CT-QLC: A Production-Ready Firmware Solution for Tail Latency QoS in Charge-Trap QLC SSDs

Yu Chen*, Maojun Yuan*, Huguang Chen*, Jianxiong Zhao*, Jie Yang*, Zhiyuan Leng*,
Zhigang Liu*, Caiqiang Chen*, Xiaobing Wang*, Feng Zhu*, and Shu Li*
*Alibaba Cloud Computing
{zhaoxi.cy, yuanmaojun.ymj, chenhuguang.chg, zhaojianxiong.zjx, daizhou.yj, lengzhiyuan.lzy}@alibaba-inc.com
{liuzhigang.lzg, caiqiang.ccq, xiaoqian.wxb, f.zhu, s.li}@alibaba-inc.com

***Abstract*—Quad-Level Cell (QLC) SSDs offer compelling cost-capacity advantages for read-intensive data-center workloads. However, charge-trap QLC SSDs face severe early data retention challenges: threshold voltage drift necessitates read retries when using stale reference voltages, significantly increasing tail latency.**

This paper presents CT-QLC, a production-ready firmware solution that addresses this problem by maintaining accurate read voltages through hierarchical table management. CT-QLC introduces a production-integrated three-tier table structure (factory, active, and staging) that enables progressive voltage refinement while maintaining system stability. The design separates voltage verification from refinement: periodic verification detects voltage drift, while conditional refinement applies either fail ratio-based reordering or adaptive window-based valley tracking to restore accuracy. This hierarchical approach achieves adaptive voltage tracking with bounded background overhead, enabling deployment on commercial SSDs without hardware modifications. Evaluation demonstrates up to 94% reduction in 99.99th percentile latency at high temperature (55°C) compared with unoptimized charge-trap QLC SSDs, and up to 89% reduction compared with floating-gate QLC SSDs.

***Index Terms*—QLC SSD, Charge-Trap Flash, Quality of Service, Read-Voltage Tuning, Data Center Storage**

I. Introduction

High-density solid-state drives (SSDs), such as quad-level cell (QLC) drives, are increasingly favored for data-center storage due to their superior performance and lower power consumption compared with hard disk drives (HDDs) [1], and are particularly well-suited for high-capacity, performance-demanding scenarios in artificial intelligence (AI) and big data applications [2]. Forward Insights predicts that QLC's market share will grow to 35% by 2026 [3], with leading NAND flash and SSD manufacturers already engaging in mass production of QLC-based products [4]–[6]. Empirical studies have demonstrated the viability of QLC SSDs for big data applications [7], and cost analyses confirm their economic advantages for data-center storage [8]. QLC SSDs are therefore becoming a natural choice for data-center deployment [9], especially for read-intensive services.

As data volumes and application diversity grow, read-heavy workloads with high-capacity and high-performance requirements have become commonplace. The rapid development of large language models (LLMs) further amplifies this trend, since model training and especially model inference processes involve extremely frequent read operations. Despite the write performance gap, the read performance of QLC NAND is relatively close to that of triple-level cell (TLC) NAND but at a much lower cost, making QLC SSDs attractive for such read-intensive scenarios.

However, QLC technology faces greater data reliability challenges than TLC NAND does. First, packing 16 threshold-voltage states into the same voltage window severely compresses per-state margins, amplifying errors from voltage shifts. Second, QLC cells exhibit accelerated charge loss and threshold-voltage drift over time, further exacerbating retention-induced errors.

Specifically for data-center QLC SSDs, current mainstream products such as Solidigm P5316 [10] and P5336 [11] rely exclusively on floating-gate flash technology. While floating-gate flash offers relatively strong data retention, it faces increasing fabrication challenges with aggressive 3D stacking and is supported by a relatively closed ecosystem. In contrast, charge-trap flash provides better scalability and cost advantages, positioning it as a promising foundation for next-generation QLC SSDs.

Unfortunately, the charge-trap structure makes electrons more susceptible to de-trapping in the short term, leading to significantly faster charge loss. In 3D charge-trap NAND, substantial voltage shifts may occur within hours after programming, resulting in *early data retention failures* that directly degrade read performance and QoS.

Prior work [12]–[16] has demonstrated that read-voltage optimization is an effective mitigation strategy. By dynamically adjusting read reference voltages to track threshold-voltage drift, it reduces misclassification errors and avoids expensive read retries.

Existing read-voltage optimization techniques broadly fall into three categories [17]–[21]:

1) **Offline table-based schemes** derive read-voltage tables from device characterization but cannot fully adapt to runtime variations such as temperature changes or non-uniform aging.
2) **Online adaptive schemes** iteratively tune voltages using runtime error feedback but may introduce non-negligible firmware or central processing unit (CPU) overhead, affecting tail latency.
3) **Hardware-assisted schemes** leverage additional on-chip capabilities to accelerate voltage search but require

controller or NAND support that limits their applicability in existing products.

These limitations make it challenging to deploy prior solutions directly in performance-critical data-center environments. To address these challenges, we present CT-QLC, a production-ready firmware solution that implements a three-tier voltage table-based optimization framework to obtain near-optimal read voltages and avoid read retries. The key contributions are:

- A **production-integrated three-tier voltage table hierarchy** (factory, active, staging) that enables progressive refinement while maintaining system stability, eliminating the stability-overhead trade-off faced by prior approaches.
- A **production-ready firmware-only solution** that requires no hardware modifications, making it immediately deployable in existing commercial SSDs while maintaining bounded background overhead.
- **Extensive experimental validation** on commercial SSD hardware demonstrating up to 94% reduction in 99.99th percentile latency versus unoptimized charge-trap QLC SSDs at high temperature, and up to 89% reduction in 99.99th percentile latency versus floating-gate QLC SSDs with twice the channel count.

The rest of this paper is organized as follows. Sections II and III present the background and design of CT-QLC. Section IV evaluates the performance of CT-QLC. Section V discusses related work, and Section VI concludes the paper.

II. Background and Motivation

In this section, we review QLC NAND flash technology, discuss reliability challenges across different manufacturing technologies, and explain why a firmware-level read-voltage optimization framework is needed for charge-trap QLC SSDs in data centers.

A. QLC NAND and Charge-Trap Technology

In NAND flash, each cell stores data by trapping electrical charge at specific voltage levels; different charge levels represent different bit patterns. Quad-Level Cell (QLC) NAND flash stores four bits per cell using 16 distinct threshold-voltage states (representing 4 bits: 0000 to 1111), compared with Triple-Level Cell (TLC)'s 8 states (representing 3 bits: 000 to 111).

QLC Technology Advantages for Data-Center Workloads. At comparable form factors, QLC's higher bit density delivers approximately 33% higher raw capacity than that of TLC, enabling cost-effective scaling for massive datasets. This density advantage aligns naturally with read-intensive data-center workloads such as AI model serving, content delivery, and large-scale analytics, where the inherent trade-off of lower write endurance is well tolerated due to minimal write amplification.

QLC Technology Challenges. Packing twice as many states into a similar voltage window compresses the per-state voltage margin from approximately 0.5 V per state in TLC to approximately 0.25 V per state in QLC. This reduced margin makes cells more susceptible to: (1) wear-out from repeated program/erase cycles; (2) read disturb during frequent accesses; (3) cell-to-cell interference during programming; and (4) charge leakage over time and temperature, which degrades data retention and causes threshold voltage drift that readily crosses the narrowed state boundaries. Consequently, meeting strict service-level agreements (SLAs) in production deployments requires robust QoS mechanisms, including adaptive read threshold tracking and latency-aware I/O scheduling.

There are two main manufacturing technologies for NAND flash: floating-gate [22] and charge-trap [23], [24]. Comparative analyses of these technologies reveal fundamental trade-offs between retention characteristics and scalability [25]. In floating-gate technology, the storage element is a conductive polysilicon layer completely surrounded by oxide insulation; charge loss occurs primarily through slow tunneling through the oxide, resulting in good retention characteristics. In charge-trap technology, the storage element is silicon nitride, a non-conductive material that traps electrons at localized sites within the nitride layer.

Physical Properties of Charge-Trap Flash. Charge-trap technology enables higher 3D stacking (currently 200+ layers) compared with floating-gate (approximately 100–176 layers in recent generations) [26], directly translating to higher capacity per die and lower cost. However, the physical structure of charge-trap cells makes them more susceptible to charge loss: electrons trapped at discrete sites can be released (de-trapped) more easily, especially shortly after programming, leading to faster charge loss particularly in the first 24–72 hours. While this makes charge-trap flash more susceptible to early retention issues, the localized charge trapping reduces cell-to-cell interference and enables more aggressive 3D stacking, improving scalability and cost-effectiveness [27]. This creates a fundamental tension: the technology with better cost scaling (charge-trap) has worse early retention characteristics.

As shown in Figure 1, V_0 denotes the default read voltage immediately after programming. As charge loss accumulates, the optimal read voltage shifts to V_1; continuing to read with V_0 will increase bit errors and may cause read failures.

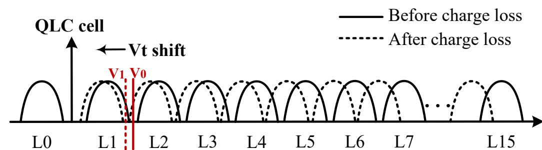


Fig. 1. Read voltage shifting due to data retention

B. SSD Architecture and Read Retry Mechanism

An SSD primarily consists of three key components: an SSD controller that handles all computations, runtime memory (typically DRAM) for temporary data storage, and multiple NAND flash memory chips for permanent data storage. The host communicates with the SSD through high-speed interfaces such as Non-Volatile Memory Express (NVMe) [28], while the controller connects to each NAND chip via dedicated channels [29].

The controller runs specialized firmware on embedded processors to orchestrate critical operations. Beyond managing host communications and I/O scheduling, the firmware implements data reliability mechanisms such as low-density parity-check (LDPC) error correction. By adding carefully designed parity bits to stored data, the firmware can detect and correct bit errors caused by NAND voltage drift within the error correction capability of the LDPC code.

When the read voltage is set improperly, the controller must issue read retries, repeatedly adjusting reference voltages and re-reading the same data, which directly increases read latency. As illustrated in Figure 2, reading data with an inaccurate read voltage requires multiple retries before success, whereas reading with an accurate read voltage requires no retry.

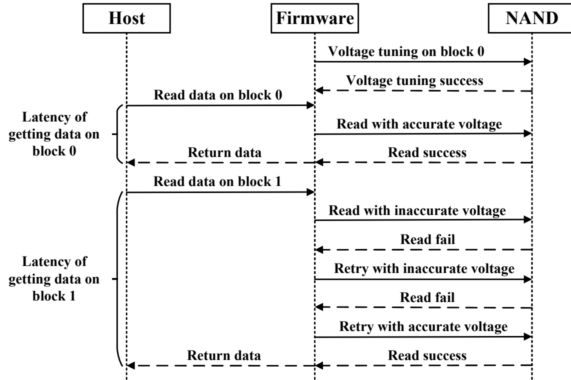


Fig. 2. Impact of read-voltage accuracy on read latency

Each retry adds substantial latency, directly impacting tail latency. CT-QLC aims to make the first read attempt succeed by maintaining accurate read voltage entries, avoiding the read retry cascade.

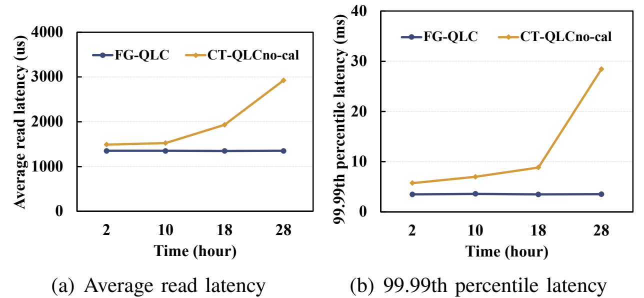
C. Data-Center QLC SSDs: Early Retention and Tail-Latency Challenges

Data-center SSDs serve latency-sensitive services in large-scale distributed systems and have markedly different requirements from client SSDs. While client SSDs primarily target cost-effectiveness and adequate responsiveness for a single host, data-center SSDs must sustain consistent performance under continuous high-I/O workloads. Operators typically specify QoS requirements in terms of high-percentile latencies (e.g., P99 or higher) rather than just averages [30]. For such environments, occasional long read retries caused by retention-induced voltage drift can violate service-level agreements (SLAs), making tail latency control a first-class design goal.

The Early Retention Problem. 3D charge-trap flash exhibits particularly pronounced *early* retention loss: substantial charge leakage can occur even after a short retention time (hours to days) following programming [17], [31]. This phenomenon differs from traditional long-term retention loss that occurs over months or years [32], [33]. The root cause lies in the physical structure of charge-trap cells—electrons trapped at discrete sites in silicon nitride can be released (de-trapped) more easily shortly after programming, leading to rapid threshold voltage shifts in the initial hours.

The convergence of QLC's narrow voltage margins and charge-trap flash's accelerated electron loss profoundly exacerbates early data retention failures in data-center SSDs. With only approximately 0.25 V per state, even small charge losses can cause the threshold voltage to cross state boundaries, triggering read errors and necessitating multiple read retries. Layer-to-layer RBER variation in 3D NAND further complicates voltage selection, as different layers may exhibit distinct error characteristics [34]. While conventional data migration to fresh blocks provides a straightforward mitigation path for client devices, this strategy is impractical in data centers. Persistent, high-utilization workloads leave little idle time for background migrations, and the rapid retention loss characteristic of charge-trap QLC would demand prohibitively frequent data movement.

Quantifying the Impact. To demonstrate the severity of early retention in charge-trap QLC SSDs, we conducted experiments comparing a charge-trap flash-based data-center QLC SSD without any early retention optimizations (CT-QLC$_{no\text{-}cal}$) against a typical floating-gate flash-based data-center QLC SSD (FG-QLC). We issued 100,000 random 64KB read operations at queue depth 128 at 2, 10, 18, and 28 hours after a full-drive write. The results (Figure 3) revealed significant performance degradation for CT-QLC$_{no\text{-}cal}$ compared with FG-QLC: average latency and tail latency increased by up to 1.2× and 7.1×, respectively. This early-stage gap indicates that electron loss in charge-trap QLC NAND begins rapidly after programming, and the observed latency spikes correlate directly with massive read retries triggered by threshold-voltage drift.



(a) Average read latency (b) 99.99th percentile latency

Fig. 3. Early data retention tests of a charge-trap flash-based SSD and a floating-gate flash-based SSD

Prior work has explored several directions, including generating tailored read-retry tables from device characterization, using sentinel cells to infer optimal voltages, employing data refresh for long-retention pages, and leveraging machine learning or hardware facilities to predict and tune read voltages [17]–[21], [35]–[41]. However, these approaches have limitations: offline characterization-based schemes struggle to track runtime variability; heavyweight online modeling introduces nontrivial latency overhead; and hardware-dependent solutions limit deployability in existing products.

CT-QLC addresses all three limitations through a firmware-only, hierarchical table-based approach with bounded background overhead. The key insight is that voltage accuracy requirements differ by context: factory defaults provide safety, active entries enable performance, and staging enables safe refinement. This progressive refinement eliminates the stability-overhead trade-off faced by prior approaches. CT-QLC is designed around three principles: (1) operate entirely in firmware without hardware modification; (2) exploit inexpensive background reads and statistical correlations rather than complex and unpredictable online learning; (3) ensure sustained voltage accuracy through a hierarchical table structure that progressively refines entries while bounding background overhead.

III. Design

This section presents the detailed design of CT-QLC. We first describe the architecture and core modules, then explain the foreground read voltage selection mechanism and background voltage calibration workflow.

A. Overview

The architecture of CT-QLC is illustrated in Figure 4. CT-QLC comprises five core components: **Block Monitor**, **Voltage Selector**, **Calibration Engine**, **Table Manager**, and **Three-Tier Voltage Tables**. These components work together to achieve adaptive voltage tracking with bounded background overhead, enabling deployment in commercial SSDs without hardware modifications. The following subsections describe each component in detail.

B. Core Modules

Block Monitor. The block monitor tracks block-level metadata, primarily including the retention time since data programming and the number of program/erase (P/E) cycles. These statistics provide the basis for the voltage selector to determine which voltage entries to use for each read operation.

Voltage Selector. The voltage selector serves as the central decision module with two primary responsibilities. First, for each foreground host read, it queries the block monitor to obtain block metadata (such as retention time and P/E cycles), and then maps this metadata to corresponding entries in the active table to determine the appropriate read voltages. If all entries in the active table fail to produce a successful read, the voltage selector falls back to sequentially trying voltages from the factory table until the read succeeds, ensuring data reliability even when active table entries become inaccurate. Second, it periodically identifies blocks that require voltage verification based on elapsed time since the last calibration, triggering the calibration engine to perform background calibration.

Calibration Engine. The calibration engine executes voltage verification and refinement tasks triggered by the voltage selector. In the verification phase, it reads sample pages using voltage entries from the active table, computing the fail ratio (percentage of uncorrectable reads) for each entry. The verification phase only accesses the active table because it evaluates the validity of currently deployed voltage entries.

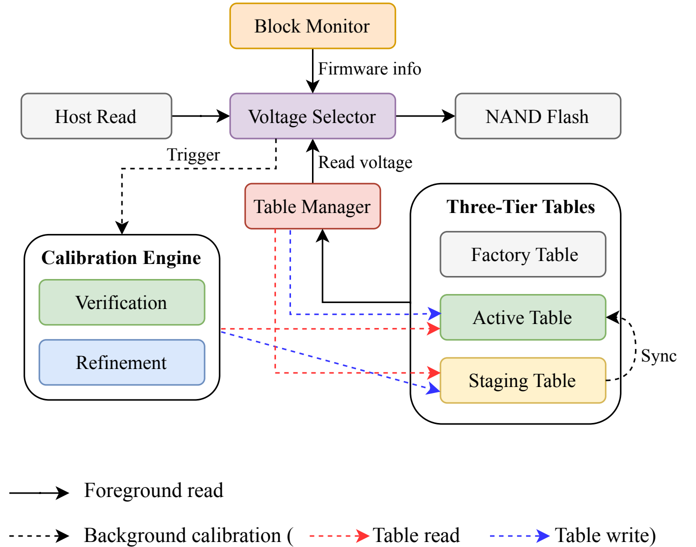


Fig. 4. Architecture of CT-QLC

In the refinement phase, the calibration engine performs two types of operations conditionally: (1) *fail ratio-based reordering*—reordering entries by their fail ratios when the current ordering is suboptimal; and (2) *adaptive window-based valley tracking*—discovering new voltage candidates through iterative window refinement when all entries fail verification. During refinement, the calibration engine reads from the active table and writes updated entries to the staging table.

Table Manager. The table manager manages the three-tier voltage tables. Based on strategies determined by the voltage selector, it performs read, copy, and update operations on the contents of these voltage tables. Specifically, the table manager performs atomic synchronization of voltage entries from the staging table to the active table upon calibration completion. Atomicity is achieved through a double-buffer mechanism: the table manager first writes updated entries to a shadow copy of the active table, then atomically updates a pointer to switch the shadow copy to the active table. This ensures that host reads always use validated entries from the active table, and that updates appear atomically rather than incrementally.

Three-Tier Voltage Tables. The three-tier table structure is designed to progressively refine voltage accuracy while maintaining system stability:

Tier 1: Factory Table. The factory table contains manufacturer-provided voltage configurations. These entries are generic—not customized for individual blocks—and use deliberately widened voltage ranges to accommodate potential shifts over the device's lifespan. While these entries are conservative (often requiring multiple retries), they serve as a reliable fallback when active entries are not available.

Tier 2: Active Table. The active table stores the current voltage entries used for read operations, with three near-optimal entries for each block (or block group). The three-entry design ensures that data can be retrieved within at most three read attempts, meeting strict QoS requirements. During SSD initialization, these entries are populated by

testing candidate voltages from the factory table and selecting those that produce the lowest error rates.

Tier 3: Staging Table. The staging table provides an isolated workspace for calibration. When voltage entries require adjustment, they are first copied to the staging table, where modifications are applied. This isolation ensures that host reads always use validated entries from the active table; only after calibration completes are updated entries atomically synchronized back to the active table.

C. Foreground Read Voltage Selection

The voltage selector determines voltage configurations for foreground host read operations by dynamically mapping block-specific metadata to corresponding entries in the voltage tables. Under normal operation, the controller uses entries from the active table as the default steady state. The block monitor provides real-time retention time and P/E cycle information, enabling the voltage selector to index into the appropriate block group and retrieve the three voltage entries optimized for that group's characteristics.

If, in rare cases, host reads using voltages from the active table still require retries, the voltage selector falls back to the factory table and sequentially traverses its voltage profiles until the read succeeds. This fallback mechanism ensures data reliability even when active table entries become inaccurate due to unexpected voltage drift.

The four states that the voltage selector may encounter when determining read voltages for foreground operations are illustrated in Figure 5. The system cycles through these states during voltage calibration, ensuring host reads always use validated entries from the active table while allowing background refinement via the staging table.

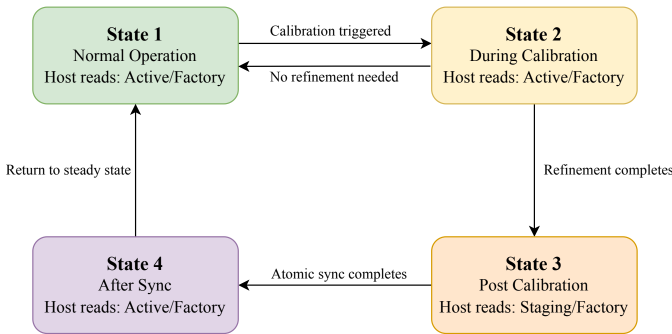


Fig. 5. State transition of three-tier voltage tables

D. Background Voltage Calibration

To maintain voltage accuracy in the active table over time, the voltage selector periodically triggers the calibration engine to perform background verification and, when necessary, voltage refinement. The calibration workflow follows a three-phase process: verification, refinement decision, and conditional execution, as illustrated in Figure 6.

Phase 1: Verification. The calibration engine reads sample pages using voltage entries from the active table and computes the fail ratio (percentage of uncorrectable reads) for each entry. This phase evaluates whether the current voltage entries remain valid for the block group's current retention characteristics. The verification uses a configurable sample size to balance accuracy with background overhead.

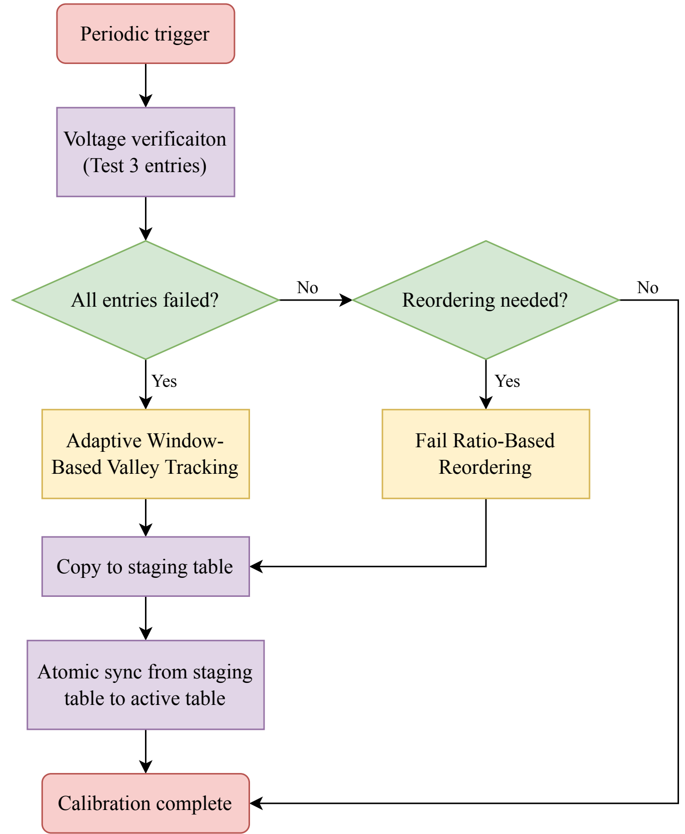


Fig. 6. Workflow of periodic voltage calibration

Phase 2: Refinement Decision. Based on the verification results, the calibration engine determines whether refinement is necessary and which type to perform. Three outcomes are possible: (1) *No action*—all entries pass verification (fail ratios below the threshold) and are optimally ordered; (2) *Fail ratio-based reordering*—all entries pass but are not optimally ordered; (3) *Adaptive window-based valley tracking*—all entries fail verification, indicating the voltages no longer align with the shifted threshold voltage distribution.

Phase 3: Conditional Execution. If refinement is needed, the calibration engine executes the selected operation. For reordering, it re-sorts the three entries by their measured fail ratios. For valley tracking, it searches for new optimal voltages using an adaptive window algorithm (Algorithm 1). After refinement, updated entries are written to the staging table and atomically synchronized to the active table.

Fail Ratio-Based Reordering. This refinement operation adjusts the priority of voltage entries to minimize read retries. Each block group has three voltage entries (Entry 1, 2, 3) that are tried sequentially if previous attempts fail. The *fail ratio* of an entry is defined as the percentage of sample pages that could not be successfully decoded using that voltage (i.e., pages with too many bit errors for LDPC correction).

Algorithm 1: Adaptive Window-Based Valley Tracking

```
Input: Base voltage v_base, base fail ratio ρ_base, threshold ϵ, limit N_max
Output: Optimized voltage entry v_opt = (v_opt^(1), v_opt^(2), v_opt^(3))
1. Initialize Voltage Window
Δ_v ← scale(8, ρ_base) ▷ Window size scales with fail ratio

v_center ← v_base
v_low ← v_center − Δ_v
v_high ← v_center + Δ_v
i ← 0 ▷ Iteration counter

last_dir ← 0 ▷ Last dir: −1=left, 1=right, 0=none

2. Dynamic Window Refinement
repeat
    i ← i + 1
    foreach v ∈ {v_low, v_center, v_high} do
        Read sample pages at voltage v
        cnt(v) ← count of bit-1 cells
    δ_left ← |cnt(v_center) − cnt(v_low)|
    δ_right ← |cnt(v_high) − cnt(v_center)|
    if δ_left − δ_right > ϵ then
        /* Valley center left */
        if last_dir = 1 then
            /* Reversal detected, shrink window */
            Δ_v ← Δ_v × 0.5
            v_low ← v_center − Δ_v
            v_high ← v_center + Δ_v
            last_dir ← 0
        else
            v_high ← v_center
            v_center ← v_low
            v_low ← v_center − Δ_v
            last_dir ← −1
    else if δ_right − δ_left > ϵ then
        /* Valley center right */
        if last_dir = −1 then
            /* Reversal detected, shrink window */
            Δ_v ← Δ_v × 0.5
            v_low ← v_center − Δ_v
            v_high ← v_center + Δ_v
            last_dir ← 0
        else
            v_low ← v_center
            v_center ← v_high
            v_high ← v_center + Δ_v
            last_dir ← 1
    else
        /* Symmetric within ϵ, terminate */
        break ;
until i ≥ N_max
3. Output Optimal Voltages
v_opt^(1) ← v_center ▷ Primary voltage

v_opt^(2) ← v_low ▷ Low backup

v_opt^(3) ← v_high ▷ High backup

return v_opt
```

During verification, the calibration engine measures fail ratios for all three entries. If all entries pass verification (fail ratios below the error threshold) but are not sorted in ascending order of fail ratio, they are reordered so that the entry with the lowest fail ratio becomes Entry 1 (tried first), the second-lowest becomes Entry 2, and the highest becomes Entry 3. This ordering ensures that the most reliable voltage is attempted first, maximizing the probability that the first read succeeds and avoiding the latency penalty of retries. If the entries are already optimally ordered, no reordering is performed.

Adaptive Window-Based Valley Tracking. This refinement operation is invoked when all three entries exceed the fail threshold, indicating that the current voltages no longer provide acceptable read accuracy. The algorithm works as follows: (1) Start from the entry with the lowest fail ratio as the base voltage and measure its fail ratio ρ_{base}; (2) Establish an initial voltage window around this base value, with the window size Δ_v scaled proportionally to ρ_{base} (base value of 8 offset units) to provide a wider search range for more severely drifted voltages; (3) Test the voltage at the boundaries and midpoint by reading sample pages and measuring the bit-1 distribution; (4) Determine the valley center direction using threshold-aligned branching — move the window only when the asymmetry exceeds ϵ, otherwise treat the current position as converged; (5) Apply direction-reversal detection to break oscillation cycles by immediately shrinking the window when the search direction reverses. The loop terminates either when the symmetry metric falls within ϵ (converged via explicit break) or when the iteration count reaches $N_{\max}$ (safety limit). The final midpoint becomes the primary voltage entry, with the boundaries serving as backup entries, as illustrated in Algorithm 1.

The symmetry of the valley is evaluated using the bit-1 count metric $\text{cnt}(v)$. In NAND flash, each programmed state follows a roughly Gaussian threshold-voltage distribution, and the valley lies at the intersection of two adjacent states. When the read voltage is centered at the true valley, the cell counts transitioning between the three sampling points are approximately balanced — that is, $\text{cnt}(v_{\text{center}}) - \text{cnt}(v_{\text{low}}) \approx \text{cnt}(v_{\text{high}}) - \text{cnt}(v_{\text{center}})$. Let $\delta_{\text{left}} = |\text{cnt}(v_{\text{center}}) - \text{cnt}(v_{\text{low}})|$ and $\delta_{\text{right}} = |\text{cnt}(v_{\text{high}}) - \text{cnt}(v_{\text{center}})|$ denote the counts in the left and right halves of the window. The algorithm identifies the valley center by checking whether the asymmetry $|\delta_{\text{left}} - \delta_{\text{right}}|$ exceeds the threshold ϵ.

E. Implementation Optimizations

1) Block Group-Based Voltage Calibration: High-capacity QLC SSDs typically contain hundreds of thousands of blocks. To facilitate management, the firmware logically organizes blocks across different channels and chips into superblocks for erase and write operations. Leveraging the observation that superblocks with similar wear levels exhibit comparable electron loss rates, we group superblocks based on P/E cycles and retention time, with each group sharing entries in the active table.

Block grouping is based on the observation that blocks with similar P/E cycle counts and retention times exhibit statistically similar voltage drift characteristics. This homogeneity assumption is grounded in the physical properties of NAND flash: blocks that have undergone similar program/erase stress and have been written at approximately the same time will experience comparable charge loss rates. While individual blocks may exhibit minor variations due to manufacturing variation, the average behavior within a group remains sufficiently consistent for voltage refinement purposes.

In our implementation, block grouping is organized as a Cartesian product of two independent dimensions: P/E cycle bins and retention time bins. Empirical measurements show that P/E cycles have a relatively minor influence on voltage drift, particularly at lower counts, while retention time exerts a greater impact. We therefore configure coarser granularity for P/E cycles (4 bins of 1,000 cycles each) and finer granularity for retention time (11 bins: 2–5 hours for the first 48 hours, 24 hours for 72 hours–7 days, and 7 days beyond one week). This partitioning yields 4×11 = 44 block groups for a typical 15.36TB SSD. Each block group maintains three voltage entries (Entry 1, 2, 3), consistent with the per-block design in Section III. The active voltage table stores 15 threshold offsets per entry (1 byte each), yielding 3×15 = 45 bytes per group, or 45×44 = 1980 bytes in total. Including incremental per-block metadata (group assignment and read-disturb counters, 4 bytes per block) and calibration working buffers (4.5 KB peak), the total firmware metadata overhead remains below 10 KB—orders of magnitude smaller than the DRAM available on enterprise SSD controllers.

2) Sentinel Voltage: For individual flash cells, intrinsic correlations exist between distinct threshold voltages. To validate this phenomenon, we collect 15 threshold-voltage distributions from the same QLC SSD across varying retention periods. The interconnected voltage curves exhibit consistent trends at each retention point, despite different amounts of shift caused by retention aging.

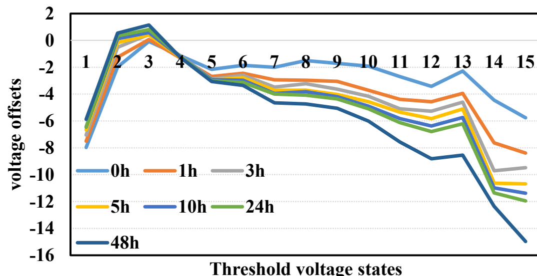


Fig. 7. Illustration of sentinel voltage correlation

Consequently, we designate one voltage per page type as the sentinel voltage and build linear models offline to characterize its relationship with all other voltages in the same page type. While a single global sentinel across all read voltages would maximize compression, we retain one sentinel per page type to preserve accuracy, as each page type exhibits distinct drift characteristics. We select representative valleys (valley 7 for LSB, valley 6 for CSB, valley 8 for MSB, and valley 9 for TSB) as sentinels because they exhibit moderate drift rates that are representative of the full voltage range for each page type. The model coefficients ($\alpha_{k,0}$, $\alpha_{k,1}$) are learned offline from device-characterization data using standard least-squares fitting. During adaptive window-based valley tracking, only the sentinel voltage $\hat{V}_s$ is actively searched within a narrowed voltage window; all remaining voltages are computed by evaluating the corresponding linear model:

$$\hat{V}_k \approx \alpha_{k,0} + \alpha_{k,1}\,\hat{V}_s, \quad \forall k \neq s, \tag{1}$$

where $\hat{V}_s$ is the calibrated sentinel voltage and $\hat{V}_k$ are the derived read reference voltages. This sentinel-based projection eliminates exhaustive voltage-window searches for every threshold, substantially reducing extra read operations during refinement while preserving the accuracy of read voltages.

Beyond reducing read operations during refinement, the sentinel-voltage projection also yields a significant storage benefit for the active table. Recall that the baseline active table stores 15 threshold offsets per entry (45 bytes per block group). With sentinel projection, each entry needs to store only a sentinel offset for each page type (4 bytes per entry instead of 15), while the three-entry structure per block group is preserved to maintain read-retry coverage. For 44 block groups, the active-table footprint drops from 1980 bytes to 528 bytes—a $3.75\times$ reduction—while maintaining coverage of all 15 read reference voltages, which are reconstructed on demand via Equation (1).

The linear correlation between sentinel and other voltages is grounded in the physical properties of charge-trap flash: all threshold voltages in a page type experience correlated drift due to shared retention aging mechanisms. Our empirical validation across 15 retention points (0–48 hours) shows consistent linear trends with a coefficient of determination $R^2 > 0.8$ for all state pairs. Nevertheless, the model leaves approximately 20% of variance unexplained; under certain conditions this can cause projected voltages to deviate toward state boundaries. When deviations occur, the fail ratio-based reordering mechanism mitigates the impact by correcting entry priority at runtime. We leave more sophisticated modeling approaches (e.g., piecewise linear or nonlinear regression) as future work to further improve projection accuracy under diverse operating conditions.

3) Read-Disturb Effect Considerations: Read disturb is another factor that affects long-term reliability: repeated reads on the same block gradually shift neighboring cells' threshold voltages, increasing raw bit error rates. CT-QLC does not introduce a dedicated read-disturb optimization; instead, its periodic voltage verification naturally captures the combined effect of retention drift and accumulated read disturb within each calibration interval, because the measured fail ratios reflect the aggregate threshold-voltage shift from both sources. Furthermore, by limiting background reads through block-group sampling and sentinel-voltage projection, CT-QLC ensures that its own calibration activity does not exacerbate read disturb. For blocks that exceed the firmware's standard

read-count thresholds, conventional read-disturb refresh mechanisms continue to operate independently.

IV. Evaluation

We integrated CT-QLC into an actual charge-trap flash-based QLC SSD product. Unless otherwise specified, all performance tests were conducted with CT-QLC's background voltage calibration enabled (periodic verification every 5 hours, with 2 blocks sampled per group and 64 pages sampled per block using a step-63 sampling pattern to cover different QLC page types), and we measured the impact on foreground read operations. In this section, we conduct comprehensive experimental evaluations to:

- Evaluate the foreground read performance of CT-QLC (with background calibration active) by comparing with existing data-center TLC and QLC SSD products.
- Evaluate the reliability of CT-QLC by comparing with state-of-the-art data-center QLC SSD products based on floating-gate flash technology.
- Decompose and evaluate the reliability improvements contributed by individual refinement techniques within CT-QLC.

A. Experimental Setup

1) Testbed: CT-QLC is implemented as a productized firmware module running on commercial SSD controllers. All experiments are conducted on production hardware without any controller or NAND modifications.

CT-QLC adopts a PCIe Gen4 SSD controller and SK hynix V7 QLC NAND [42]. The SSD capacity is 15.36TB with 8 channels, 8 chips per channel, 2 dies per chip, 4 planes per die, 410 blocks per plane, and 1408 wordlines per block.

All experiments are performed on Inspur TV75S9 servers equipped with a 48-core Intel Xeon Platinum 8331C CPU and 256GB DRAM, running Ubuntu 20.04, with data transferred via the PCI Express (PCIe) 4.0 interface.

2) Workloads: We conduct tests using synthetic workloads with configurable numbers of threads (Job), command queue depths (QD), and block/data sizes, as well as real business workload traces and six popular workloads from Microsoft Research Cambridge [43].

3) SSD Configurations: The experiments compare CT-QLC with several configurations summarized in Table I.

TABLE I
Summary of SSD Configurations

Config	Description
A-TLC	Data-center TLC SSD, 8 channels, 7.68TB
S-TLC	Data-center TLC SSD, 16 channels, 7.68TB
FG-QLC	Floating-gate QLC SSD (vendor-default firmware), 16 channels, 15.36TB
CT-QLC	Proposed charge-trap QLC SSD, 8 channels, 15.36TB
CT-QLC$_{no\text{-}cal}$	CT-QLC without voltage calibration (baseline)
CT-QLC$_{w/or}$	CT-QLC without fail-ratio reordering
CT-QLC$_{w/oc}$	CT-QLC without adaptive window-based valley tracking
CT-QLC$_{w/ora}$	CT-QLC with read-ahead firmware mechanism (optimizing low-QD sequential reads) disabled

4) Metrics: Since CT-QLC focuses on guaranteeing QoS for read-intensive workloads, the primary evaluation metric is the 99.99th percentile read latency. We additionally report the average read latency as a complementary metric. All experimental results are measured at steady state, with each experiment repeated five times to ensure reliability. We report the average values across these repeated runs, noting that the standard deviations observed were within 3–5% of the mean for all reported metrics, indicating consistent and reproducible evaluation results.

B. Synthetic Workload Performance

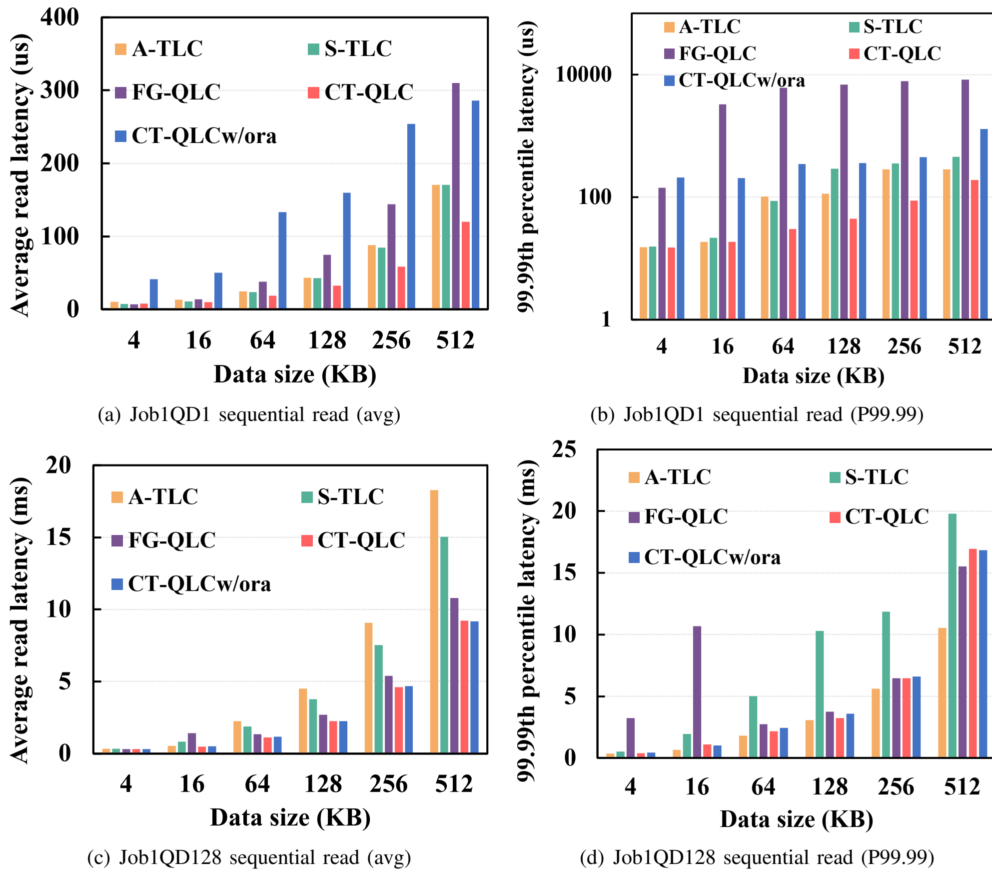
The synthetic workload tests include pure read scenarios with varying numbers of threads and queue depths. For completeness, the results include comparisons with TLC SSDs (A-TLC and S-TLC) to demonstrate that QLC NAND can achieve competitive read performance. However, the primary focus of the following analysis is on comparing CT-QLC with existing QLC SSDs (FG-QLC), as this directly evaluates the impact of our voltage calibration framework. Figure 8 shows sequential read results, and Figure 9 shows random read results.

As shown in Figure 8, read-ahead mechanisms in modern SSD firmware can substantially impact Job1QD1 sequential read latency. At low queue depths, NAND channels typically have significant available idle time, allowing controllers to proactively prefetch subsequent data blocks. This prefetching masks the raw NAND access latency and any potential retry overhead, which is why we utilize the CT-QLC$_{w/ora}$ configuration to isolate and evaluate the specific benefits of our voltage tracking algorithm. Under the Job1QD128 configuration, the performance of CT-QLC and CT-QLC$_{w/ora}$ is nearly identical, confirming that at high queue depths the read-ahead mechanism has negligible effect.

At high queue depths (Job1QD128), frequent read retries in FG-QLC create head-of-line blocking across the deep request queue, causing severe tail latency degradation; CT-QLC's voltage optimization reduces retry frequency and eliminates this queue buildup effect. Under Job1QD128, the sequential read latency of CT-QLC is reduced by up to 65% compared with FG-QLC, with an average reduction of 21%. For the 99.99th percentile latency, CT-QLC achieves a reduction of up to 89% in the 16KB data size scenario, with an overall average reduction of 27%.

As shown in Figure 9, CT-QLC demonstrates outstanding average-latency performance due to optimized voltage selection reducing read retries. Overall, the random read latency of CT-QLC is reduced by up to 49% compared with FG-QLC, and the 99.99th percentile latency is reduced by up to 52%.

Similar to the sequential read results, we observe that FG-QLC occasionally achieves lower latency in specific random read scenarios, particularly at smaller data sizes (4KB) with low queue depths. This can be attributed to FG-QLC's inherent advantage in data retention characteristics due to floating-gate technology, which exhibits slower charge loss compared with charge-trap flash. In these light-load scenarios, overall read latency is dominated by base NAND access time rather



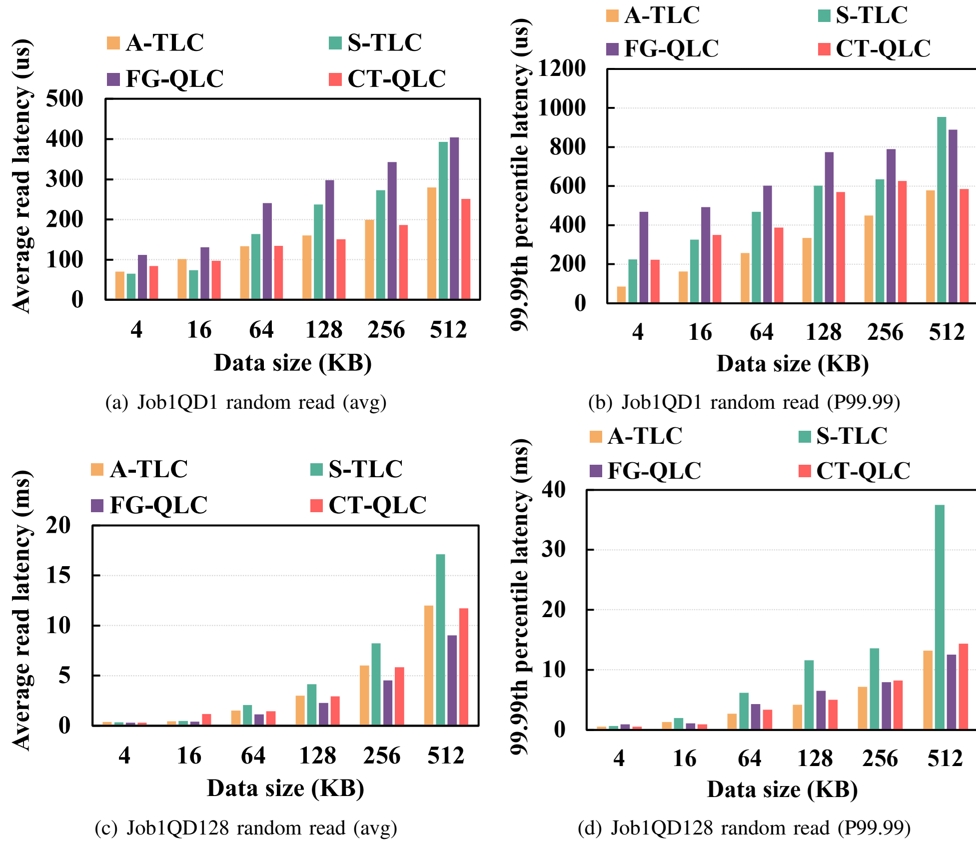
(a) Job1QD1 sequential read (avg)

(b) Job1QD1 sequential read (P99.99)

(c) Job1QD128 sequential read (avg)

(d) Job1QD128 sequential read (P99.99)

Fig. 8. Sequential read performance results



(a) Job1QD1 random read (avg)

(b) Job1QD1 random read (P99.99)

(c) Job1QD128 random read (avg)

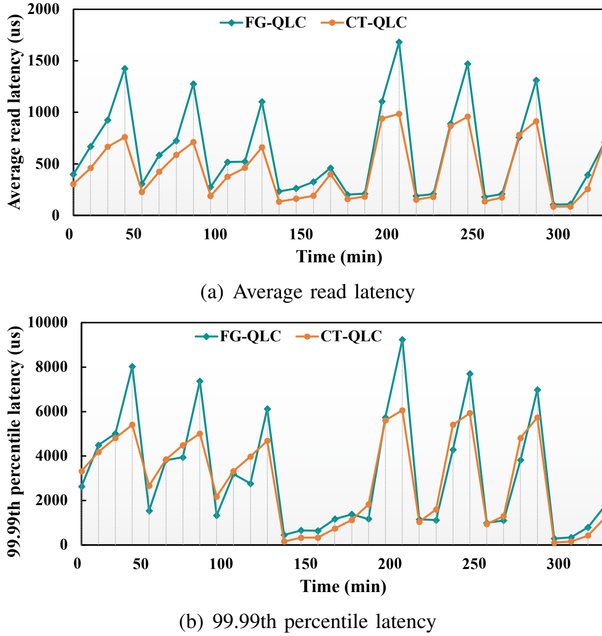
(d) Job1QD128 random read (P99.99)

Fig. 9. Random read performance results

than retry overhead, allowing FG-QLC's retention advantage to manifest as slightly lower baseline latency. However, as data size increases or queue depth grows, retry frequency rises and CT-QLC's adaptive voltage calibration becomes dominant, consistently delivering superior tail-latency performance across the majority of tested scenarios.

C. Real-World Trace Performance

The business trace includes 32 workloads with different access patterns. As shown in Figure 10, CT-QLC demonstrates superior performance throughout the tests. Compared with FG-QLC, the average read latency is reduced by up to 44% (with an overall average reduction of 25%), while the 99.99th percentile latency is reduced by up to 65%.



(a) Average read latency

(b) 99.99th percentile latency

Fig. 10. Business trace performance results

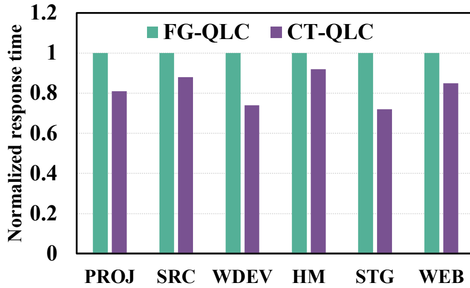
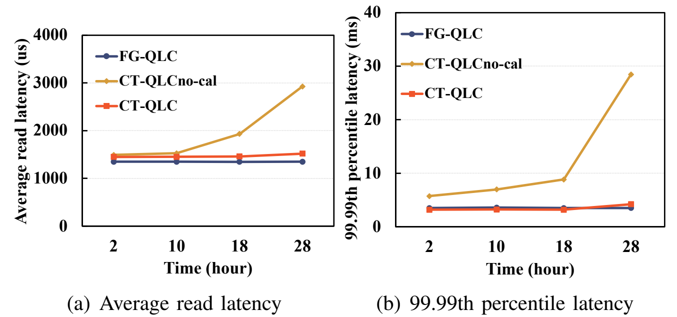


Fig. 11. Real-world traces

We further conduct experiments on six real workloads from Microsoft Research Cambridge, and the performance results of CT-QLC and FG-QLC are shown in Figure 11. On average, CT-QLC demonstrates lower response times with reductions of up to 28%, indicating significantly fewer read retry occurrences during the testing phase.

D. Retention and Temperature Tests

1) Short-Duration Early Retention Tests: Figure 12 shows that the early data retention problem is significantly mitigated by the periodic voltage verification and refinement mechanisms compared with the CT-QLC$_{no\text{-}cal}$ configuration. This test uses the same experimental setup as described in Section II-C (100,000 random 64KB read operations at queue depth 128 at 2, 10, 18, and 28 hours after programming), enabling direct comparison with the results in Figure 3. While the unoptimized CT-QLC$_{no\text{-}cal}$ exhibited 7.1× tail latency degradation due to early retention loss, CT-QLC with periodic calibration maintained stable performance throughout the test period. This improvement stems from CT-QLC's three-phase calibration workflow: Phase 1 (Verification) detects voltage drift through fail ratio measurement; Phase 2 (Refinement Decision) determines whether reordering or valley tracking is needed; Phase 3 (Conditional Execution) applies the appropriate refinement. The fail ratio-based reordering ensures that the most reliable voltage is tried first, while adaptive window-based valley tracking discovers new optimal voltages when all existing entries fail. This hierarchical approach directly addresses the early retention problem identified in Section II-C, enabling rapid response to charge loss without requiring continuous voltage search.



(a) Average read latency

(b) 99.99th percentile latency

Fig. 12. Short-duration tests of early data retention

2) Long-Duration Data Retention Tests: Long-duration tests at 25°C (Figure 13) and 55°C (Figure 14) show that CT-QLC reduces the 99.99th percentile latency by 12% and 94%, respectively, compared with CT-QLC$_{no\text{-}cal}$. The dramatic improvement at high temperature (55°C) is attributed to the adaptive window-based valley tracking algorithm, which effectively compensates for accelerated charge loss under thermal stress. At high temperatures, charge-trap cells exhibit faster electron de-trapping, causing threshold voltage distributions to shift more rapidly and produce higher fail ratios. The adaptive window mechanism scales the initial search window proportionally to the measured fail ratio, enabling wider search ranges for severely drifted voltages while using narrower windows for milder drift. Combined with iterative direction-guided refinement and direction-reversal detection, this allows the algorithm to efficiently converge to optimal voltages across varying drift severities. At room temperature (25°C), the improvement is more modest because voltage drift occurs more slowly; the fail ratio-based reordering alone is

often sufficient to maintain good read performance, with valley tracking invoked only occasionally.

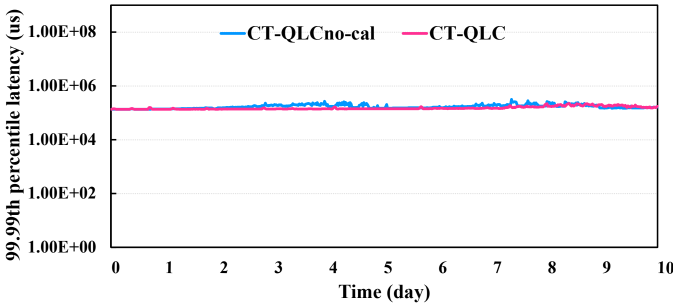


Fig. 13. Long-duration tests at room temperature (25°C)

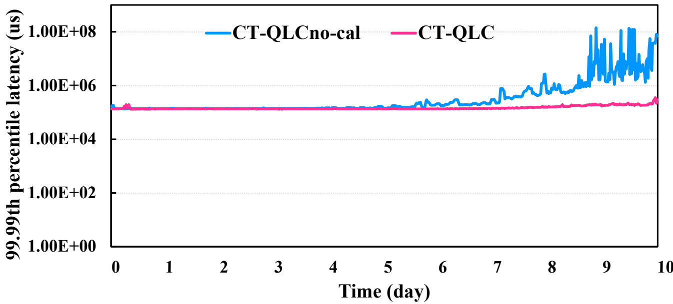
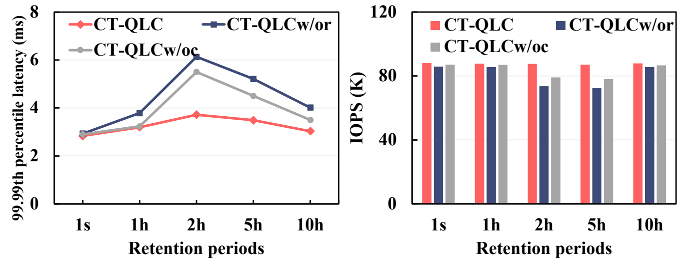


Fig. 14. Long-duration tests at high temperature (55°C)

3) Component Analysis: Figure 15 shows the contributions of voltage reordering and adaptive window-based valley tracking, demonstrating that both QoS and IOPS are optimized by these techniques. The fail ratio-based reordering provides immediate benefits by ensuring the most reliable voltage is attempted first, reducing the average number of read retries. This optimization is lightweight—requiring only the comparison and sorting of three entries—and can be performed frequently without significant overhead. The adaptive window-based valley tracking provides deeper optimization when reordering alone is insufficient: by iteratively refining the voltage window based on bit distribution balance metrics, the algorithm converges to near-optimal voltages that minimize bit errors. The combination of these two techniques—fast reordering for minor drift and thorough valley tracking for major shifts—enables CT-QLC to maintain low tail latency across varying retention conditions while keeping background overhead bounded.

E. End-of-Life Tests

Given that data-center QLC SSDs typically have a P/E cycle limit of 2,000–3,000, we conducted end-of-life (EOL) testing to evaluate performance stability across the device lifetime. We selected 300 and 2,500 P/E cycles as representative early-life and near-end-of-life points. While intermediate P/E cycle measurements (e.g., 500, 1,000, 1,500, 2,000) could provide finer-grained insights, our focus is on demonstrating performance stability across the operational lifetime rather than characterizing the complete degradation curve. The two-point



(a) 99.99th percentile latency (b) IOPS

Fig. 15. Performance benefits of voltage refinement

comparison effectively captures the worst-case performance variation between initial deployment and near-end-of-life conditions, which is the primary concern for data-center operators planning multi-year deployments. Comparative QoS results at these two stages are shown in Figure 16.

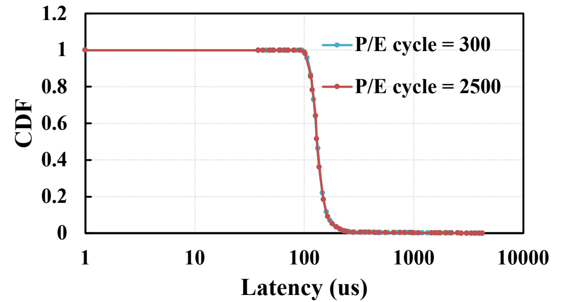


Fig. 16. Latency distribution of end-of-life tests

TABLE II
QoS Comparison at Different P/E Cycles

	QoS - percentile latency (μs)			
	99.9th	99.99th	99.999th	99.9999th
P/E = 300	162	242	1704	4030
P/E = 2500	165	246	1753	4325

The results reveal negligible differences in QoS performance across the entire latency distribution, with less than 3% degradation at the 99.9th–99.999th levels and about 7% at the 99.9999th percentile.

F. Overhead Analysis

The overhead of CT-QLC primarily stems from background read operations during voltage verification and refinement. These background operations run concurrently with foreground host I/O, competing for NAND channel bandwidth and controller resources. When voltage entries pass verification and no reordering is needed, extra reads result solely from verification. When reordering is needed, no additional reads are required since fail ratios are already available from verification. When valley tracking is invoked, additional reads are needed but remain bounded by the iteration limit. Importantly, all performance results reported in previous sections (Sections IV-B–IV-E) include this background overhead, demonstrating that CT-QLC achieves significant tail-latency improvements even when accounting for the cost of maintaining voltage accuracy.

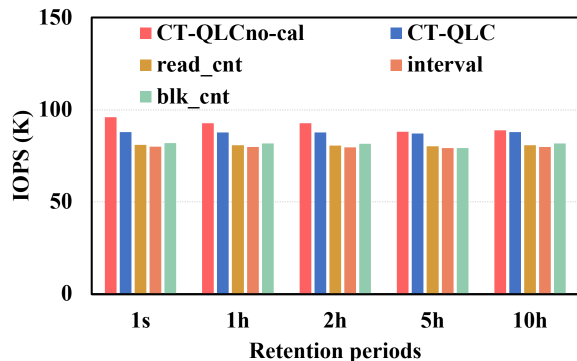


Fig. 17. Overhead of background voltage calibration

Figure 17 shows the overhead evaluation under varying configuration parameters. In addition to CT-QLC and CT-QLC$_{no\text{-}cal}$, the bars are grouped by three configuration parameters shown in the legend, each representing a more aggressive setting compared with the default configuration: increasing `blk_cnt` (more blocks verified per group), increasing `read_cnt` (more pages sampled per block), and decreasing `interval` (shorter verification interval). Each bar shows the IOPS under a specific parameter configuration. Compared with the default CT-QLC configuration, increasing `blk_cnt` or `read_cnt` and shortening `interval` all result in additional background read overhead.

These parameter adjustments improve voltage tracking accuracy by providing more samples or fresher data, but they also increase background read overhead. Compared with the default CT-QLC configuration, increasing `blk_cnt` or `read_cnt` and shortening `interval` all result in additional IOPS degradation. The results demonstrate that even with aggressive calibration settings (more blocks sampled, more pages read, shorter intervals), the IOPS degradation remains within 5–8% of the baseline. Under typical operational configurations, the overhead is limited to 2–3%, which is considered acceptable for data-center deployments where tail-latency guarantees are prioritized. This trade-off between accuracy and overhead allows operators to tune the calibration aggressiveness based on workload requirements and retention characteristics.

V. Related Work

Read voltage optimization has been extensively studied to address threshold voltage drift in NAND flash storage [16], [31], [44]–[46]. Parallel read-retry mechanisms have been explored to reduce read latency in 3D NAND SSDs [47], and invalid-data-assisted techniques have shown promise for improving reliability in high-density flash [48]. Prior approaches can be broadly categorized along two dimensions: (1) whether voltages are determined statically at manufacturing time or adapted dynamically at runtime; and (2) whether the solution requires hardware support or can be implemented purely in firmware. CT-QLC occupies a unique position in this design space as a firmware-only runtime adaptive solution.

Static Voltage Optimization. These approaches determine optimal read voltages through offline device characterization. Ye et al. [17], [18] propose methodologies to generate tailored read-retry tables for each flash model based on extensive testing across temperature and retention conditions. While achieving good accuracy for characterized conditions, static approaches cannot adapt to runtime variations such as temperature changes, workload-dependent aging, or manufacturing process variations across different SSD batches. CT-QLC addresses this limitation by combining factory-provided defaults with lightweight runtime calibration.

Dynamic Voltage Optimization. These approaches adjust voltages during SSD operation to track threshold voltage drift. Several techniques have been explored: (1) *Sentinel-based inference*—Shaving [19] infers voltages from OOB sentinel cells, requiring a custom controller unavailable in commercial firmware. Its zero-overhead sentinel read assumes a single-voltage LSB page, a mapping specific to its evaluated chip; commercial NAND typically uses multiple voltages per page, and QLC has no single-voltage page. Thus Shaving is inapplicable to commercial QLC SSDs. (2) *Data refresh*—LDR [35] actively corrects errors by rewriting data with refreshed charge levels, but introduces write amplification; (3) *Machine learning*—recent works [20], [36]–[40] use ML to predict optimal voltages from error statistics, but introduce significant computational overhead; (4) *Independent read schemes*—recent work [49] proposes independent read operations for QLC SSDs to reduce read latency by exploiting page-type-specific characteristics. CT-QLC provides a lightweight alternative: it implements sentinel voltage modeling in firmware without hardware dependencies, avoids write amplification through voltage tuning rather than data movement, and achieves accuracy comparable to that of ML approaches through hierarchical table management with bounded overhead.

Hardware-Assisted Optimization. These approaches exploit specialized hardware capabilities [41] or NAND-specific commands [21] to accelerate voltage tuning. While effective, such solutions require specific controller or NAND support that limits deployability in existing products. CT-QLC achieves similar calibration accuracy entirely through firmware, enabling immediate deployment on commercial SSDs without hardware modifications.

VI. Conclusion

In this paper, we present CT-QLC, a production-ready firmware solution for addressing early data retention challenges in charge-trap QLC SSDs deployed in data centers. CT-QLC introduces a three-tier voltage table structure that progressively refines read voltage accuracy while maintaining system stability. Through hierarchical offset management, periodic verification, and adaptive window-based valley tracking, CT-QLC tracks near-optimal read voltages with bounded background overhead.

The key technical contribution lies in the design of a lightweight, production-ready voltage management framework that balances accuracy, overhead, and practical deployability.

Unlike prior approaches that require hardware modifications or complex online learning, CT-QLC can be implemented entirely in the firmware of commercial SSDs with bounded resource usage and predictable performance impact. Experimental results demonstrate that CT-QLC reduces 99.99th percentile read latency by up to 94% at high temperature compared with unoptimized charge-trap QLC SSDs and by up to 89% compared with floating-gate QLC SSDs, while maintaining stable performance across the device lifetime. These results establish the viability of charge-trap QLC SSDs for read-intensive data-center applications when equipped with appropriate firmware-level optimizations.

We acknowledge several limitations that suggest directions for future work. First, CT-QLC assumes that blocks with similar P/E cycles and retention times exhibit comparable voltage drift characteristics. While this assumption holds statistically, individual blocks may deviate due to manufacturing variations or localized wear patterns, potentially affecting voltage accuracy for outlier blocks. Second, the sentinel voltage projection uses a linear model to estimate other voltage states. While validated with $R^2 > 0.8$, this remains an approximation that may introduce suboptimal voltages under extreme operating conditions or for devices with non-standard retention characteristics. Third, CT-QLC primarily targets data retention drift. While we include basic read-disturb mitigation, read-intensive workloads may still experience performance degradation due to read-disturb effects that are not fully compensated for by the current calibration intervals. Future work includes refining block-grouping strategies to handle manufacturing variations, evaluating the sensitivity of block-grouping parameters and quantifying within-group voltage variance bounds to further validate the homogeneity assumption, exploring nonlinear projection models for improved voltage estimation, and developing adaptive calibration intervals that dynamically adjust to workload characteristics while maintaining the bounded overhead guarantees established in this work.

REFERENCES

[1] K. Smith, "Using qlc ssds to improve cost/performance tradeoffs for warm data," in *Proc. Flash Memory Summit*, 2019.

[2] Q. Chen, S. Wang, Y. Zhou, F. Wu, S. Li, Z. Wang, and C. Xie, "Paca: A page type aware read cache scheme in qlc flash-based ssds," in *2022 IEEE 40th International Conference on Computer Design (ICCD)*. IEEE, 2022, pp. 59–66.

[3] F. Insights, "Forecast Market Share of QLC," 2023. [Online]. Available: http://www.forward-insights.com/reportslist.html

[4] Y. Sun, "QLC Considerations for Mainstream Adoption — Data Center Storage," 10 2023. [Online]. Available: https://www.solidigmtechnology.cn/products/technology/qlc-considerations-for-mainstream-adoption.html

[5] N. Shibata, K. Kanda, T. Shimizu, J. Nakai, O. Nagao, N. Kobayashi, M. Miakashi, Y. Nagadomi, T. Nakano, T. Kawabe *et al.*, "A 1.33-tb 4-bit/cell 3-d flash memory on a 96-word-line-layer technology," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 1, pp. 178–188, 2019.

[6] W. Cho, J. Jung, J. Kim, J. Ham, S. Lee, Y. Noh, D. Kim, W. Lee, K. Cho, K. Kim *et al.*, "A 1-tb, 4b/cell, 176-stacked-wl 3d-nand flash memory with improved read latency and a 14.8 gb/mm2 density," in *2022 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 65. IEEE, 2022, pp. 134–135.

[7] S. Liang, Z. Qiao, S. Tang, J. Hochstetler, S. Fu, W. Shi, and H.-B. Chen, "An empirical study of quad-level cell (qlc) nand flash ssds for big data applications," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 3676–3685.

[8] "Qlc nand flash: Cost analysis for data center storage," Forward Insights NAND Market Report, 2025, industry analysis report on QLC SSD cost advantages.

[9] Y. Zhou, E. Xu, L. Zhang, K. Karkra, M. Barczak, W. Gao, W. Malikowski, M. Kozlowski, Ł. Łasek, R. Lu *et al.*, "Csal: the next-gen local disks for the cloud," in *Proceedings of the Nineteenth European Conference on Computer Systems*, 2024, pp. 608–623.

[10] L. Smith, "Intel P5316 SSD Review (30.72TB)," 12 2021. [Online]. Available: https://www.storagereview.com/review/intel-p5316-ssd-review-30-72tb

[11] Solidigm, "SolidIgmTM D5-P5336 Product Brief," 6 2023. [Online]. Available: https://www.solidigm.com/products/data-center/product-briefs/d5-p5336-product-brief.html

[12] E. F. Haratsch, "Nand flash media management algorithms," *Flash Memory Summit*, 2017.

[13] Y. Shim, M. Kim, M. Chun, J. Park, Y. Kim, and J. Kim, "Exploiting process similarity of 3d flash memory for high performance ssds," in *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019, pp. 211–223.

[14] Y. Cai, Y. Luo, E. F. Haratsch, K. Mai, and O. Mutlu, "Data retention in mlc nand flash memory: Characterization, optimization, and recovery," in *2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2015, pp. 551–563.

[15] Y. Luo, S. Ghose, Y. Cai, E. F. Haratsch, and O. Mutlu, "Enabling accurate and practical online flash channel modeling for modern mlc nand flash memory," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 9, pp. 2294–2311, 2016.

[16] ——, "Heatwatch: Improving 3d nand flash memory device reliability by exploiting self-recovery and temperature awareness," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2018, pp. 504–517.

[17] M. Ye, Q. Li, Y. Lv, J. Zhang, T. Ren, D. Wen, T.-W. Kuo, and C. J. Xue, "Achieving near-zero read retry for 3d nand flash memory," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2024, pp. 55–70.

[18] M. Ye, Q. Li, J. Nie, T.-W. Kuo, and C. J. Xue, "Valid window: a new metric to measure the reliability of nand flash memory," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2020, pp. 109–114.

[19] Q. Li, M. Ye, Y. Cui, L. Shi, X. Li, T.-W. Kuo, and C. J. Xue, "Shaving retries with sentinels for fast read over high-density 3d flash," in *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 2020, pp. 483–495.

[20] H. Hu, G. Han, W. Wu, and C. Liu, "Channel parameter and read reference voltages estimation in 3-d nand flash memory using unsupervised learning algorithms," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 43, no. 1, pp. 305–318, 2023.

[21] J. Park, M. Kim, M. Chun, L. Orosa, J. Kim, and O. Mutlu, "Reducing solid-state drive read latency by optimizing read-retry," in *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2021, pp. 702–716.

[22] A. Khakifirooz, S. Balasubrahmanyam, R. Fastow, K. H. Gaewsky, C. W. Ha, R. Haque, O. W. Jungroth, S. Law, A. S. Madraswala, B. Ngo *et al.*, "30.2 a 1tb 4b/cell 144-tier floating-gate 3d-nand flash memory with 40mb/s program throughput and 13.8 gb/mm 2 bit density," in *2021 IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 64. IEEE, 2021, pp. 424–426.

[23] W. Liu, F. Wu, X. Chen, M. Zhang, Y. Wang, X. Lu, and C. Xie, "Characterization summary of performance, reliability, and threshold voltage distribution of 3d charge-trap nand flash memory," *ACM Transactions on Storage (TOS)*, vol. 18, no. 2, pp. 1–25, 2022.

[24] W. Liu, F. Wu, J. Zhou, M. Zhang, C. Yang, Z. Lu, Y. Wang, and C. Xie, "Modeling of threshold voltage distribution in 3d nand flash memory," in *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2021, pp. 1729–1732.

[25] L. Shijun and Z. Xuecheng, "Analysis of 3d nand technologies and comparison between charge-trap-based and floating-gate-based flash devices," *The Journal of China Universities of Posts and Telecommunications*, vol. 24, no. 3, pp. 75–96, 2017.

[26] K. Kim and J.-H. Park, "3d nand flash memory technology: Layer scaling trends," *IEEE Transactions on Electron Devices*, vol. 70, no. 10, pp. 4985–4993, 2023.

[27] F. Wu, Z. Lu, Y. Zhou, X. He, Z. Tan, and C. Xie, "Ospada: One-shot programming aware data allocation policy to improve 3d nand flash read performance," in *2018 IEEE 36th International Conference on Computer Design (ICCD)*. IEEE, 2018, pp. 51–58.

[28] N. Express, "NVM Express," 6 2022. [Online]. Available: https://nvmexpress.org/resource/features-for-error-reporting-smart-log-pages-failures-and-management-capabilities-in-nvme-architectures/

[29] Y. Luo, "Architectural techniques for improving nand flash memory reliability," *arXiv preprint arXiv:1808.04016*, 2018.

[30] A. Higginbotham, "NAND Flash 101: Enterprise vs. Client SSDs - Phison Blog," 6 2022. [Online]. Available: https://phisonblog.com/nand-flash-101-enterprise-vs-client-ssds-2/

[31] Y. Luo, S. Ghose, Y. Cai, E. F. Haratsch, and O. Mutlu, "Improving 3d nand flash memory lifetime by tolerating early retention loss and process variation," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 2, no. 3, pp. 1–48, 2018.

[32] H. Park, J. Kim, J. Choi, D. Lee, and S. H. Noh, "Incremental redundancy to reduce data retention errors in flash-based ssds," in *2015 31st Symposium on Mass Storage Systems and Technologies (MSST)*. IEEE, 2015, pp. 1–13.

[33] B. Schroeder, R. Lagisetty, and A. Merchant, "Flash reliability in production: The expected and the unexpected," in *14th USENIX Conference on File and Storage Technologies (FAST 16)*, 2016, pp. 67–80.

[34] S. Nie, Y. Zhang, W. Wu, and J. Yang, "Layer rber variation aware read performance optimization for 3d flash memories," in *2020 57th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2020, pp. 1–6.

[35] Y. Du, Q. Li, L. Shi, D. Zou, H. Jin, and C. J. Xue, "Reducing ldpc soft sensing latency by lightweight data refresh for flash read performance improvement," in *Proceedings of the 54th Annual Design Automation Conference 2017*, 2017, pp. 1–6.

[36] G. Wu, X. Yao, Y. Chen, Q. Li, J. Zhang, and C. J. Xue, "On the accurate and robust prediction of optimal read voltages in 3d nand via data augmentation," in *2024 4th International Conference on Intelligent Technology and Embedded Systems (ICITES)*. IEEE, 2024, pp. 47–53.

[37] Z. Piao, D. Wei, H. Xiang, L. Qiao, and X. Peng, "Lvde: A lightweight threshold voltage distribution estimation strategy for high-performance 3-d nand flash memory," *IEEE Transactions on Circuits and Systems I: Regular Papers*, 2025.

[38] Y. Cai, E. F. Haratsch, O. Mutlu, and K. Mai, "Threshold voltage distribution in mlc nand flash memory: Characterization, analysis, and modeling," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2013, pp. 1285–1290.

[39] Y. Li, G. Han, S. Huang, C. Liu, M. Zhang, and F. Wu, "Exploiting metadata to estimate read reference voltage for 3-d nand flash memory," *IEEE Transactions on Consumer Electronics*, vol. 69, no. 1, pp. 9–17, 2022.

[40] M. Zhang, F. Wu, Q. Yu, W. Liu, Y. Wang, and C. Xie, "Exploiting error characteristic to optimize read voltage for 3-d nand flash memory," *IEEE Transactions on Electron Devices*, vol. 67, no. 12, pp. 5490–5496, 2020.

[41] M. Chun, J. Lee, M. Kim, J. Park, and J. Kim, "Rif: Improving read performance of modern ssds using an on-die early-retry engine," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024, pp. 643–656.

[42] Hynix, "FMS 2023: SK hynix Showcases World's First 321-Layer NAND," 8 2023. [Online]. Available: https://news.skhynix.com/sk-hynix-showcases-worlds-first-321-layer-nand-samples-storage-solutions-at-fms-2023/

[43] "SNIA - Storage Networking Industry Association: IOTTA Repository Home," 2007. [Online]. Available: http://iotta.snia.org/traces/block-io/388

[44] E. H. Wilson, M. Jung, and M. T. Kandemir, "Zombienand: Resurrecting dead nand flash for improved ssd longevity," in *2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*. IEEE, 2014, pp. 229–238.

[45] H.-Y. Liao, W.-L. Hsu, J.-W. Hsieh, and H.-P. Chen, "Read retry mechanism for 3d nand flash memory: Observations, analyses, and solutions," in *2024 13th Non-Volatile Memory Systems and Applications Symposium (NVMSA)*. IEEE, 2024, pp. 1–6.

[46] M. Kim, M. Chun, D. Hong, Y. Kim, G. Cho, D. Lee, and J. Kim, "Realwear: Improving performance and lifetime of ssds using a nand aging marker," *ACM SIGMETRICS Performance Evaluation Review*, vol. 48, no. 3, pp. 120–121, 2021.

[47] J. Cui, Z. Zeng, J. Huang, W. Yuan, and L. T. Yang, "Improving 3-d nand ssd read performance by parallelizing read-retry," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 42, no. 3, pp. 768–780, 2022.

[48] Q. Li, H. Dang, Z. Wan, C. Gao, M. Ye, J. Zhang, T.-W. Kuo, and C. J. Xue, "Midas touch: Invalid-data assisted reliability and performance boost for 3d high-density flash," in *2024 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 2024, pp. 657–670.

[49] D. Huang, D. Feng, Q. Liu, B. Ding, W. Zhao, X. Wei, W. Tong, S. Li, F. Zhu, M. Yuan, and J. Yang, "An efficient independent read scheme for contemporary qlc ssds," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 44, no. 5, pp. 1760–1773, 2025.