2 \\
&- (\frac{\beta\gamma}{F_{\max}} - 3\beta^2)\left\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\right\|^2 \\
&+ 3\beta\Theta(\Delta w_{\min}) + 3\beta^2\|W_{t_{n+1}+T-1}^{(n+1)} - P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\|^2 \\
\leq& (1 - \frac{\beta H_{\min}}{2\gamma F_{\max}})\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2 + \frac{3\beta F_{\max}^3\gamma}{H_{\min}}\|W_{t_{n+1}+T-1}^{(n+1)} - P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\|^2 \\
&+ \beta\Theta(\Delta w_{\min}) - (\frac{\beta\gamma}{F_{\max}} - 3\beta^2)\left\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)}) \odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big) - |P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})| \odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\right\|^2.
\end{aligned} \tag{106}$$

The last inequality holds by setting $\beta \leq \frac{F_{\max}^3\gamma}{3H_{\min}}$. Executing T_n iterations through (106) yields:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta_n}}[\|W_{t_n+T_n-1}^n - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2] \\
\leq& \left(1 - \frac{\beta H_{\min}}{2\gamma F_{\max}}\right)^{T_n}\|W_{t_n}^{(n)} - P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2 + \sum_{i=0}^{T_n-1}\left(1 - \frac{\beta H_{\min}}{2\gamma F_{\max}}\right)^i(\beta\Theta(\Delta w_{\min}) \\
&+ \frac{3\beta F_{\max}^3\gamma}{H_{\min}}\|W_{t_{n+1}+(i+1)T_{n+1}-1}^{n+1} - P_{n+1}^*(\overline{W}_{t_n+i}^{(n+1)})\|^2)
\end{aligned} \tag{107}$$

$$
\begin{aligned}
&\leq \left(1-\frac{\beta H_{\min}}{2\gamma F_{\max}}\right)^{T_n}\|W_{t_n}^{(n)}-P_n^*(\overline{W}_{t_{n-1}}^{(n)})\|^2+\frac{6F_{\max}^4\gamma^2}{H_{\min}^2}\|W_{t_{n+1}+T_{n+1}-1}^{(n+1)}-P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\|^2\\
&\quad-(\frac{2\gamma^2}{H_{\min}}-\frac{6\beta\gamma F_{\max}}{H_{\min}})\Big\|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)-|P_{n+1}^*(\overline{W}_{t_n}^{(n+1)})|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big\|^2\\
&\quad+\frac{2F_{\max}\Theta(\Delta w_{\min})}{H_{\min}}.
\end{aligned}
$$

When $k\neq 0$, (107) can be written as the general case:

$$
\begin{aligned}
&\mathbb{E}_{\boldsymbol{\zeta_n}}[\|W_{t_n+(k+1)T_n-1}^{(n)}-P_n^*(\overline{W}_{t_{n-1}+k}^{(n)})\|^2] \qquad (108)\\
&\leq \left(1-\frac{\beta H_{\min}}{2\gamma F_{\max}}\right)^{T_n}\|W_{t_n+kT_n}^{(n)}-P_n^*(\overline{W}_{t_{n-1}+k}^{(n)})\|^2\\
&\quad+\frac{6F_{\max}^4\gamma^2}{H_{\min}^2}\|W_{t_{n+1}+(k+1)T_{n+1}-1}^{(n+1)}-P_{n+1}^*(\overline{W}_{t_n+k}^{(n+1)})\|^2+\frac{2F_{\max}\Theta(\Delta w_{\min})}{H_{\min}}\\
&\quad-(\frac{2\gamma^2}{H_{\min}}-\frac{6\beta\gamma F_{\max}}{H_{\min}})\Big\|P_{n+1}^*(\overline{W}_{t_n+k}^{(n+1)})\odot F\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)-|P_{n+1}^*(\overline{W}_{t_n+k}^{(n+1)})|\odot G\Big(\frac{W_{t_n}^{(n)}}{\kappa}\Big)\Big\|^2
\end{aligned}
$$

which completes the proof. □

G PSEUDO CODE

Algorithm 1 Multi-timescale Residual Learning with Warm Start Initialization

1: Initialize $W^{(0)}$ with the weight from digital side, working tiles $W^{(n)} \leftarrow 0$ for $n = 1, \ldots, N$
2: Initialize tile index counter: $t_n \leftarrow 0$ for all n, and set **current update tile** $k \leftarrow 0$
3: Initialize inner loop length: $T_n = \textit{Transfer\_every\_vec}[n]$
4: Initialize switching flag: `trigger_tile_switch` $\leftarrow$ `False`
5: Initialize loss history buffer: $\mathcal{L}$
6: **for** each iteration $t = 1, 2, \ldots$ **do**
7: $\quad W_{t_N}^{(N)} \leftarrow W_t^{(N)} - \alpha \nabla f(\overline{W}_t; \xi_t) \odot F\big(W_t^{(N)}\big) - \big|\alpha \nabla f(\overline{W}_t; \xi_t)\big| \odot G\big(W_t^{(N)}\big) + \zeta_t$
$\quad$ // Gradient accumulation
8: $\quad$ Append ℓ_t to loss history $\mathcal{L}$
9: $\quad$ **if** `LossPlateau`$(\mathcal{L}, k)$ **and** $k \geq 0$ **then**
10: $\quad\quad$ `trigger_tile_switch` $\leftarrow$ `True` // Detected plateau: trigger tile switch
11: $\quad$ **end if**
12: $\quad$ **if** `trigger_tile_switch` $=$ `True` **and** $k < N$ **then**
13: $\quad\quad k \leftarrow k + 1$ // Progressive per-tile transfer switch
14: $\quad\quad$ `trigger_tile_switch` $\leftarrow$ `False`
15: $\quad$ **end if**
16: $\quad$ **if** $k < N$ **and** $t \bmod T_N = 0$ **then**
17: $\quad\quad W_{t_k}^{(k)} \leftarrow W_{t_k}^{(k)} + \beta W_{t+T_N-1}^{(N)} \odot F\big(W_{t_k}^{(k)}\big) - \big|\beta W_{t+T_N-1}^{(N)}\big| \odot G\big(W_{t_k}^{(k)}\big) + \zeta_{t_k}$
$\quad\quad$ // Transfer update from $W^{(N)}$ to $W^{(k)}$
18: $\quad$ **end if** // Warm start initialization finished
19: $\quad$ **if** $k \geq N$ **then**
20: $\quad\quad$ **for** $n = N - 1$ **to** 0 **do**
21: $\quad\quad\quad$ **if** $t_{n+1} \bmod T_{n+1} = 0$ **then**
22: $\quad\quad\quad\quad W_{t_n}^{(n)} \leftarrow W_{t_n}^{(n)} + \beta \tilde{W}^{(n+1)} \odot F\big(W_{t_n}^{(n)}\big) - \big|\beta \tilde{W}^{(n+1)}\big| \odot G\big(W_{t_n}^{(n)}\big) + \zeta_{t_n}$
$\quad\quad\quad\quad$ // Transfer update from $W^{(n+1)}$ to $W^{(n)}$
23: $\quad\quad\quad$ **end if**
24: $\quad\quad$ **end for**
25: $\quad$ **end if**
26: $\quad \overline{W}_t = \sum_{n=0}^{N} \gamma^n W_{t_n}^{(n)}$ // Combine all tiles to form effective weight
27: **end for**

28: **Function:** LossPlateau$(\mathcal{L}, k)$
29: $\quad$ **if** $k \leq 3$:
30: $\quad\quad$ **if** $|\mathcal{L}| < 2$: **return** False // Not enough history
31: $\quad\quad$ **else**: **return** $\mathcal{L}[t] > \mathcal{L}[t-1]$ // Aggressive mode
32: $\quad$ **else**:
33: $\quad\quad$ **if** $|\mathcal{L}| < 6$: **return** False
34: $\quad\quad$ **else**:
35: $\quad\quad\quad v \leftarrow 0$
36: $\quad\quad\quad$ **for** $i = t-5$ to $t-1$ **do**
37: $\quad\quad\quad\quad$ **if** $\mathcal{L}[i+1] > \mathcal{L}[i]$: $v \leftarrow v + 1$
38: $\quad\quad\quad$ **end for**
39: $\quad\quad\quad$ **return** $v \geq 2$ // mild mode

Algorithm 1 includes an optional warm start phase (lines 1–18), which is only used in our experimental implementation to accelerate convergence and stabilize early training. This warm start is not required in the general method (Section 3) nor in the theoretical analysis (Section 4). The main results of our method should focus on the multiscale residual learning process beginning from line 19 onward. The warm start process in this algorithm uses the gradient accumulated on the tile $W^{(N)}$ to successively update tiles $W^{(0)}, W^{(1)}, \ldots, W^{(N-1)}$. Initially, only $W^{(0)}$ is initialized with the digital model weights (line 1), and the current update tile index is set to $k = 0$ (line 2). During training, $W^{(N)}$ is updated at every step using the gradient of the current composite weight $\overline{W}_t$ (line 6). Every T_N steps, the content of $W^{(N)}$ is transferred to tile $W^{(k)}$ (line 14). This continues until the loss plateaus, as determined by the `LossPlateau` function (lines 7–9). When a plateau is detected, the algorithm increments k (lines 10–13), thereby the content of $W^{(N)}$ is transferred to tile $W^{(k+1)}$. This procedure repeats until $k > N$, which means that all tiles have been updated, and then the warm start initialization is complete.

Algorithm	TT-v2	Analog SGD	MP	Ours
Digital storage [byte]	$\mathcal{O}(D^2+2D)$	$\mathcal{O}(2D)$	$\mathcal{O}(D^2+2DB)$	$\mathcal{O}(2D)$
Memory ops [bit]	$\mathcal{O}(16D/n_s)$	$\mathcal{O}(1)$	$\mathcal{O}(16D^2/B)$	$\mathcal{O}(1)$
FP ops	$\mathcal{O}(2D+2D/n_s)$	$\mathcal{O}(2D)$	$\mathcal{O}(2D^2+D)$	$\mathcal{O}(2D)$
Analog ops [time]	$(l_{\text{avg}}+\frac{1}{n_s})t_{\text{sp}}+\frac{t_{\text{M}}}{n_s}$	$l_{\text{avg}}t_{\text{sp}}$	$\frac{D}{B}t_{\text{sp}}$	$l_{\text{avg}}\frac{t_{\text{sp}}n_s}{n_s-1}+\frac{t_{\text{M}}}{n_s-1}$
$\approx$ Time est. [ns]	56.3	30.9	3024.5	95.9

Table 7: Comparison of complexity and estimated runtime for per-sample weight update. Here, D is the vector/matrix dimension and B is the mini-batch size. For the time estimates, we assume $D = 512$, $B = 100$, $n_s = 2$ as the transfer period, $l_{\text{avg}} = 5$ is the average number of pulses per sample, $t_{\text{sp}} = 5$ ns is the duration of a single pulse, $t_{\text{M}} = 40$ ns is the time for matrix-vector readout, and FP operations compute time is calculated assuming throughput of 0.7 TFLOPS (Jain et al., 2023). Statistics for TT-v2, Analog SGD, and MP are based on (Rasch et al., 2024).

H ANALOG CIRCUIT IMPLEMENTATION DETAILS

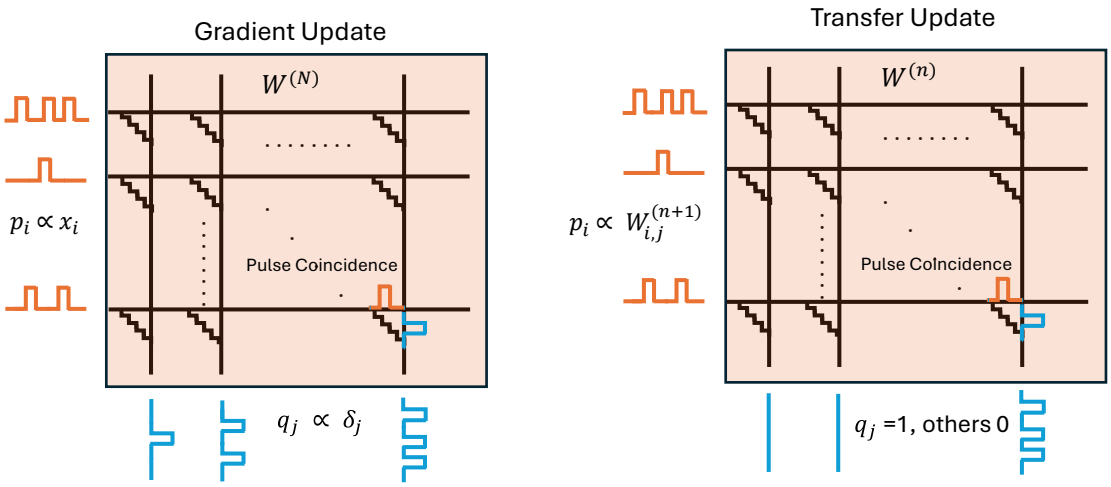


Figure 7: Implementation of gradient update and transfer update process.

H.1 Circuit-level implementation of update process

Figure 7 illustrates how gradient updates are applied to tile $W^{(N)}$ and how transfer updates are applied to the remaining tiles $W^{(n)}$. As described in Section 2, stochastic pulse streams whose probabilities are proportional to the error δ_j and input x_i are injected along each row and column of the crossbar array, realizing gradient updates with expectation $\Delta w_{ji} = \alpha\delta_j x_i = \frac{\partial f(W)}{\partial w_{ji}}$. For each transfer update on $W^{(n)}$, residual learning reads one column from $W^{(n+1)}$, and the column selection may follow either a cyclic or a sequential schedule. To support the feasibility of implementing the composite weight structure in Figure 3 (b), we note that similar analog crossbar-based weight composition and accumulation schemes have been demonstrated in practice (Song et al., 2024). These prior works focus on inference, where the weights remain static, but this does not affect our in-memory training scenario because the ADCs are only used during forward and backward passes, which are present in both training and inference. Moreover, sharing analog peripheral circuits (e.g., ADCs, DACs, and drivers) across multiple subarrays has been adopted in several recent compute-in-memory designs to reduce power and area overhead (Xu et al., 2024a,b).

H.2 Runtime and update complexity comparison.

As shown in Section 5 and supplement experiments in Section I, our algorithm achieves superior performance under a limited-precision setting, consistently outperforming the Tiki-Taka series and in some cases performing on par with MP. However, this performance advantage sometimes comes at the cost of using multiple analog tiles (from 3 to 8), in contrast to TT-v1, which employs only 2 tiles, TT-v2, which adds one digital tile to TT-v1, and MP, which uses a single analog tile combined with a high-precision digital unit. To evaluate the impact of using multiple tiles, we first summarize the per-sample weight update complexity of different algorithms in Table 7. This table provides a unified metric for comparing the hardware load across methods, including digital storage, memory operations, floating-point operations, and analog operations. Based on these results, we then present a more detailed analysis of the specific hardware costs including digital storage, runtime, energy, and area to highlight the efficiency advantages of our proposed approach.

Digital storage cost. We evaluate the digital storage cost of our proposed method in comparison with TT-v2, Analog SGD and MP. Here, digital storage refers exclusively to the memory used to buffer intermediate forward input, backward error, and gradients during training, which reside in SRAM or DRAM. In analog crossbar-based training, only the input $x \in \mathbb{R}^D$ and error signal $\delta \in \mathbb{R}^D$ are digitized via ADCs and temporarily stored for the backward pass. These two vectors lead to a digital storage requirement of $O(2D)$ bytes for both Analog SGD and our method. In contrast, TT-v2 requires an additional $D \times D$ digital transfer buffer between auxiliary and core arrays, incurring a total cost of $O(2D + D^2)$. For MP, gradient accumulation is performed in the digital domain over a batch of size B, resulting in $O(D^2 + 2DB)$ digital storage.

To quantify this storage in real scenarios, we extract the analog tile dimensions used in our experiments in Tables 1 and 3 and sum the corresponding vector and matrix sizes across all analog layers. Assuming each element occupies 1 byte (i.e., 8 bits of precision), we report total digital storage in kilobytes (KB). For MP, we incorporate mini-batch accumulation with batch size $B = 8$ for LeNet-5 and $B = 128$ for ResNet-18.

Model	**TT-v2**	**Analog SGD**	**MP**	**Ours**
LeNet-5 (KB)	80.2	2.13	94.8	2.13
ResNet-18 (KB)	10,600	50.2	17,000	50.2

Table 8: Digital storage required by different algorithms on LeNet-5 and ResNet-18.

Table 8 demonstrates that on LeNet-5 and ResNet-18 , our method achieves digital storage reductions of 37×–211× compared to TT-v2, and 44×–339× compared to MP while matching the minimal cost of Analog SGD. This efficiency stems from the fact that increasing the number of tiles does not incur additional cost for storing x and δ, as they are computed collectively across all tiles as illustrated in Figure 3(b). Moreover, unlike MP and TT-v2, our method eliminates the need for any digital tile to store weights or gradients. We believe our algorithm offers a substantial advantage in terms of digital memory overhead.

Runtime cost. We analyze the runtime cost, which includes both the *FP operations time* and *analog operations time*. For the MP baseline, the outer-product δx^T requires D^2 multiplications and D^2 additions, resulting in a total of $2D^2$ floating-point operations per input. Including additional scaling and preprocessing for x and δ, the total is $\mathcal{O}(2D^2 + D)$ FP ops. Dividing by the effective throughput of 0.175 TFLOPS (assuming 0.7 TFLOPS shared across 4 tiles as in (Rasch et al., 2024)), the FP operations take 2998.9ns, while the analog operations require $\frac{D}{B}t_{\text{sp}} = 25.6$ns, resulting in a total estimated latency of 3024.5ns. In contrast, our method maintains the same $\mathcal{O}(2D)$ FP operations as Analog SGD, as it only requires computing the absolute maximum values of x and δ to scale the probabilities used in stochastic pulse updates. For analog operations, we focus on deriving the runtime expression specific to our method, which includes the pulse update and the MVM-based readout for weight transfer. For pulse update, when the final tile $W^{(N)}$ is updated once, each preceding tile $W^{(n)}$ is updated approximately every $(1/n_s)^{N-n}$ iterations, and the aggregate latency is bounded by $l_{\text{avg}} \cdot t_{\text{sp}} \cdot \sum_{n=0}^{N}(1/n_s)^{N-n}$, whose upper bound as $N \to \infty$ converges to $\frac{l_{\text{avg}} t_{\text{sp}} n_s}{n_s - 1}$. Similarly, the MVM-based readout incurs an additional delay bounded by $\frac{t_{\text{M}}}{n_s - 1}$. As a result, our method's analog latency is $l_{\text{avg}} \cdot \frac{t_{\text{sp}} n_s}{n_s - 1} + \frac{t_{\text{M}}}{n_s - 1}$, which gives a total estimated latency of 95.9ns in Table 7.

We apply the same methodology to estimate runtime on full-model configurations. For both LeNet-5 and ResNet-

18, we assume each layer is processed in parallel, and the slowest layer dominates the total latency. For MP, the largest matrix in LeNet-5 is of size 128×512, resulting in a total latency of 457.4ns. In the case of ResNet-18, the largest analog matrix is 512×4608, leading to an FP time of 13508.0ns and an analog latency of 20.0ns (with batch size $B = 128$), yielding a total latency of 13528.0ns We summarize the runtime across all algorithms and architectures in Table 9 below:

Model	**TT-v2**	**Analog SGD**	**MP**	**Ours**
LeNet-5 (ns)	56.3	30.9	457.4	95.9
ResNet-18 (ns)	126.5	77.7	13528.0	142.7

Table 9: Estimated runtime for different analog training methods on LeNet-5 and ResNet-18.

These results highlight the efficiency of our method: on LeNet-5, MP is 4.8× slower than ours, while on ResNet-18, it is 94.7× slower. At the same time, our method introduces only modest latency overhead compared to TT-v2.

Energy cost. We analyze the energy consumption per training sample and compare our method to the MP baseline. As reported in (Le Gallo et al., 2018, Table 1), MP consumes 83.2nJ to process a single training image on a two-layer perceptron (785 inputs, 250 hidden neurons, 10 outputs; total 198,760 synapses). For consistency, we adopt the same model configuration for energy estimation across methods. For our method, the energy is composed of three parts: pulse update energy, transfer update energy, and forward/backward propagation energy. To estimate the pulse update energy , we follow (Gokmen and Vlasov, 2016), which reports a combined power consumption of 0.7W for op-amps and stochastic translators (STRs) on a 4096×4096 crossbar. We scale this power by the number of active rows and columns in our perceptron, yielding $P_{\text{scaled}} = 0.1107\text{W}$. Assuming a worst-case pulse update time of $\frac{l_{\text{avg}} t_{\text{sp}} n_s}{n_s - 1} = 50\text{ns}$, we obtain the pulse update energy: $E_{\text{pulse\_update}} = P_{\text{scaled}} \cdot 50\text{ns} \approx 5.53\text{nJ}$. In addition, we consider the transfer update energy for analog readout. Since in our architecture each tile is read roughly every n_s steps, the aggregate readout time is upper bounded by the MVM latency of a single tile, $\frac{t_{\text{M}}}{n_s - 1} = 40\text{ns}$. The MP work reports a forward-propagation energy in computational memory of 7.29nJ under similar conditions, which can be interpreted as the estimated energy cost for a single-tile analog forward pass closely matching the MVM read time in our design. Therefore, we adopt 7.29nJ as an upper-bound estimate for our total transfer energy: $E_{\text{trans}} = 7.29\text{nJ}$. Hence, the total update energy per sample in our method is the sum:

$$E_{\text{update}} = E_{\text{pulse\_update}} + E_{\text{trans}} \approx 12.82\text{nJ}.$$

For the forward and backward passes, our method distributes the computation across N analog tiles. Each tile requires the full sequence of operations including data input, PWM generation, read-voltage regulation, analog computation, ADC conversion and data output. We adopt a conservative estimation by assuming that these operations are not shared across tiles, so the energy cost of each tile is independent and must be incurred individually. Following the values reported in (Le Gallo et al., 2018), the forward and backward passes consume 7.29nJ and 2.15nJ per tile, respectively. Consequently, the total propagation energy scales linearly with N as $N \cdot (7.29 + 2.15) = N \cdot 9.44\text{nJ}$. Putting all the components together, we compare the energy consumption per training sample in Table 10:

Component	**MP (nJ)**	**Ours (nJ)**
Weight update	62.03	12.82
Forward/backward pass	21.21	$N \cdot 9.44$
Total	83.2	$12.82 + N \cdot 9.44$

Table 10: Estimated energy consumption per image for MP and our method based on (Gokmen and Vlasov, 2016; Le Gallo et al., 2018).

As the table shows, our method becomes less energy-efficient than MP when $N \geq 8$. In practice, however, much of the forward/backward overhead can be shared across tiles. For example, the PWM counter and comparator logic that generate input vectors into modulated pulses, and the ADCs that digitize accumulated outputs can all be amortized across multiple tiles rather than duplicated. Such shared-ADC designs have been demonstrated in recent analog accelerators (Xu et al., 2024a,b; Song et al., 2024). Only the per-tile operational transconductance

amplifiers used for voltage regulation, along with the intrinsic device conduction energy, scale directly with N. This means that the effective energy growth with tile count is substantially slower than the conservative upper bound assumed in Table 10. Combined with the fact that our method achieves higher accuracy than TT baselines with as few as 3–4 tiles, these considerations indicate that our design remains more energy-efficient than MP even when more than eight tiles are employed, making it a practical and scalable solution.

Area cost. For our methods, since the overall architecture closely resembles that of TT-v1 except for the increased number of tiles, the corresponding estimates for area and execution time are derived based on models presented in (Gokmen and Haensch, 2020) and (Gokmen and Vlasov, 2016). We analyze the area overhead of our method using the RPU tile design methodology described in (Gokmen and Vlasov, 2016), which assumes a realistic CMOS-compatible fabrication stack. Specifically, RPU arrays are implemented in the back-end-of-line (BEOL) region, with resistive memory devices placed between intermediate metal layers. Each crossbar array contains $D \times D$ devices, and the interconnect pitch (wire width plus spacing) is set to 400 nm , based on typical dimensions of intermediate BEOL levels. This yields a physical tile area of $((0.4D)\mu\text{m})^2 = (0.16D^2)\mu\text{m}^2$. Following (Gokmen and Vlasov, 2016), we adopt a baseline configuration of $D = 4096$, which corresponds to a tile area of approximately 2.68mm^2. In our experimental configurations (e.g., LeNet-5 and ResNet-18), which use smaller crossbar sizes, the tile area is scaled proportionally under the same pitch assumption; for instance, a 128×512 tile occupies 0.0105mm^2 . Based on the actual tile dimensions used in each layer of Tables 1 and 3, we estimate the total analog area required by our method to be 0.0128mm^2 for LeNet-5 and 1.69mm^2 for ResNet-18.

To map logical weights to physical devices, each weight W is represented as the difference between two conductance values: a main conductance C_{main} and a reference C_{ref}, i.e., $W \propto C_{\text{main}} - C_{\text{ref}}$. This implies that both Analog SGD and MP require 2 analog tiles per layer . TT-v1 and TT-v2, which maintain both core and assistant matrices, require 4 analog tiles per layer . In our method, the number of physical analog tiles is twice the count reported in Tables 1 and 3, due to our multi-tile residual structure. However, because these tiles can be vertically stacked using BEOL integration, the actual die area remains compact. Even without stacking, the total area overhead of our method remains within practical limits. For example, in the worst-case configuration using 10 analog tiles, our method incurs approximately 10× the area of MP and 5× that of TT-v2. Yet this level of overhead is still feasible: modern processors such as IBM Power8 CPUs (Stuecheli et al., 2013) supports chip areas up to 600mm^2. Such systems are capable of integrating hundreds of analog tiles, indicating that our method remains scalable and realistic under practical hardware constraints.

In summary, our multi-tile framework provides clear advantages over TT-v2 and MP: it reduces digital storage by up to two orders of magnitude and achieves substantially lower runtime latency, benefiting from parallel analog updates even when more tiles are used. While area and energy scale with tile count, area overhead can be mitigated through BEOL stacking, and energy only exceeds MP under *the most conservative estimates* when $N > 8$, since those estimates assume that all I/O, PWM logic, and ADC resources are replicated per tile rather than shared, meaning that in practical our design can tolerate substantially more tiles before its energy surpasses MP. Importantly, our method consistently delivers higher accuracy than TT baselines with only 3–4 tiles, keeping energy well within practical limits. These results establish our approach as a scalable and efficient solution for high-precision analog training.

I SUPPLEMENT SIMULATIONS

To further demonstrate the scalability, robustness, and practical relevance of our proposed method, we conduct a series of supplementary experiments across diverse settings, as supplement simulations to Section 5. Lastly, we extend our method to a Transformer-based natural language modeling task to demonstrate its applicability beyond vision workloads. Together, these results strengthen the case for our method as a scalable and general solution for analog training under various model architectures and hardware regimes.

I.1 CIFAR-100 experiments

To further validate the effectiveness of our method on more challenging datasets, we conduct additional experiments on CIFAR-100 using ResNet-18, with devices limited to 4 conductance states. Given the increased

Method	MP	TT-v1	TT-v2	Ours (3 tiles)	Ours (4 tiles)
Loss	2.43	2.75	2.62	2.49	2.47

Table 12: Test loss on 2-layer LSTM.

complexity and number of classes in CIFAR-100, we extend the training schedule to 400 epochs to ensure sufficient convergence.

Model	TT-v1	TT-v2	MP	Ours (4 tiles)	Ours (6 tiles)	Ours (8 tiles)
ResNet-18	$14.97_{\pm 1.93}$	$27.91_{\pm 0.65}$	$64.08_{\pm 0.44}$	$58.36_{\pm 0.36}$	$59.68_{\pm 0.33}$	$60.62_{\pm 0.24}$

Table 11: Test accuracy on CIFAR-100 using 4-state analog devices.

The results summarized in Table 11 demonstrate that our method consistently outperforms TT-v1 and TT-v2 baselines and approaches the performance of the MP method as $6-8$ tiles are used.

I.2 Extension to LSTM-based NLP tasks

We implemented a character-level next-character prediction experiment on the War and Peace dataset using a two-layer LSTM. The dataset is split into a training set (about 2.9M characters) and a test set (about 0.16 M characters). We feed sequences of length 100 into a 2-layer LSTM with hidden size 64, followed by a linear layer that maps the 64-dimensional hidden state at each time step to a 101-dimensional output vector representing the predicted next character. We train this network with cross-entropy loss to predict, at each position in the sequence, the next character in the text.

Same as in the CNN experiments, we simulate SoftBounds devices with 10 conductance states. We set the global learning rate to $\text{lr} = 0.01$ for all algorithms. For TT-v1 and TT-v2, we use $\text{fast\_lr} = 0.1$, and for TT-v2 we additionally set $\text{transfer\_lr} = 1$. For our residual-learning scheme, we set $\gamma = 0.1$ and $\text{transfer\_lr} = 0.1 * (1.2)^n$ for all tiles. Under this common setup, we obtain the following test cross-entropy losses after 100 epochs:

I.3 Extension to Transformer-based NLP tasks

To further demonstrate the scalability and general applicability of our method, we conducted an additional experiment on a natural language processing task using a GPT-2-style Transformer architecture. Specifically, we trained a 6-layer, 6-head, 768-dimensional model from scratch on the standard *Shakespeare character-level language modeling* benchmark. The total number of trainable parameters is approximately 10.65M, and each iteration processes 16384 tokens. We ran the training for 5000 iterations using 4-tile analog devices under non-ideal I/O conditions to simulate realistic hardware noise. The analog device used has 4 discrete states. We compare our method against representative analog training baselines, including TT-v1, TT-v2 and MP.

Method	TT-v1	TT-v2	MP	Ours (4 tiles)
Loss	3.0336	2.6137	2.7213	2.5971

Table 13: Validation loss on 6-layer GPT-style model.

As shown in Table 13, our method achieves comparable final validation loss, demonstrating both accuracy and robustness on this standard NLP benchmark. These results confirm that our method is not limited to vision tasks, but also scales effectively to Transformer-based sequence modeling, maintaining accuracy and resilience under analog non-idealities.

J SIMULATION DETAILS

This section provides details about the experiments in Section 5. The analog training algorithms, including Mixed Precision and Tiki-Taka, are provided by the open-source simulation toolkit AIHWKIT, which has an Apache-2.0 license; see `https://github.com/IBM/aihwkit`. We use the Softbound device provided by AIHWKIT to simulate the asymmetric linear device. Digital algorithms and datasets used in this paper are provided by

PyTorch, which has a BSD license; see `https://github.com/pytorch/pytorch`. Our implementation builds upon the `TT-v1` preset in `AIHWKIT v0.9.2`, with modifications to the gradient routing and transfer mechanisms to support our proposed Residual Learning scheme. We conduct our experiments on an NVIDIA RTX 3090 GPU, which has 24GB memory. The simulation time ranges from one to three hours, depending on the model size, dataset, and the number of training epochs. The code is available at `https://github.com/Jindanli898/AIMC`. The simulations reported are repeated three times. The randomness originates from the data shuffling, random initialization, and random noise in the analog device simulator. The mean and standard deviation are calculated using the **statistics** library.

J.1 MNIST and Fashion-MNIST training with analog LeNet-5

Data and preprocessing. The MNIST dataset is used with standard normalization and no data augmentation. The training and testing sets use the default PyTorch torchvision splits (60,000 training and 10,000 testing samples). In our experiments, we utilize the full training set with a batch size of 8. The Fashion-MNIST dataset is used with the default PyTorch `torchvision` splits, consisting of 60,000 training and 10,000 testing samples. No additional data augmentation is applied. A simple normalization transform is used through `ToTensor()`, and the full training set is utilized with a batch size of 16.

Model architecture. We adopt a LeNet-5 model in which all convolutional and fully-connected layers are implemented using AIHWKit's analog modules (`AnalogConv2d`, `AnalogLinear`). Digital non-linear operations, such as Tanh activations and MaxPooling, are interleaved and remain executed in the digital domain.

Optimizer and learning rate. For MNIST, we employ the `AnalogSGD` optimizer with an initial global learning rate of 0.05 applied uniformly to all trainable parameters. For Fashion-MNIST, we use the `AnalogSGD` optimizer with an initial global learning rate of 0.2 for our method, and 0.1 for TT-v1, TT-v2, and MP. A learning rate scheduler based on `LambdaLR` decays this global rate by a factor of 0.5 every 30 epochs. In analog layers, for our algorithm, each tile $W^{(n)}$ is assigned a fixed internal learning rate `transfer_lr_vec[n]`$= 0.1 \cdot 1.2^n$. These internal learning rates remain constant throughout the training and are not affected by the global schedule, as `scale_transfer_lr=False`. For TT-v1, we set the auxiliary tile learning rate `fast_lr` α to 0.01 and the transfer learning rate `transfer_lr` to 0.1 on both datasets. For TT-v2 we set $\alpha = 0.1$ and $\beta = 1$ for MNIST and $\alpha = 0.05$ for Fashion-MNIST. Additionally, we set `scale_transfer_lr=True` for TT-v1, TT-v2 and MP as default.

Tile parameter configuration. We configure the behavior of each analog tile through the following parameter vectors, all generated dynamically as a function of the total number of tiles `num_tile`:

- For MNIST:
  - `transfer_every_vec = [2 * (5^n) for n in range(num_tile)]`
  - `gamma_vec = [0.5^(num_tile - 1 - i) for i in range(num_tile)]`
- For Fashion-MNIST:
  - `transfer_every_vec = [2 * (5^n) for n in range(num_tile)]`
  - `gamma_vec = [0.2^(num_tile - 1 - i) for i in range(num_tile)]`

These vectors control the per-tile transfer schedule, readout scaling, and learning rate, respectively. The number of tiles `num_tile` is a configurable parameter that we vary in experiments (e.g., Table 1). It is worth noting that in the actual implementation, tile index `0` serves as the fixed gradient accumulation tile and plays the role of $W^{(N)}$ in the main text. The remaining tiles at indices $1, 2, \ldots,$ `num_tile` $- 1$ correspond to $W^{(N-1)}, W^{(N-2)}, \ldots, W^{(0)}$, respectively. While this index order is opposite to the mathematical notation used in the main text, the transfer logic and learning behavior are equivalent. The index inversion only affects naming, not the functional correctness or conclusions of the training algorithm.

I/O configuration. As acknowledged in Section 6, this work assumes idealized I/O settings throughout all experiments. I/O behavior is configured as nearly ideal, with:

- **Forward path:** The input vector x is injected into the crossbar without finite resolution quantization, amplitude clipping, or additive noise. The resulting output current is integrated ideally, bypassing any ADC or nonlinear feedback models.
- **Backward path:** The backpropagated vector δ is encoded and applied in a similarly ideal manner, ignoring input resolution limits, digital-to-analog conversion noise, or output quantization during the gradient computation.
- **Transfer path:** When using compound devices such as `TransferCompound`, the internal transfer of weights between tiles (e.g., during warm start or periodic updates) also involves analog readout and write operations. Setting `device.transfer_forward.is_perfect = True` disables all I/O imperfections during this internal read phase, ensuring clean accumulation and precise programming of weights across tiles.

Tile switching schedule. To avoid early saturation of coarse tiles, the training monitors convergence plateaus via loss history. Early epochs use an aggressive trigger if training loss does not drop sufficiently between epochs. After four tile switches, a smoother criterion is used based on the recent 5-step moving window. Upon plateau detection, a C++ flag `trigger_tile_switch` is activated via Python binding for each tile.

Training and Evaluation. The network is trained for up to 100 epochs. Classification loss is computed using `nn.NLLLoss()` applied to the log-softmax outputs. Evaluation is performed after each epoch using classification accuracy on the full MNIST test set.

J.2 CIFAR-10 and CIFAR-100 training with ResNet

Dataset and augmentation. The CIFAR-10 dataset is used for all ResNet experiments. Following the default splits provided by `torchvision`, the dataset consists of 50,000 training samples and 10,000 test samples, selected via the `train=True/False` flag. For additional experiments on more fine-grained recognition, we also use the CIFAR-100 dataset, which has the same image size and train/test splits but with 100 object categories grouped into 20 superclasses. We utilize the entire training set and set the batch size to 128. All images are normalized to zero mean and unit variance per channel. During training, strong data augmentation is applied, including random cropping, horizontal flipping, Cutout regularization and AutoAugment using 25 CIFAR-10-specific sub-policies. No augmentation is applied to the test set beyond normalization.

Model architecture. In different experiments, we use ResNet-18 and ResNet-34 models, where `layer3`, `layer4`, and the final classifier are mapped to analog tiles, while the remaining layers remain digital. Batch normalization and residual shortcuts are preserved unless explicitly disabled.

Tile parameter configuration. We configure the behavior of each analog tile through the following parameter vectors, all generated dynamically as a function of the total number of tiles `num_tile`:

- For 4-state experiments:
  - `transfer_every_vec = [3 * (2^n) for n in range(num_tile)]`
  - `gamma_vec = [0.5^(num_tile - 1 - i) for i in range(num_tile)]`
- For 16-state experiments:
  - `transfer_every_vec = [3 * (2^n) for n in range(num_tile)]`
  - `gamma_vec = [0.1^(num_tile - 1 - i) for i in range(num_tile)]`

I/O configuration. The I/O configuration is the same as in the MNIST experiments.

Optimizer and Scheduler. All analog parameters are trained using `AnalogSGD` with an initial learning rate of 0.1. A `StepLR` scheduler reduces the learning rate by a factor of 0.1 every 100 epochs. In analog layers, for our algorithm, each tile $W^{(n)}$ is assigned a fixed internal learning rate `transfer_lr_vec[n]`$= 0.3 \cdot 1.2^n$. These internal learning rates remain constant throughout the training and are not affected by the global schedule, as `scale_transfer_lr=False`. For both TT-v1 and TT-v2, we set the auxiliary tile learning rate `fast_lr` α to

0.1 and the transfer learning rate `transfer_lr` to 1. Additionally, we set `scale_transfer_lr=True` for TT-v1, TT-v2 and MP as default.

Training and Evaluation. The network is trained for 200 epochs for CIFAR-10 and 400 epochs for CIFAR-100 with a batch size of 128. Classification loss is computed using label-smoothed cross-entropy, implemented via `LabelSmoothingLoss` with a smoothing factor of 0.1.

Tile switching. The tile switching strategy is the same as in the MNIST experiments.

J.3 Least square problem

Model architecture. We use a scalar analog layer $\mathbb{R}^1 \to \mathbb{R}^1$: `AnalogLinear(1,1)`.

Tile parameter configuration. We instantiate a `TransferCompound` device with `num_tile` identical unit cells, each a `SoftBoundsDevice` with $w_{\min} = -1$, $w_{\max} = 1$, and $\Delta w_{\min} = 0.5$. Column transfers are enabled and multi-sample updates are treated as mini-batch units (`transfer_columns=True`, `units_in_mbatch=True`). The two key parameter vectors are generated from `num_tile`:

- `transfer_every_vec = [2 * (2^n) for n in range(num_tile)]`
- `gamma_vec = [0.1^(num_tile - 1 - i) for i in range(num_tile)]`

Tile-internal transfer learning rate is fixed to `transfer_lr`=0.01 with `scale_transfer_lr=False`. We also set the pulse update scheme as `update_bl_management=False`, `update_management=False`, as well as weight scaling scheme `digital_bias=False`, `learn_out_scaling=False`, `weight_scaling_columnwise=False`, and `weight_scaling_omega`=0.0.

I/O configuration. The I/O configuration is the same as in the MNIST experiments.

Target generation. Batch size defaults to `batch_size`=1. The regression target $b \in [-1, 1]$ is sampled from a uniform 16-bit quantizer:

$$b = -1 + k \cdot \frac{2}{2^{16} - 1}, \qquad k \sim \text{Uniform}\{0, \ldots, 2^{16} - 1\}.$$

Optimizer. All analog parameters are trained using `AnalogSGD` with an initial learning rate of 0.001. In analog layers, for our algorithm, each tile $W^{(n)}$ is assigned a fixed internal learning rate `transfer_lr`= 0.01. These internal learning rates remain constant throughout the training and are not affected by the global schedule, as `scale_transfer_lr=False`. We set the auxiliary tile learning rate `fast_lr` α to 0.01.