Sparsity in Linear Attention Models: Bregman Testing Time Learning

Anonymous ACL submission

Abstract

Linear attention architectures enable efficient long-context modeling by compressing sequence history into a fixed-size memory state. However, dense key value updates can overload the state as the context grows, causing memory collisions and degraded retrieval. Existing methods rely on memory decay, rank reduction, or heuristic routing to reduce interference, but they largely preserve dense online updates and do not explicitly control the structural complexity of memory. We introduce BregNet, a linear sequence architecture that formulates associative memory updates as a regularized online regression problem. Based on *Linearized Bregman Iteration* (LBI), BregNet decouples memory into a continuous dual accumulator and a structurally sparse primal state. The dual state records incoming information, while the sparse primal state keeps only important associations and removes noise through soft shrinkage. Experiments show that BregNet matches strong linear baselines in language modeling even with high memory sparsity. In this way, BregNet prevents the memory state from being overloaded by dense and interfering updates.

1 Introduction

Softmax attention (Vaswani et al., 2017) is highly expressive but scales quadratically in computation and memory with sequence length, making it costly for long-context modeling. Linear attention (Katharopoulos et al., 2020) addresses this bottleneck by replacing the explicit $d_v \times d_k$ attention matrix with recurrent formulations that run in linear time and support constant-memory inference: $\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t\mathbf{k}_t^\top$. These models maintain a matrix-valued hidden state $\mathbf{S}_t \in \mathbb{R}^{d_v \times d_k}$ as a compact associative memory for sequence history. However, the linear efficiency comes with a fundamental trade-off: the entire context must be compressed into a fixed-size state rather than represented through explicit pairwise token interactions. As the context grows, the recurrent state with bounded capacity must absorb new key–value associations while preserving history information that may be needed later. When the amount of stored information exceeds the intrinsic capacity of the state, newly written associations interfere with older ones, causing memory collisions (Schlag et al., 2021).

To overcome the limitation, recent work has sought to improve memory utilization through more selective update rules. An effective update rule should write useful associations, discard obsolete or noisy content, and preserve separable memory directions for future retrieval. Existing methods address this problem from different angles. Gated mechanisms (Yang et al., 2024b; Zhang et al., 2024; Team et al., 2025) introduce data-dependent decay to erase outdated information; delta-rule models (Yang et al., 2024c) perform targeted key–value replacement; rank-based (Nazari and Rusch, 2026) or top-K pruning (Hatamizadeh et al., 2026; Pan et al.) removes redundant key/query channels; and normalization-based methods (Yang et al., 2024a) reduce interference by constraining the magnitude. Although these methods differ in implementation, they point to the same underlying principle: long-term memory fidelity requires *selective write and forget mechanisms* that limit interference within the recurrent state.

However, most existing approaches still evolve through dense memory updates. Scalar decay (Sun et al., 2023; Qin et al., 2024; Dao and Gu, 2024) forgets information uniformly, post-hoc pruning (Nazari and Rusch, 2026) acts only after dense states have already formed and normalization (Yang et al., 2024b,a) mainly bounds update magnitude rather than controlling the structural complexity of memory. As a result, they do not provide a principled trajectory that induces sparsity during the sequential memory-writing process itself. This motivates a regularized online opti-

Table 1: Comparison of different linear attention models using the framework from (Yang et al., 2024a). For convenience, we use β_t to denote the step size of the update at time step t.

Method	Minimization Objective	Online Update
LA	$-\langle \mathbf{S}, \mathbf{v}_t\mathbf{k}_t^\top \rangle$	$\mathbf{S}_t = \mathbf{S}_{t-1} + \mathbf{v}_t\mathbf{k}_t^\top$
RetNet	$-\langle \mathbf{S}, \mathbf{v}_t\mathbf{k}_t^\top \rangle + \frac{1-\alpha_t}{2}\|\mathbf{S}\|_F^2$	$\mathbf{S}_t = \alpha_t\mathbf{S}_{t-1} + \mathbf{v}_t\mathbf{k}_t^\top$
DeltaNet	$\frac{1}{2}\|\mathbf{S}\mathbf{k}_t - \mathbf{v}_t\|_2^2$	$\mathbf{S}_t = \mathbf{S}_{t-1} - \beta_t(\mathbf{S}_{t-1}\mathbf{k}_t - \mathbf{v}_t)\mathbf{k}_t^\top$
Gated DeltaNet	$\frac{1}{2}\|\mathbf{S}\mathbf{k}_t - \mathbf{v}_t\|_2^2 + \frac{1-\alpha_t}{2\beta_t}\|\mathbf{S}\|_F^2$	$\mathbf{S}_t = \alpha_t\mathbf{S}_{t-1} - \beta_t(\mathbf{S}_{t-1}\mathbf{k}_t - \mathbf{v}_t)\mathbf{k}_t^\top$
BregNet	$\lambda\|\mathbf{S}\|_1 + \frac{1}{2}\|\mathbf{S}\|_F^2 \quad \text{s.t.} \quad \mathbf{S}\mathbf{k}_t = \mathbf{v}_t$	$\begin{cases} \mathbf{Z}_t = \mathbf{Z}_{t-1} - \beta_t(\mathbf{S}_{t-1}\mathbf{k}_t - \mathbf{v}_t)\mathbf{k}_t^\top, \\ \mathbf{S}_t = T_\lambda(\mathbf{Z}_t) \end{cases}$

mization view of recurrent memory, where dense overlapping updates are discouraged and salient associations are retained in sparse memory directions.

To overcome these optimization bottlenecks, we introduce BregNet, a linear sequence architecture grounded in Linearized Bregman Iteration (LBI) (Osher et al., 2010). BregNet formulates recurrent memory writing as a regularized online optimization process with two coupled states: a continuous dual accumulator $\mathbf{Z}_t$ and a sparse primal memory $\mathbf{S}_t$. The dual state acts as a write accumulator that integrates residual-correcting updates toward the associative constraint $\mathbf{S}_t\mathbf{k}_t \approx \mathbf{v}_t$, while the primal state is the filtered memory obtained by applying the soft-shrinkage operator $T_\lambda(\cdot)$ to the dual state, i.e., $\mathbf{S}_t = T_\lambda(\mathbf{Z}_t)$. This operator defines the selection rule, *forgetting* weak updates and *writing* strong memory directions. To enable chunk-wise parallelization, BregNet adopts a lazy-update strategy: it accumulates dual updates in matrix form within each chunk and applies soft shrinkage only at chunk boundaries, equivalent to one LBI step on the chunk-level loss. Table 1 compares BregNet with representative linear recurrent models from the perspective of optimization objectives and memory update rules.

Our resulting architecture, BregNet, improves linear attention across a comprehensive suite of benchmarks, including language modeling, commonsense reasoning, and structured information retrieval. Across 340M and 1.3B models trained under matched settings, BregNet achieves competitive perplexity with strong linear baselines like Gated DeltaNet, while maintaining 20% to 50% average sparsity in the memory matrix. These results show that LBI-induced sparse memory updates provide an effective mechanism for improving long-context retrieval while preserving the efficiency of recurrent linear attention.

2 Background

Standard softmax attention (Vaswani et al., 2017) models sequence interactions through a query–key–value retrieval mechanism. Given an input sequence of length L, the model forms queries $\mathbf{Q} \in \mathbb{R}^{L\times d_k}$, keys $\mathbf{K} \in \mathbb{R}^{L\times d_k}$, and values $\mathbf{V} \in \mathbb{R}^{L\times d_v}$. Each query specifies what information a token seeks, keys index the information stored by tokens, and values contain the corresponding content. The attention output is computed as

$$\mathrm{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}.$$

This formulation explicitly computes all pairwise query–key interactions, leading to quadratic computation and memory complexity with respect to L.

2.1 Linear Attention

Linear attention (Katharopoulos et al., 2020) addresses this bottleneck by replacing softmax attention with a kernelized dot-product formulation. For simplicity, we present the vanilla form without normalization or query/key nonlinearities. Let $\mathbf{q}_t, \mathbf{k}_t \in \mathbb{R}^{d_k}$ and $\mathbf{v}_t \in \mathbb{R}^{d_v}$ denote the query, key, and value at time step t. Linear attention can be written as the recurrent update, noted as LA's online update in Table. 1, where $\mathbf{S}_t \in \mathbb{R}^{d_v\times d_k}$ is a matrix-valued hidden state that stores past key–value associations. This recurrent formulation enables constant-memory streaming inference and linear-time sequence processing.

Recent linear attention models extend the vanilla recurrence by introducing data-dependent decay gates. For example, Mamba2 (Dao and Gu, 2024) uses a scalar-valued gate to decay the recurrent state, while RetNet (Sun et al., 2023) and Light-

ning Attention (Qin et al., 2024) adopt related decay mechanisms. A common simplified form is RetNet's online update form in Table. 1 where $\alpha_t \in (0, 1)$ controls how much past memory is retained. Such gates provide a simple forgetting mechanism, but they typically decay the state uniformly rather than selectively modifying specific key–value associations.

Another perspective on linear sequence modeling is test-time regression. DeltaNet (Yang et al., 2024c; Schlag et al., 2021) uses the delta rule to remove the old value associated with the current key before writing a new one (Table. 1, DeltaNet), where β_t controls the update strength. Gated DeltaNet (Yang et al., 2024a) further combines the delta rule with a data-dependent decay gate (Table. 1, Gated DeltaNet). This progression from additive accumulation to gated and regression-based state updates is a key mechanism behind modern high-performance linear attention architectures.

2.2 Linearized Bregman Iteration

Linearized Bregman Iteration (LBI) (Osher et al., 2010) was originally developed for computing sparse or low-rank solutions to underdetermined linear inverse problems. A canonical example is basis pursuit, which seeks a regularized solution under a linear constraint: $\min_{\mathbf{s}\in\mathbb{R}^n} \ \omega(\mathbf{s}) \quad \text{s.t.} \quad \mathbf{A}\mathbf{s} = \mathbf{f}$.

LBI solves such problems through a two-step iterative scheme that couples a dual variable $\mathbf{z}_t$ with a primal variable $\mathbf{s}_t$:

$$\text{Dual update:} \qquad \mathbf{z}_{t+1} = \mathbf{z}_t - \beta \nabla \ell(\mathbf{s}_t),$$

$$\text{Primal update:} \qquad \mathbf{s}_{t+1} = \nabla \omega^*(\mathbf{z}_{t+1}),$$

where ℓ denotes the data-fidelity loss, β is the step size, and ω^* is the convex conjugate of the regularizer ω. The dual variable accumulates descent information toward the constraint, while the primal variable is obtained through the regularized mapping induced by ω^*.

The key mathematical tool behind LBI is the Bregman distance, which measures discrepancy under the geometry of a convex regularizer. For a convex function J, the Bregman distance between $\mathbf{z}$ and $\mathbf{s}$ with respect to a subgradient $\mathbf{p} \in \partial J(\mathbf{z})$ is defined as

$$D_J^{\mathbf{p}}(\mathbf{s}, \mathbf{z}) = J(\mathbf{s}) - J(\mathbf{z}) - \langle \mathbf{s} - \mathbf{z}, \mathbf{p} \rangle.$$

This geometry allows LBI to approach the feasible region while regularizing the optimization path, often producing sparse solutions before full convergence.

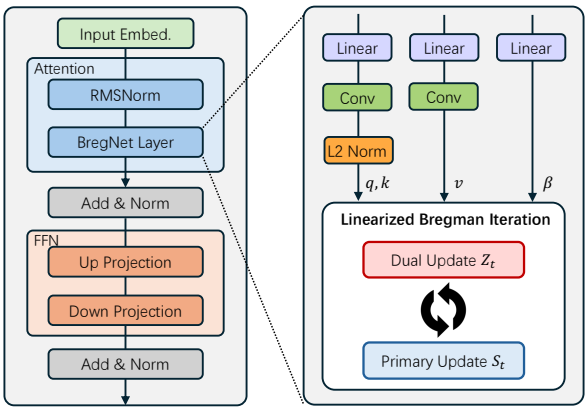

Figure 1: Model Structure for BregNet.

Prior work has developed LBI from basic ℓ_1-minimization (Cai et al., 2009a) to broader structural sparsity frameworks (Cai et al., 2009b; Osher et al., 2010; Yin, 2010). For structured problems such as total variation and graph-regularized models, Split LBI introduces auxiliary variables whose dynamics are closely related to Inverse Scale Space dynamics. With early stopping, Split LBI has been shown to provide robust regularization and model-selection consistency under irrepresentable conditions weaker than those required by the generalized Lasso. More recent work further abstracts LBI into a general algorithmic framework for bilevel optimization and inverse problems, including non-Gaussian noise models (Huang et al., 2020; Dai et al., 2024). These analyses establish convergence guarantees under Bregman-distance growth conditions, including sublinear and linear rates for different LBI variants.

3 Method

3.1 Recurrent Updating Rule

To incorporate LBI into linear attention, we propose BregNet, which seeks a sparse associative memory by solving the following elastic-net-like constrained optimization problem:

$$\min_{\mathbf{S}} \quad \lambda \|\mathbf{S}\|_1 + \frac{1}{2}\|\mathbf{S}\|_F^2 \quad \text{s.t.} \quad \mathbf{S}\mathbf{k}_t = \mathbf{v}_t. \tag{1}$$

Under the LBI framework, Eq. 1 is approximated by a two-step iterative process with a dual variable $\mathbf{Z}_t$ and a primal variable $\mathbf{S}_t$.

Dual update. Given the previous primal memory $\mathbf{S}_{t-1}$, the dual update performs gradient descent

on the data-fidelity loss $L_t(\mathbf{S}) = \frac{1}{2}\|\mathbf{v}_t - \mathbf{S}\mathbf{k}_t\|_2^2$ by accumulating its negative gradient in the dual variable:

$$\mathbf{Z}_t = \mathbf{Z}_{t-1} - \beta_t(\mathbf{S}_{t-1}\mathbf{k}_t - \mathbf{v}_t)\mathbf{k}_t^\top. \tag{2}$$

Thus, $\mathbf{Z}_t$ acts as a residual-correcting accumulator that drives the associative memory toward the constraint $\mathbf{S}\mathbf{k}_t = \mathbf{v}_t$.

Primal update. The primal update maps the dual variable back to the regularized memory space by solving

$$\mathbf{S}_t = \arg\min_{\mathbf{S}} \left\{ \lambda\|\mathbf{S}\|_1 + \frac{1}{2}\|\mathbf{S} - \mathbf{Z}_t\|_F^2 \right\}. \tag{3}$$

This proximal problem has the closed-form solution

$$\mathbf{S}_t = T_\lambda(\mathbf{Z}_t) = \operatorname{sgn}(\mathbf{Z}_t) \odot \max\big(|\mathbf{Z}_t| - \lambda, 0\big), \tag{4}$$

where $T_\lambda(\cdot)$ denotes the element-wise soft-shrinkage operator. A detailed proof is provided in § C.4.

This dual–primal update gives BregNet an explicit write-and-select mechanism. The dual state $\mathbf{Z}_t$ accumulates residual-correcting evidence, including weak or noisy signals, while the primal state $\mathbf{S}_t$ retains only entries that pass the threshold λ. In this sense, $T_\lambda(\cdot)$ acts as the selection rule: weak or inconsistent updates are suppressed to zero, whereas strong and persistent memory directions remain active.

3.2 Chunkwise Implementation

The goal of chunkwise implementation is to replace token-by-token recurrent updates within a chunk with matrix-level computation. Ideally, the model should advance from the initial state of a chunk to its final state using parallelizable operations, rather than executing a sequential loop over tokens.

For token-wise BregNet, however, directly applying LBI at every token makes exact chunkwise parallelization difficult. Substituting the primal update into the dual update introduces an unfactorized transition:

$$\mathbf{Z}_t = \underbrace{\mathbf{Z}_{t-1} - \beta_t(\mathbf{M}_{t-1} \odot \mathbf{Z}_{t-1})\mathbf{k}_t\mathbf{k}_t^\top}_{\text{state-dependent transition}} + \underbrace{\beta_t\mathbf{v}_t\mathbf{k}_t^\top}_{\text{write term}}.$$

where $\mathbf{M}_t$ is the mask induced by the soft-shrinkage operator in Eq. 4. The application of $\mathbf{M}_t$ introduces three computational obstacles for chunkwise parallelization:

i. $\mathbf{M}_t$ depends on the dynamic state $\mathbf{Z}_t$, rather than only on the input tokens;

ii. the operator $T_\lambda(\cdot)$ introduces a non-smooth nonlinearity into the recurrence;

iii. the Hadamard product does not generally factorize through standard matrix multiplication, e.g., $(\mathbf{A} \odot \mathbf{M}_t)\mathbf{B} \neq \mathbf{A}(\mathbf{M}_t \odot \mathbf{B})$.

The row-wise algorithm in § C.5 is equivalent to the exact recurrent rule, but it does not expose sufficient matrix-level computation for efficient training. To reduce overhead, we adopt a *lazy-update strategy*: within each chunk, the primal memory is kept fixed, the dual updates are accumulated in matrix form, and soft shrinkage is applied only at the chunk boundary. We further prove that this lazy variant preserves sublinear convergence in terms of the Bregman divergence.

Let the chunked objective be

$$J_C(\mathbf{S}) = \frac{1}{2}\sum_{i=1}^{C} \|\mathbf{v}_{t_i} - \mathbf{S}\mathbf{k}_{t_i}\|_2^2, \tag{5}$$

$$\nabla J_C(\mathbf{S}) = \sum_{i=1}^{C} (\mathbf{S}\mathbf{k}_{t_i} - \mathbf{v}_{t_i})\,\mathbf{k}_{t_i}^\top. \tag{6}$$

Under the lazy update, the token-wise updates within a chunk are compressed into a standard LBI step at the chunk level, where m denotes the chunk index.

Dual update (chunked).

$$\mathbf{Z}_m = \mathbf{Z}_{m-1} - \beta\nabla J_C(\mathbf{S}_{m-1}), \tag{7}$$

$$= \mathbf{Z}_{m-1} - \mathbf{S}_{m-1}\mathbf{K}^\top\mathbf{B}\mathbf{K} + \mathbf{V}^\top\mathbf{B}\mathbf{K}. \tag{8}$$

Primal update (chunked). After accumulating the dual update over the whole chunk, BregNet applies soft shrinkage once at the chunk boundary:

$$\mathbf{S}_m = T_\lambda(\mathbf{Z}_m) \tag{9}$$

$$= \operatorname{sgn}(\mathbf{Z}_m) \odot \max\big(|\mathbf{Z}_m| - \lambda, 0\big). \tag{10}$$

Although the lazy update seems to be simple, it is mathematically well motivated: keeping the primal variable $\mathbf{S}_{m-1}$ fixed within a chunk and accumulating the dual updates over its C tokens is equivalent to performing one LBI step on the sum of the corresponding convex losses.

4 Convergence Rate and Sparsity Tuning

BregNet adopts an indirect optimization mechanism that incorporates ℓ_1 sparsity and ℓ_2 shrinkage through a two-step primal–dual update. This section addresses two core questions of this indirect optimization:

i. Convergence: Do the chunked iterates converge to the optimal solution of the underlying regularized objective?

ii. Scaling law for λ: To suppress false discoveries in the memory matrix $\mathbf{S}$, how should the threshold parameter λ scale as training proceeds?

4.1 Preliminaries

Let the chunked objective be

$$J_C(\mathbf{S}) = \frac{1}{2}\sum_{i=1}^{C}\left\|\mathbf{v}_{t_i} - \mathbf{S}\mathbf{k}_{t_i}\right\|_2^2,$$

with an optimal solution $\bar{\mathbf{S}}$ satisfying $\nabla J_C(\bar{\mathbf{S}}) = 0$. Let L_C denote the Lipschitz constant of ∇J_C, e.g., $L_C = \left\|\sum_{i=1}^{C}\mathbf{k}_{t_i}\mathbf{k}_{t_i}^\top\right\|_2$.

Consider the chunked LBI updates

$$\mathbf{Z}_m = \mathbf{Z}_{m-1} - \beta\nabla J_C(\mathbf{S}_{m-1}), \tag{11}$$

$$\mathbf{S}_m = \nabla\omega^*(\mathbf{Z}_m), \tag{12}$$

where $\omega(\mathbf{S}) = \lambda\|\mathbf{S}\|_1 + \frac{1}{2}\|\mathbf{S}\|_F^2$ is strongly convex with modulus 1, and ω^* is its convex conjugate.

4.2 Sublinear Convergence for Chunked Update

Proposition 1 and Theorem 2 require a step-size condition on β determined by L_C. This condition clarifies why modern architectures often apply ℓ_2-norm scaling to the key matrix in Fig. 1: such scaling controls L_C and helps guarantee monotonic descent in Bregman divergence.

Proposition 1 (Monotonic Bregman Descent for Chunked Update)**.** *If the step size satisfies*

$$\beta \le \frac{2}{L_C},$$

then the Bregman distance $D_\omega^{\mathbf{Z}}(\cdot,\cdot)$ to the optimum is monotonically non-increasing and obeys the one-step descent bound:

$$\begin{aligned} D_\omega^{\mathbf{Z}_m}(\bar{\mathbf{S}},\mathbf{S}_m) &\le D_\omega^{\mathbf{Z}_{m-1}}(\bar{\mathbf{S}},\mathbf{S}_{m-1}) + \left(\frac{\beta^2}{2} - \frac{\beta}{L_C}\right) \\ &\quad \cdot \|\nabla J_C(\mathbf{S}_{m-1})\|_F^2 \\ &\le D_\omega^{\mathbf{Z}_{m-1}}(\bar{\mathbf{S}},\mathbf{S}_{m-1}). \end{aligned} \tag{13}$$

Theorem 2 (Sublinear $O(1/T)$ Convergence for Chunked Update)**.** *Given that ∇J_C is L_C-Lipschitz continuous, if the step size satisfies $\beta \le \frac{1}{L_C}$, then for all $T \ge 0$,*

$$J_C(\mathbf{S}_{T+1}) - J_C(\bar{\mathbf{S}}) \le \frac{D_\omega^{\mathbf{Z}_0}(\bar{\mathbf{S}},\mathbf{S}_0)}{\beta(T+1)}.$$

Because the dual gradients are accumulated over C tokens, the effective Lipschitz constant L_C increases. As a result, the recurrent single-token step size β should be reduced by a factor of up to C to prevent divergence.

4.3 Early Stopping for Chunked Update

This subsection studies how to avoid spurious sparsity, known as *false positives* in sparse recovery for $\mathbf{S}$ (Osher et al., 2016). Equivalently, we aim to keep irrelevant features inactive and derive a scaling rule for choosing λ across training steps.

Let the ground-truth parameter be $\mathbf{S}^*$ with support $\mathcal{S}$, where $\mathbf{S}^*_{\mathcal{S}^c} = 0$. Assume the observation noise $\boldsymbol{\epsilon}_{t_i} = \mathbf{v}_{t_i} - \mathbf{S}^*\mathbf{k}_{t_i}$ is independent, zero-mean, and sub-Gaussian with variance proxy σ^2. Define the oracle sequence $\mathbf{S}'_m$ as the LBI updates restricted to the true support $\mathcal{S}$, enforcing $\mathbf{S}'_{m,\mathcal{S}^c} = 0$ for all m.

The dual variable integrates the chunked gradients over m iterations and can be decomposed into signal integration $\mathbf{I}_m$ and noise integration $\mathbf{W}_m$. Substituting $\mathbf{v}_{t_i} = \mathbf{S}^*\mathbf{k}_{t_i} + \boldsymbol{\epsilon}_{t_i}$, the dual variable evaluated along the oracle sequence is

$$\mathbf{Z}_m = \mathbf{Z}_0 - \beta\sum_{j=0}^{m-1}\sum_{i=1}^{C}(\mathbf{S}'_j\mathbf{k}_{t_{j,i}} - \mathbf{v}_{t_{j,i}})\mathbf{k}_{t_{j,i}}^\top.$$

$$\begin{aligned} \mathbf{Z}_m = &\underbrace{-\beta\sum_{j=0}^{m-1}\sum_{i=1}^{C}(\mathbf{S}'_j - \mathbf{S}^*)\mathbf{k}_{t_{j,i}}\mathbf{k}_{t_{j,i}}^\top}_{\text{Signal Integration }(\mathbf{I}_m)} \\ &+ \underbrace{\beta\sum_{j=0}^{m-1}\sum_{i=1}^{C}\boldsymbol{\epsilon}_{t_{j,i}}\mathbf{k}_{t_{j,i}}^\top}_{\text{Noise Integration }(\mathbf{W}_m)}. \end{aligned}$$

Theorem 3 (Early Stopping Bound for No-False–Positive Path Consistency)**.** *Assume the inputs are bounded: $\|\mathbf{k}_{t_i}\|_2 \le K$. If the Irrepresentable Condition (IRR) holds with margin $\eta \in (0,1]$,*

$$\|\mathbf{I}_{m,\mathcal{S}^c}\|_\infty \le (1-\eta)\lambda,$$

then a false positive cannot occur provided

$$\|\mathbf{W}_{m,\mathcal{S}^c}\|_\infty < \eta\lambda.$$

Moreover, for any $\delta \in (0,1)$*, ensuring*

$$\mathbb{P}(\|\mathbf{W}_{m,\mathcal{S}^c}\|_\infty \geq \eta\lambda) \leq \delta$$

is sufficient to guarantee path consistency up to the stopping time

$$M_{\text{stop}} = \left\lfloor \frac{\eta^2\lambda^2}{2\beta^2\sigma^2K^2\log(2d/\delta)} \cdot \frac{1}{C} \right\rfloor,$$

where d *denotes the dimension of the non-support space.*

Scaling law for λ. To identify the true support $\mathcal{S}$ and keep the non-support parameters $\mathcal{S}^c$ exactly zero, Theorem 3 requires the dual variable on the non-support space to remain below the ℓ_1 threshold, i.e., $\|\mathbf{Z}_{m,\mathcal{S}^c}\|_\infty < \lambda$. Equivalently, bounding $\mathbb{P}(\|\mathbf{W}_{m,\mathcal{S}^c}\|_\infty \geq \eta\lambda)$ yields the following scaling law for λ, with details provided in § C.3:

$$\lambda \;\propto\; \beta\sqrt{mC}\sqrt{\log\left(\frac{d}{\delta}\right)}. \tag{14}$$

5 Experiment

Experimental Setup Our experiment compares standard Transformer models, linear attention models and hybrid architectures. We evaluate on common sense reasoning and context recall tasks, including LAMBADA (Paperno et al., 2016), PiQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-easy (ARC-e) and ARC-challenge (Arc-c) (Clark et al., 2018). Furthermore, we also evaluate the models real-world recall-intensive tasks following (Arora et al., 2024), including FDA (Arora et al., 2023), SWDE (Lockard et al., 2019). Both SWDE and FDA focus on extracting structured information: SWDE from raw HTML to identify semi-structured relationships, and FDA from PDFs to retrieve key-value pairs. All evaluations are performed with `lm-evaluation-harness` (Gao et al., 2021). See § A for hyperparameter setting.

Common-sense reasoning. In Table 1, we present the language modeling perplexity and zero-shot accuracy on common-sense reasoning benchmarks for models scaled to 340M parameters, trained on 15B tokens. Gated DeltaNet establishes a strong perplexity baseline on Wiki and LAMBADA. However, the BregNet (w/ gate) architecture demonstrates superior zero-shot transfer capabilities, matching or exceeding all baseline models on LAMBADA perplexity task, while maintaining comparable zero-shot reasoning accuracy across COPA and SWDE, validating that the underlying Bregman dynamics successfully manage sequence state tracking at this scale.

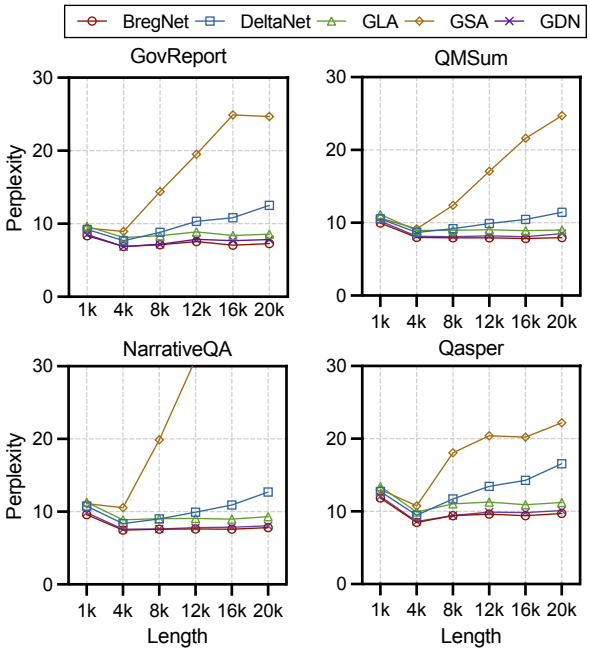

Figure 2: Length extrapolation on four long benchmarks.

Scaling to downstream tasks. Table. 3 extends our evaluation to the 1B parameter scale. Consistent with the 340M results, BregNet exhibits robust generalization, achieving the lowest perplexity on LAMBADA in pure linear models. Notably, BregNet scales highly effectively, outperforming the baseline DeltaNet by a significant margin in language modeling perplexity. Furthermore, BregNet tightly matches the performance of Gated DeltaNet on key reasoning benchmarks such as PIQA and WinoGrande, indicating that the projection mechanisms governing the state updates do not bottleneck broader semantic understanding.

Long-context ability. To better understand the memory retention and associative recall capabilities of these architectures, we evaluate the models on the Single Needle-In-A-Haystack (S-NIAH) and length extrapolation tasks. As shown in Table. 4, models were trained with a 4K context window and evaluated on sequences up to 8K tokens. BregNet

Table 2: Commonsense reasoning and language modeling results for 340M-parameter models trained on 15B tokens, with batch size of 0.1M tokens. All the models are trained under the same SlimPajama dataset. BregNet (*w/o gate*) is the vanilla variant built on top of DeltaNet, while (*w/ gate*) incorporates the same gating mechanism as Gated DeltaNet.

Model	Sparsity	LM		Commonsense Reasoning									Context Retrieval		Avg
		Wiki ppl↓	LMB ppl↓	LMB acc	PIQA acc	Hella acc_n	Wino acc	ARC-e acc	ARC-c acc_n	SciQ acc	BoolQ acc	COPA acc	FDA contains	SWDE contains	
Transformer	–	27.2	73.1	26.0	65.3	36.0	51.4	45.8	23.9	78.2	52.6	69.0	**19.3**	**49.8**	**47.0**
Mamba2	–	**26.4**	55.4	26.8	**66.3**	**37.7**	51.5	**48.1**	**24.7**	78.2	56.4	**70.0**	13.0	40.9	46.7
DeltaNet	–	27.5	67.2	24.7	64.8	35.2	49.3	46.3	24.5	77.1	61.0	68.0	11.8	31.6	44.9
Gated DeltaNet	–	**26.4**	55.5	27.6	65.2	36.5	**52.4**	45.9	23.6	78.7	57.0	65.0	9.4	30.8	44.8
BregNet (*w/o gate*)	50%	27.1	63.2	26.2	65.3	36.0	50.4	45.4	23.7	**79.4**	**61.3**	67.0	11.6	29.9	45.1
(*w/ gate*)	20%	26.5	**48.5**	**30.1**	65.7	36.4	51.4	46.2	24.0	77.9	57.9	69.0	8.2	34.0	45.5

Table 3: Commonsense reasoning, language modeling, and context retrieval performance of 1.3B-parameter models trained with 100B tokens, batch size of 0.5M tokens. GLA, GSA, Gated DeltaNet, and BregNet variants are trained under the same setting on the SlimPajama dataset. The hybrid H2 setting, including the 3:1 MLA ratio and Gated DeltaNet-H2 results, follows (Yang et al., 2024c). "–" indicates not applicable or not evaluated.

Model	Sparsity	LM		Commonsense Reasoning								Context Retrieval		Avg.
		Wiki ppl↓	LMB ppl↓	LMB acc	PIQA acc	Hella acc_n	Wino acc	ARC-e acc	ARC-c acc_n	SciQ acc	COPA acc	FDA contains	SWDE contains	
Transformer	–	17.7	18.6	41.1	69.9	49.0	55.0	56.0	28.3	85.1	**76.0**	17.1	65.8	54.3
Pure linear models														
GLA	–	17.7	19.7	40.3	69.9	48.9	53.2	55.4	27.5	80.5	70.0	12.0	43.0	50.1
GSA	–	16.8	15.7	41.8	72.1	50.9	52.6	58.7	28.5	83.0	73.0	11.3	45.5	51.8
DeltaNet	–	16.7	14.7	42.9	71.0	50.5	53.8	58.6	26.2	84.8	74.0	17.7	54.6	53.4
Gated DeltaNet	–	15.6	11.2	47.6	**73.1**	55.6	57.4	59.9	29.7	87.4	**76.0**	22.7	62.5	57.2
BregNet (*w/o gate*)	50%	15.8	10.9	47.8	72.9	55.1	**58.7**	61.2	29.0	87.5	**76.0**	24.2	58.2	57.1
(*w/ gate*)	20%	**15.3**	**10.2**	**49.3**	72.9	56.1	57.3	60.1	30.1	**88.1**	75.0	24.2	61.6	57.5
Hybrid models														
BregNet-H2 (*w/o gate*)	50%	15.8	11.1	48.1	72.2	54.7	57.8	60.0	29.4	87.6	**76.0**	**27.2**	**69.2**	**58.2**
GDN-H2	–	15.9	12.6	48.8	72.2	**56.9**	57.8	**71.3**	**39.1**	–	61.6	23.7	25.4	–

demonstrates exceptional, near-perfect associative recall up to its training length, achieving 100.0% accuracy on both the pass-key (S-NIAH-1) and numerical (S-NIAH-2) retrieval tasks at 4K lengths.

Crucially, on the more challenging S-NIAH-2 task, where target values are replaced by numbers, BregNet demonstrates stronger retention capacity at 8K context length, significantly outperforming both Mamba2 and Gated DeltaNet. Although performance naturally degrades when extrapolating to 4K and 8K sequences, BregNet remains relatively stable at 4K, suggesting that its localized updates help mitigate the catastrophic memory collisions observed in standard linear recurrent models. Figure 2 further reports perplexity under length extrapolation up to 20K tokens. The results again show that BregNet consistently achieves the best perplexity across extended context lengths.

λ Scheduling. We first compare different strategies for applying the threshold in Table 5. Scaling Law follows Eq. 14 by adapting the effective threshold using the β statistics during each forward pass, while Fixed uses a constant threshold. We find that Fixed achieves comparable overall performance

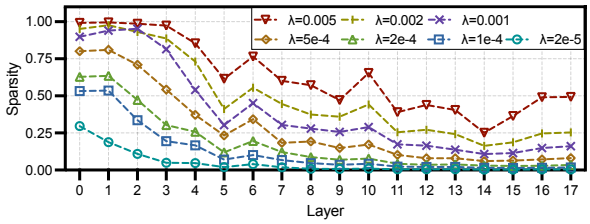

Figure 3: Sparsity level as a function of the λ regularization strength.

to Scaling Law and layer-wise schedules, while being simpler to implement and faster to execute. Therefore, we use the Fixed strategy in subsequent experiments.

Sparsity. We vary the sparsity level of primal state $\mathbf{S}$ by changing the fixed threshold λ, as it allows easier implementation and faster execution. Table 6 shows that BregNet remains stable as the maximum layer-wise sparsity increases from 30% to 81%. The 54% sparsity configuration achieves the best overall performance. Figure 3 illustrates the sequence-level sparsity distributions generated by the competing models. Unlike standard linear attention or pure delta rule mechanisms, BregNet

Table 4: Associative recall accuracy (%) on S-NIAH under increasing context length. Models are trained on the same setting with context length of 4k. Results for Mamba2 are taken from (Yang et al., 2024c).

Model	S-NIAH-1				S-NIAH-2				S-NIAH-3		
	1K	2K	4K	8K	1K	2K	4K	8K	1K	2K	4K
GLA	99.8	88.6	16.4	2.0	100.0	99.6	60.2	19.4	75.4	73.0	17.0
GSA	100.0	100.0	99.8	93.6	100.0	100.0	84.0	9.2	72.0	64.0	24.8
Mamba2	99.2	98.8	65.4	30.4	99.4	98.8	56.2	17.0	64.4	47.6	4.6
DeltaNet	100.0	100.0	100.0	100.0	100.0	100.0	67.8	19.2	86.2	74.0	59.6
Gated DeltaNet	100.0	100.0	100.0	100.0	100.0	100.0	98.2	28.4	**99.8**	99.2	**75.0**
BregNet (*w/o gate*)	100.0	100.0	90.0	92.2	100.0	100.0	**98.4**	31.2	**99.8**	**99.4**	61.8
(*w/ gate*)	100.0	100.0	100.0	100.0	100.0	100.0	96.0	**67.8**	89.6	99.2	59.6

Table 5: Threshold-scheduling comparison for 340M BregNet (*w/o gate*) with 50% target mean sparsity. Scaling Law follows Eq. 14; Fixed uses a constant threshold of 0.01; L2S and S2L schedule the threshold from large to small and small to large, respectively. DeltaNet is included as the baseline. Common Avg. and Retrieval Avg. are averaged over the commonsense reasoning and context retrieval tasks in Table 2.

λ Type	Wiki PPL↓	LMB PPL↓	Common Avg.↑	Retrieval Avg.↑
Scaling Law	27.3	**61.5**	**50.6**	19.3
Fixed	**27.1**	63.2	50.5	20.7
L2S	**27.1**	63.3	50.4	19.2
S2L	27.4	70.3	50.0	17.4
DeltaNet	27.4	67.2	50.1	**21.7**

Table 6: Performance under different memory sparsity levels for 340M BregNet (*w/ gate*).

Sparsity (max)	Sparsity (avg)	Wiki PPL↓	LMB PPL↓	Common Avg.↑	Retrieval Avg.↑
30%	5%	**26.3**	48.8	51.3	20.3
54%	13%	26.4	**45.7**	**51.4**	20.1
64%	20%	26.5	48.5	50.9	**21.1**
81%	28%	26.5	47.7	50.3	20.4
95%	39%	26.5	52.6	50.9	19.5
98%	48%	26.7	51.2	51.2	17.9

uniquely induces high structural sparsity within its hidden states. Because the update dynamics are implicitly governed by the soft-shrinkage operator $T_\lambda(x) = \text{sign}(x)\max(|x| - \lambda, 0)$, the model dynamically masks out redundant or noisy historical tokens, storing only the most critical key-value representations. This sparse state footprint directly mitigates information superimposition, which accounts for BregNet's superior performance in the high-collision environments of the S-NIAH evaluations, and long benchmark's extrapolation.

6 Conclusion

In this work, we introduce BregNet, a linear sequence architecture designed to improve state saturation and memory overloading in standard linear recurrent models. We formulate associative memory updates as a regularized online regression problem and leverage *Linearized Bregman Iteration* (LBI) to decouple the hidden state into a continuous dual accumulator and a structurally sparse primal state. Empirical results at the 340M and 1B parameter scales show that BregNet outperforms state-of-the-art linear baselines on language modeling, demonstrating that the induced structural sparsity effectively preserves memory quality. These results highlight regularized memory updates as a key mechanism for scalable sequence modeling and future infinite-context architectures.

Limitations

BregNet targets scalable long-context sequence modeling, but our current evaluation only provides an initial validation of this goal. Specifically, our experiments focus on 340M and 1.3B models across language modeling, reasoning, and retrieval benchmarks, so larger-scale pretraining and broader long-context evaluations are still needed to fully assess its scalability and generality. As BregNet is scaled to larger models and longer contexts, its memory dynamics may also become more sensitive to implementation and regularization choices. In particular, the sparsity threshold λ controls the trade-off between memory sparsity and task performance, and may require scale-specific tuning despite being stable within a reasonable range. Meanwhile, the chunk size determines how closely the hardware-efficient lazy shrinkage approximates token-wise LBI dynamics, making its impact on sparsity evolution and retrieval quality an important direction for further study.

References

Simran Arora, Sabri Eyuboglu, Michael Zhang, Aman Timalsina, Silas Alberti, Dylan Zinsley, James Zou, Atri Rudra, and Christopher Ré. 2024. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv preprint arXiv:2402.18668*.

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language models enable simple systems for generating structured views of heterogeneous data lakes. *arXiv preprint arXiv:2304.09433*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Jian-Feng Cai, Stanley Osher, and Zuowei Shen. 2009a. Convergence of the linearized bregman iteration for l1-norm minimization. *Mathematics of Computation*, 78(268):2127–2136.

Jian-Feng Cai, Stanley Osher, and Zuowei Shen. 2009b. Linearized bregman iterations for compressed sensing. *Mathematics of computation*, 78(267):1515–1536.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Yu-Hong Dai, Kangkang Deng, and Hui Zhang. 2024. A cut-and-project perspective for linearized bregman iterations. *arXiv preprint arXiv:2404.09776*.

Tri Dao and Albert Gu. 2024. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, and 1 others. 2021. A framework for few-shot language model evaluation. *Zenodo*.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. 2026. Gated deltanet-2: Decoupling erase and write in linear attention. *arXiv preprint*.

Chendi Huang, Xinwei Sun, Jiechao Xiong, and Yuan Yao. 2020. Boosting with structural sparsity: A differential inclusion approach. *Applied and Computational Harmonic Analysis*, 48(1):1–45.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR.

Colin Lockard, Prashant Shiralkar, and Xin Luna Dong. 2019. Openceres: When open information extraction meets the semi-structured web. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3047–3056.

Philipp Nazari and T Konstantin Rusch. 2026. The key to state reduction in linear attention: A rank-based perspective. *arXiv preprint arXiv:2602.04852*.

Stanley Osher, Yu Mao, Bin Dong, and Wotao Yin. 2010. Fast linearized bregman iteration for compressive sensing and sparse denoising.

Stanley Osher, Feng Ruan, Jiechao Xiong, Yuan Yao, and Wotao Yin. 2016. Sparse recovery via differential inclusions. *Applied and Computational Harmonic Analysis*, 41(2):436–469.

Yuqi Pan, Yongqi An, Zheng Li, Yuhong Chou, Rui-Jie Zhu, Xiaohui Wang, Mingxuan Wang, Jinqiao Wang, and Guoqi Li. Scaling linear attention capacity with sparse state expansion. In *The Fourteenth International Conference on Learning Representations*.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1525–1534.

Zhen Qin, Weigao Sun, Dong Li, Xuyang Shen, Weixuan Sun, and Yiran Zhong. 2024. Lightning attention-2: A free lunch for handling unlimited sequence lengths in large language models. *arXiv preprint arXiv:2401.04658*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pages 9355–9366. PMLR.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. 2023. Retentive network: A successor to transformer for large language models. *arXiv preprint arXiv:2307.08621*.

Kimi Team, Yu Zhang, Zongyu Lin, Xingcheng Yao, Jiaxi Hu, Fanqing Meng, Chengyin Liu, Xin Men, Songlin Yang, Zhiyuan Li, and 1 others. 2025. Kimi linear: An expressive, efficient attention architecture. *arXiv preprint arXiv:2510.26692*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. 2024a. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. 2024b. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning*.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. 2024c. Parallelizing linear transformers with the delta rule over sequence length. *Advances in neural information processing systems*, 37:115491–115522.

Wotao Yin. 2010. Analysis and generalizations of the linearized bregman method. *SIAM Journal on Imaging Sciences*, 3(4):856–877.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.

Yu Zhang, Songlin Yang, Ruijie Zhu, Yue Zhang, Leyang Cui, Yiqiao Wang, Bolun Wang, Freda Shi, Bailin Wang, Wei Bi, and 1 others. 2024. Gated slot attention for efficient linear-time sequence modeling. *Advances in Neural Information Processing Systems*, 37:116870–116898.

Table 7: Sensitivity analysis of the static threshold λ for the 340M BregNet (*w/ gate*) model with 2K-token inputs. We vary a fixed λ and report the resulting maximum layer-wise sparsity across layers.

λ	Sparsity (max)	LM		Commonsense Reasoning									Context Retrieval
		Wiki ppl↓	LMB ppl↓	LMB acc	PIQA acc	Hella acc_n	Wino acc	ARC-e acc	ARC-c acc_n	SciQ acc	BoolQ acc	COPA acc	SWDE contains
2×10^{-5}	29.7%	26.3	48.8	29.5	65.5	36.9	49.6	46.8	24.7	76.8	61.3	71.0	31.2
1×10^{-4}	53.5%	26.4	45.7	29.5	65.5	36.8	53.4	46.8	24.2	80.6	59.8	66.0	32.9
2×10^{-4}	63.5%	26.5	48.5	30.2	65.5	36.4	51.3	45.9	24.2	77.7	57.8	69.0	33.6
5×10^{-4}	81.0%	26.5	47.7	29.7	64.9	36.9	51.6	47.3	24.7	78.6	47.8	71.0	32.9
1×10^{-3}	95.3%	26.5	52.6	28.6	65.3	36.5	51.0	45.5	24.7	78.5	58.4	70.0	31.8
2×10^{-3}	97.7%	26.7	51.2	28.3	65.3	36.1	51.1	45.8	22.8	78.6	60.4	72.0	28.3

Table 8: Sensitivity analysis of λ scheduling strategies for 340M-parameter BregNet trained on 15B tokens with target 50% mean sparsity. Fixed uses a constant threshold; L2S and S2L schedule λ from large-to-small and small-to-large, respectively; Dynamic adapts the effective threshold using β statistics during each forward pass. DeltaNet is included as the baseline.

Model	Type	LM		Commonsense Reasoning									Context Retrieval	
		Wiki ppl↓	LMB ppl↓	LMB acc	PIQA acc	Hella acc_n	Wino acc	ARC-e acc	ARC-c acc_n	SciQ acc	BoolQ acc	COPA acc	FDA contains	SWDE contains
BregNet	Fixed	27.1	63.2	26.2	65.3	36.0	50.4	45.4	23.7	79.4	61.3	67.0	11.6	29.9
	L2S	27.1	63.3	26.0	64.6	35.9	52.4	44.7	24.0	79.4	60.0	68.0	11.3	27.1
	S2L	27.4	70.3	24.4	63.9	35.6	51.1	45.5	23.6	78.2	58.4	69.0	6.6	28.2
	Dynamic	27.3	61.5	25.9	64.3	35.6	53.8	47.5	22.4	78.2	55.1	73.0	9.8	28.8
DeltaNet	–	27.5	67.2	24.7	64.8	35.2	49.3	46.3	24.5	77.1	61.0	68.0	11.8	31.6

A Hyperparameter

For fair comparison, we conduct all models on the same setup setting, head dimension of 128 and employ the Llama2 tokenizer with a vocabulary size of 32,000. For 340M models, we follow the setting in DeltaNet (Yang et al., 2024c). Each model uses AdamW optimizer with a peak learning rate of 3×10^{-4}. weight decay of 0.1, and gradient clipping of 1.0. The learning rate follows a cosine annealing schedule with a 0.5B token warm-up period and batch size of 0.1M tokens. For sequence modeling, we set the training length to 2K tokens.

For 1B models, we follow the setting in Gated DeltaNet (Yang et al., 2024a) to use the AdamW optimizer with a peak learning rate of 4×10^{-4}, weight decay of 0.1, and gradient clipping of 1.0. The learning rate follows a cosine annealing schedule with a 1B token warm-up period and batch size of 0.5M tokens. For sequence modeling, we set the training length to 4K tokens, with the hybrid versions using a sliding window size of 2K.

B Sensitivity Analysis

Sensitivity Analysis Table. 7 shows under a wide range of sparsity levels, BregNet (Gate) maintains largely stable performance on language modeling, commonsense reasoning, and context retrieval benchmarks. In particular, increasing the maximum layer-wise sparsity from 29.7% to 97.7% only mildly changes Wiki perplexity while LAMBADA perplexity gets improved. Downstream accuracy also shows no consistent degradation trend across PIQA, HellaSwag, WinoGrande, ARC, SciQ, BoolQ, COPA, and SWDE. These results suggest that BregNet can induce substantial sparsity while preserving competitive modeling quality, especially on LAMBADA perplexity, which remains robust even under high sparsity.

Table. 8 compares different λ scheduling strategies in BregNet. In the "Type" column, Fixed uses a constant λ across layers, L2S schedules λ from large to small, S2L schedules λ from small to large, and Dynamic adapts the effective λ from the β statistics during each forward pass, resulting in layer-wise adaptive sparsity. While the Dynamic strategy obtains the best LAMBADA perplexity, the Fixed strategy consistently provides competitive overall performance and exhibits stronger robustness and

implementation simplicity compared to layer-wise scheduling strategies such as L2S and S2L. Therefore, we adopt the Fixed strategy in the remaining experiments as a practical trade-off between simplicity and performance.

C Proofs

C.1 Proof for Proposition 1

Proof. We evaluate the Bregman distance at the end of the chunk update $\mathbf{Z}_{m+1}$ using the convex-conjugate form in Lemma 5:

$$D_{\omega}^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) = \omega^*\big(\mathbf{Z}_m - \beta\nabla J_C(\mathbf{S}_m)\big) - \big\langle \mathbf{Z}_m - \beta\nabla J_C(\mathbf{S}_m), \bar{\mathbf{S}}\big\rangle + \omega(\bar{\mathbf{S}}).$$

Apply the gradient-Lipschitz upper bound to the conjugate function ω^*, whose gradient $\nabla\omega^*$ is 1-smooth:

$$\omega^*\big(\mathbf{Z}_m - \beta\nabla J_C(\mathbf{S}_m)\big) \le \omega^*(\mathbf{Z}_m) - \beta\,\langle \mathbf{S}_m, \nabla J_C(\mathbf{S}_m)\rangle + \frac{\beta^2}{2}\big\|\nabla J_C(\mathbf{S}_m)\big\|_F^2.$$

Substituting this bound into the Bregman distance and regrouping the terms gives

$$\begin{aligned} D_{\omega}^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) \le\ & \big[\omega^*(\mathbf{Z}_m) - \langle \mathbf{Z}_m, \bar{\mathbf{S}}\rangle + \omega(\bar{\mathbf{S}})\big] \\ & - \beta\,\big\langle \nabla J_C(\mathbf{S}_m), \mathbf{S}_m - \bar{\mathbf{S}}\big\rangle + \frac{\beta^2}{2}\big\|\nabla J_C(\mathbf{S}_m)\big\|_F^2 \\ =\ & D_{\omega}^{\mathbf{Z}_m}(\bar{\mathbf{S}}, \mathbf{S}_m) - \beta\,\big\langle \nabla J_C(\mathbf{S}_m), \mathbf{S}_m - \bar{\mathbf{S}}\big\rangle + \frac{\beta^2}{2}\big\|\nabla J_C(\mathbf{S}_m)\big\|_F^2. \end{aligned}$$

Because $J_C(\mathbf{S})$ is convex and ∇J_C is L_C-Lipschitz continuous from § C.6, and because $\nabla J_C(\bar{\mathbf{S}}) = 0$, the Baillon–Haddad co-coercivity theorem gives

$$\big\langle \nabla J_C(\mathbf{S}_m), \mathbf{S}_m - \bar{\mathbf{S}}\big\rangle \ge \frac{1}{L_C}\big\|\nabla J_C(\mathbf{S}_m)\big\|_F^2.$$

Substituting this inequality yields the chunk-level descent bound:

$$D_{\omega}^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) \le D_{\omega}^{\mathbf{Z}_m}(\bar{\mathbf{S}}, \mathbf{S}_m) + \left(\frac{\beta^2}{2} - \frac{\beta}{L_C}\right)\big\|\nabla J_C(\mathbf{S}_m)\big\|_F^2.$$

Step-size requirement. To guarantee monotonic descent across chunks, the additional term must be non-positive:

$$\frac{\beta^2}{2} - \frac{\beta}{L_C} \le 0 \quad\Longrightarrow\quad \beta \le \frac{2}{L_C}.$$

□

C.2 Proof for Thm. 2

Proof. By the standard three-point identity for Bregman distances, the relationship between the optimal parameter $\bar{\mathbf{S}}$, the current parameter $\mathbf{S}_m$, and the next parameter $\mathbf{S}_{m+1}$ is

$$\begin{aligned} & D_{\omega}^{\mathbf{Z}_m}(\bar{\mathbf{S}}, \mathbf{S}_m) - D_{\omega}^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) - D_{\omega}^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) \\ & = \big\langle \bar{\mathbf{S}} - \mathbf{S}_{m+1}, \mathbf{Z}_{m+1} - \mathbf{Z}_m\big\rangle. \end{aligned}$$

Substituting the dual update $\mathbf{Z}_{m+1} - \mathbf{Z}_m = -\beta\nabla J_C(\mathbf{S}_m)$ gives

$$\begin{aligned} \text{LHS} &= -\beta\,\big\langle \bar{\mathbf{S}} - \mathbf{S}_{m+1}, \nabla J_C(\mathbf{S}_m)\big\rangle && (15) \\ &= \beta\,\big\langle \mathbf{S}_m - \bar{\mathbf{S}}, \nabla J_C(\mathbf{S}_m)\big\rangle - \beta\,\langle \mathbf{S}_m - \mathbf{S}_{m+1}, \nabla J_C(\mathbf{S}_m)\rangle. && (16) \end{aligned}$$

Next, we use two standard properties of $J_C(\mathbf{S})$. First, by convexity,

$$\left\langle \mathbf{S}_m - \bar{\mathbf{S}}, \nabla J_C(\mathbf{S}_m) \right\rangle \geq J_C(\mathbf{S}_m) - J_C(\bar{\mathbf{S}}).$$

Second, since ∇J_C is L_C-Lipschitz continuous,

$$J_C(\mathbf{S}_{m+1}) \leq J_C(\mathbf{S}_m) + \langle \nabla J_C(\mathbf{S}_m), \mathbf{S}_{m+1} - \mathbf{S}_m \rangle + \frac{L_C}{2} \|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2.$$

Rearranging gives

$$-\langle \mathbf{S}_m - \mathbf{S}_{m+1}, \nabla J_C(\mathbf{S}_m) \rangle \geq J_C(\mathbf{S}_{m+1}) - J_C(\mathbf{S}_m) - \frac{L_C}{2} \|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2.$$

Substituting these two bounds into Eq. 16 gives

$$\begin{aligned} & D_\omega^{\mathbf{Z}_m}(\bar{\mathbf{S}}, \mathbf{S}_m) - D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) - D_\omega^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) \\ & \geq \beta\big(J_C(\mathbf{S}_m) - J_C(\bar{\mathbf{S}})\big) + \beta\big(J_C(\mathbf{S}_{m+1}) - J_C(\mathbf{S}_m)\big) \\ & \quad - \frac{\beta L_C}{2} \|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2 \\ & = \beta\big(J_C(\mathbf{S}_{m+1}) - J_C(\bar{\mathbf{S}})\big) - \frac{\beta L_C}{2} \|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2. \end{aligned}$$

Rearranging the inequality gives

$$\begin{aligned} \beta\big(J_C(\mathbf{S}_{m+1}) - J_C(\bar{\mathbf{S}})\big) \leq & \; D_\omega^{\mathbf{Z}_m}(\bar{\mathbf{S}}, \mathbf{S}_m) - D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) \\ & - D_\omega^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) + \frac{\beta L_C}{2} \|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2. \end{aligned}$$

Because $\omega(\mathbf{S})$ is strongly convex with modulus 1, Lemma 6 gives

$$D_\omega^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) \geq \frac{1}{2} \|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2.$$

Therefore,

$$\begin{aligned} \beta\big(J_C(\mathbf{S}_{m+1}) - J_C(\bar{\mathbf{S}})\big) \leq & \; D_\omega^{\mathbf{Z}_m}(\bar{\mathbf{S}}, \mathbf{S}_m) - D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) \\ & + \left(\frac{\beta L_C}{2} - \frac{1}{2} \right) \|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2. \end{aligned}$$

If $\beta \leq 1/L_C$, then

$$\left(\frac{\beta L_C}{2} - \frac{1}{2} \right) \leq 0.$$

Dropping the non-positive term yields

$$\beta\big(J_C(\mathbf{S}_{m+1}) - J_C(\bar{\mathbf{S}})\big) \leq D_\omega^{\mathbf{Z}_m}(\bar{\mathbf{S}}, \mathbf{S}_m) - D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}).$$

Summing over $m = 0, \ldots, T$ gives

$$\sum_{m=0}^{T} \beta\big(J_C(\mathbf{S}_{m+1}) - J_C(\bar{\mathbf{S}})\big) \leq D_\omega^{\mathbf{Z}_0}(\bar{\mathbf{S}}, \mathbf{S}_0) - D_\omega^{\mathbf{Z}_{T+1}}(\bar{\mathbf{S}}, \mathbf{S}_{T+1}) \leq D_\omega^{\mathbf{Z}_0}(\bar{\mathbf{S}}, \mathbf{S}_0).$$

Since the objective values are non-increasing under the step-size condition, the final objective error is the smallest in the sequence. Thus,

$$(T+1)\beta\big(J_C(\mathbf{S}_{T+1}) - J_C(\bar{\mathbf{S}})\big) \leq D_\omega^{\mathbf{Z}_0}(\bar{\mathbf{S}}, \mathbf{S}_0).$$

Dividing by $(T+1)\beta$ gives

$$J_C(\mathbf{S}_{T+1}) - J_C(\bar{\mathbf{S}}) \leq \frac{D_\omega^{\mathbf{Z}_0}(\bar{\mathbf{S}}, \mathbf{S}_0)}{\beta(T+1)}.$$

□

C.3 Proof for Thm. 3

Proof. Lemma D.1 shows that the actual sequence $\mathbf{S}_m$ matches the oracle sequence $\mathbf{S}'_m$, i.e., no false positives occur, as long as the dual variables in the non-support space $\mathcal{S}^c$ remain below the ℓ_1 penalty threshold:

$$\|\mathbf{Z}_{m,\mathcal{S}^c}\|_\infty < \lambda.$$

For the parameters to remain in the oracle subspace, the Irrepresentable Condition (IRR) requires the signal term in the non-support space $\mathbf{I}_{m,\mathcal{S}^c}$ to be bounded away from the threshold by a margin η (Osher et al., 2016):

$$\|\mathbf{I}_{m,\mathcal{S}^c}\|_\infty \leq (1-\eta)\lambda \quad \text{for some } \eta \in (0,1].$$

Consequently, a false positive cannot occur if the accumulated noise $\mathbf{W}_m$ satisfies

$$\|\mathbf{W}_{m,\mathcal{S}^c}\|_\infty < \eta\lambda.$$

The accumulated noise matrix $\mathbf{W}_m$ is a weighted sum of mC independent sub-Gaussian random variables:

$$\mathbf{W}_m = \beta \sum_{t=1}^{mC} \boldsymbol{\epsilon}_t \mathbf{k}_t^\top.$$

Since the noise $\boldsymbol{\epsilon}_t$ is sub-Gaussian, $\mathbf{W}_m$ behaves as a random walk over m iterations and chunk size C, as shown in Lemma 7. Let d be the dimension of the non-support space $\mathcal{S}^c$. The variance proxy for any single coordinate of $\mathbf{W}_m$ is bounded by

$$\beta^2\sigma^2 \sum_{t=1}^{mC} \|\mathbf{k}_t\|_2^2 \leq \beta^2\sigma^2 mCK^2.$$

By the union bound, the probability that the maximal entry of the noise matrix exceeds the tolerance $\eta\lambda$ is bounded as

$$\mathbb{P}(\|\mathbf{W}_{m,\mathcal{S}^c}\|_\infty \geq \eta\lambda) \leq 2d \exp\left(-\frac{(\eta\lambda)^2}{2\beta^2\sigma^2 mCK^2}\right).$$

To keep the false-positive probability below δ, it suffices to require

$$2d \exp\left(-\frac{\eta^2\lambda^2}{2\beta^2\sigma^2 mCK^2}\right) \leq \delta.$$

Scaling law for λ. To guarantee path consistency and keep the false-positive probability below δ, the scaling law for λ is, up to constants and log factors,

$$\lambda \;\propto\; \beta\sqrt{mC}\sqrt{\log\left(\frac{d}{\delta}\right)}.$$

Stopping time for training steps. Taking logarithms and solving for m gives

$$\frac{\eta^2\lambda^2}{2\beta^2\sigma^2 mCK^2} \geq \log\left(\frac{2d}{\delta}\right),$$

$$m \leq \frac{\eta^2\lambda^2}{2\beta^2\sigma^2 CK^2 \log(2d/\delta)}.$$

Thus, to prevent the accumulated chunk noise from escaping the oracle subspace and causing overfitting, the algorithm should stop at

$$M_{\text{stop}} = \left\lfloor \frac{\eta^2\lambda^2}{2\beta^2\sigma^2 K^2 \log(2d/\delta)} \cdot \frac{1}{C} \right\rfloor.$$

This shows that as the chunk size C or sequence noise σ^2 increases, the number of safe primal updates M_{stop} decreases at rate $O(1/C)$. □

C.4 Closed-Form Solution for Eq. 3

The objective function is

$$F(\mathbf{S}) = \lambda \|\mathbf{S}\|_1 + \frac{1}{2}\|\mathbf{S} - \mathbf{Z}_{t+1}\|_F^2.$$

Because the ℓ_1 norm and Frobenius norm decompose element-wise,

$$\|\mathbf{S}\|_1 = \sum_{i,j} |s_{i,j}|, \qquad \|\mathbf{S} - \mathbf{Z}_{t+1}\|_F^2 = \sum_{i,j} (s_{i,j} - z_{i,j})^2,$$

the optimization decouples into independent scalar problems.

For any scalar element s of $\mathbf{S}$ and corresponding element z of $\mathbf{Z}_{t+1}$, we minimize

$$f(s) = \lambda |s| + \frac{1}{2}(s - z)^2.$$

Since $|s|$ is non-differentiable at $s = 0$, we set the subdifferential of $f(s)$ to zero:

$$0 \in \partial f(s) = \lambda\, \partial|s| + (s - z), \qquad z - s \in \lambda\, \partial|s|.$$

The subdifferential of the absolute value is

$$\partial|s| = \begin{cases} \{1\}, & s > 0, \\ \{-1\}, & s < 0, \\ [-1, 1], & s = 0. \end{cases}$$

Solving $z - s \in \lambda\, \partial|s|$ gives three cases:

$$s = \begin{cases} z - \lambda, & z > \lambda, \\ 0, & |z| \le \lambda, \\ z + \lambda, & z < -\lambda. \end{cases}$$

Equivalently,

$$s = \operatorname{sgn}(z) \max\big(|z| - \lambda, 0\big).$$

Applying this scalar result element-wise gives the matrix-level closed form

$$\mathbf{S}_{t+1} = T_\lambda(\mathbf{Z}_{t+1}).$$

C.5 Row-Wise Dual Update

Consider the update rule for row i of the dual state matrix $\mathbf{Z}$. Let $\mathbf{z}_i$ be the i-th row of $\mathbf{Z}$, and let $\mathbf{r}_i$ be the i-th row of the static chunk mask $\mathbf{M}_{\text{chunk}}$.

The full matrix update term is

$$\Delta\mathbf{Z} = -\beta(\mathbf{M} \odot \mathbf{Z})\mathbf{k}\mathbf{k}^\top.$$

For row i, the element-wise product $\mathbf{M} \odot \mathbf{Z}$ scales that row by $\mathbf{r}_i$:

$$(\Delta\mathbf{Z})_i = -\beta(\mathbf{r}_i \odot \mathbf{z}_i)\mathbf{k}\mathbf{k}^\top.$$

Using the identity $(\mathbf{a} \odot \mathbf{b}) \cdot \mathbf{c} = \mathbf{a} \cdot (\mathbf{b} \odot \mathbf{c})$, we obtain

$$(\mathbf{r}_i \odot \mathbf{z}_i)\mathbf{k} = \sum_j r_{i,j} z_{i,j} k_j = \sum_j z_{i,j}(r_{i,j} k_j) = \mathbf{z}_i \cdot (\mathbf{r}_i \odot \mathbf{k}).$$

Define the effective read key for row i as

$$\tilde{\mathbf{k}}_i = \mathbf{r}_i \odot \mathbf{k}.$$

Then the row-wise update becomes a standard rank-one transition:

$$\mathbf{z}_i^{(t)} = \mathbf{z}_i^{(t-1)} \underbrace{\left(\mathbf{I} - \beta_t \tilde{\mathbf{k}}_{t,i} \mathbf{k}_t^\top\right)}_{\text{row-specific transition}} + \text{updates}.$$

C.6 Lipschitz Continuity of the Gradient

We compute the gradient of $J_C(\mathbf{S})$ with respect to $\mathbf{S}$:

$$\nabla J_C(\mathbf{S}) = \sum_{i=1}^{C} \big(\mathbf{S}\mathbf{k}_{t_i} - \mathbf{v}_{t_i}\big)\mathbf{k}_{t_i}^\top,$$

or equivalently,

$$\nabla J_C(\mathbf{S}) = \mathbf{S}\left(\sum_{i=1}^{C} \mathbf{k}_{t_i}\mathbf{k}_{t_i}^\top\right) - \sum_{i=1}^{C} \mathbf{v}_{t_i}\mathbf{k}_{t_i}^\top.$$

Let

$$\mathbf{K}_C = \sum_{i=1}^{C} \mathbf{k}_{t_i}\mathbf{k}_{t_i}^\top.$$

For any two matrices $\mathbf{S}_1$ and $\mathbf{S}_2$,

$$\begin{aligned}\|\nabla J_C(\mathbf{S}_1) - \nabla J_C(\mathbf{S}_2)\|_F &= \left\|\big(\mathbf{S}_1\mathbf{K}_C - \sum_{i=1}^{C} \mathbf{v}_{t_i}\mathbf{k}_{t_i}^\top\big) - \big(\mathbf{S}_2\mathbf{K}_C - \sum_{i=1}^{C} \mathbf{v}_{t_i}\mathbf{k}_{t_i}^\top\big)\right\|_F \\ &= \|(\mathbf{S}_1 - \mathbf{S}_2)\mathbf{K}_C\|_F.\end{aligned}$$

Using the submultiplicative property $\|\mathbf{AB}\|_F \le \|\mathbf{A}\|_F\|\mathbf{B}\|_2$, where $\|\cdot\|_2$ is the spectral norm, we have

$$\|(\mathbf{S}_1 - \mathbf{S}_2)\mathbf{K}_C\|_F \le \|\mathbf{K}_C\|_2\|\mathbf{S}_1 - \mathbf{S}_2\|_F.$$

Therefore, the gradient satisfies the Lipschitz condition

$$\|\nabla J_C(\mathbf{S}_1) - \nabla J_C(\mathbf{S}_2)\|_F \le L_C\|\mathbf{S}_1 - \mathbf{S}_2\|_F,$$

with Lipschitz constant

$$L_C = \left\|\sum_{i=1}^{C} \mathbf{k}_{t_i}\mathbf{k}_{t_i}^\top\right\|_2 = \lambda_{\max}\left(\sum_{i=1}^{C} \mathbf{k}_{t_i}\mathbf{k}_{t_i}^\top\right).$$

D Lemmas

D.1 Primal–Dual Sparsity Condition

Lemma 4 (Primal–Dual Sparsity Condition (Cai et al., 2009b))**.** *Under the ℓ_1-regularized LBI framework, if the dual variable $\mathbf{Z}_m$ satisfies*

$$\|\mathbf{Z}_{m,\mathcal{S}^c}\|_\infty < \lambda$$

on a subspace $\mathcal{S}^c$, then the corresponding primal variable satisfies

$$\mathbf{S}_{m,\mathcal{S}^c} = 0.$$

Proof. In LBI with the ℓ_1 regularization term $\lambda\|\mathbf{S}\|_1$, the primal variable $\mathbf{S}_m$ is recovered from the dual variable $\mathbf{Z}_m$ through the element-wise soft-shrinkage operator:

$$\mathbf{S}_m = T_\lambda(\mathbf{Z}_m),$$

where T_λ is defined element-wise for any scalar z as

$$T_\lambda(z) = \operatorname{sgn}(z)\max(|z| - \lambda, 0).$$

Consider any coordinate $(p, q) \in \mathcal{S}^c$. The primal update gives

$$[\mathbf{S}_m]_{p,q} = \operatorname{sgn}([\mathbf{Z}_m]_{p,q})\max\big(|[\mathbf{Z}_m]_{p,q}| - \lambda, 0\big).$$

The condition

$$\|\mathbf{Z}_{m,\mathcal{S}^c}\|_\infty = \max_{(p,q)\in\mathcal{S}^c} |[\mathbf{Z}_m]_{p,q}| < \lambda$$

implies that, for every $(p,q) \in \mathcal{S}^c$,

$$|[\mathbf{Z}_m]_{p,q}| - \lambda < 0.$$

Therefore,

$$\max\big(|[\mathbf{Z}_m]_{p,q}| - \lambda, 0\big) = 0,$$

and hence

$$[\mathbf{S}_m]_{p,q} = 0, \qquad \forall (p,q) \in \mathcal{S}^c.$$

Thus, $\mathbf{S}_{m,\mathcal{S}^c} = 0$.

Since the oracle sequence $\mathbf{S}'_m$ is defined by enforcing $\mathbf{S}'_{m,\mathcal{S}^c} = 0$ while applying identical updates on the true support $\mathcal{S}$, the actual sequence matches the oracle sequence, i.e., $\mathbf{S}_m = \mathbf{S}'_m$, as long as the threshold is not exceeded. □

D.2 Dual Form for Bregman Divergence

Lemma 5 (Dual Form for Bregman Divergence (Cai et al., 2009b))**.** *Let ω be a convex function and let ω^* denote its convex conjugate. For $\bar{\mathbf{S}}$ and $\mathbf{S}_{m+1}$ with subgradient $\mathbf{Z}_{m+1} \in \partial\omega(\mathbf{S}_{m+1})$, the Bregman distance admits the dual form*

$$D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) = \omega(\bar{\mathbf{S}}) - \langle \mathbf{Z}_{m+1}, \bar{\mathbf{S}} \rangle + \omega^*(\mathbf{Z}_{m+1}).$$

In particular, if $\mathbf{S}_{m+1} = \nabla\omega^*(\mathbf{Z}_{m+1})$ and $\mathbf{Z}_{m+1} = \mathbf{Z}_m - \beta\nabla J_C(\mathbf{S}_m)$, then

$$\begin{aligned} D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) = \omega(\bar{\mathbf{S}}) &- \big\langle \mathbf{Z}_m - \beta\nabla J_C(\mathbf{S}_m), \bar{\mathbf{S}} \big\rangle \\ &+ \omega^*(\mathbf{Z}_m - \beta\nabla J_C(\mathbf{S}_m)) \,. \end{aligned}$$

Proof. By definition, the Bregman distance between $\bar{\mathbf{S}}$ and $\mathbf{S}_{m+1}$ with respect to the subgradient $\mathbf{Z}_{m+1} \in \partial\omega(\mathbf{S}_{m+1})$ is

$$D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) = \omega(\bar{\mathbf{S}}) - \omega(\mathbf{S}_{m+1}) - \big\langle \mathbf{Z}_{m+1}, \bar{\mathbf{S}} - \mathbf{S}_{m+1} \big\rangle \,.$$

Expanding the inner product gives

$$D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) = \omega(\bar{\mathbf{S}}) - \omega(\mathbf{S}_{m+1}) - \langle \mathbf{Z}_{m+1}, \bar{\mathbf{S}} \rangle + \langle \mathbf{Z}_{m+1}, \mathbf{S}_{m+1} \rangle.$$

Because $\mathbf{S}_{m+1}$ and $\mathbf{Z}_{m+1}$ form a conjugate pair, the Fenchel–Young equality gives

$$\omega(\mathbf{S}_{m+1}) + \omega^*(\mathbf{Z}_{m+1}) = \langle \mathbf{Z}_{m+1}, \mathbf{S}_{m+1} \rangle.$$

Thus,

$$-\omega(\mathbf{S}_{m+1}) + \langle \mathbf{Z}_{m+1}, \mathbf{S}_{m+1} \rangle = \omega^*(\mathbf{Z}_{m+1}).$$

Substituting this identity yields

$$D_\omega^{\mathbf{Z}_{m+1}}(\bar{\mathbf{S}}, \mathbf{S}_{m+1}) = \omega(\bar{\mathbf{S}}) - \langle \mathbf{Z}_{m+1}, \bar{\mathbf{S}} \rangle + \omega^*(\mathbf{Z}_{m+1}).$$

Finally, substituting $\mathbf{Z}_{m+1} = \mathbf{Z}_m - \beta\nabla J_C(\mathbf{S}_m)$ proves the stated expression. □

D.3 Bregman Distance Lower Bound

Lemma 6 (Lower Bound for Bregman Distance under Strong Convexity (Cai et al., 2009b))**.** *Let $\omega(\mathbf{S})$ be μ-strongly convex with respect to the Frobenius norm $\|\cdot\|_F$. For any $\mathbf{S}_{m+1}, \mathbf{S}_m$ and any subgradient $\mathbf{Z}_m \in \partial\omega(\mathbf{S}_m)$, the associated Bregman distance satisfies*

$$D_\omega^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) \geq \frac{\mu}{2}\|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2.$$

Proof. The Bregman distance generated by ω is

$$D_\omega^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) = \omega(\mathbf{S}_{m+1}) - \omega(\mathbf{S}_m) - \langle \mathbf{Z}_m, \mathbf{S}_{m+1} - \mathbf{S}_m \rangle .$$

Since ω is μ-strongly convex,

$$\omega(\mathbf{S}_{m+1}) \geq \omega(\mathbf{S}_m) + \langle \mathbf{Z}_m, \mathbf{S}_{m+1} - \mathbf{S}_m \rangle + \frac{\mu}{2}\|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2.$$

Rearranging gives

$$D_\omega^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) \geq \frac{\mu}{2}\|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2.$$

For our choice of ω, the strong convexity modulus is $\mu = 1$, and therefore

$$D_\omega^{\mathbf{Z}_m}(\mathbf{S}_{m+1}, \mathbf{S}_m) \geq \frac{1}{2}\|\mathbf{S}_{m+1} - \mathbf{S}_m\|_F^2.$$

□

D.4 Accumulated Sub-Gaussian Noise as a Random Walk

Lemma 7 (Accumulated Sub-Gaussian Noise as a Random Walk)**.** *Let $N = mC$ be the total number of processed vectors. Define the accumulated noise matrix*

$$\mathbf{W}_m = \beta \sum_{t=1}^{N} \boldsymbol{\epsilon}_t \mathbf{k}_t^\top,$$

and the step increment

$$\Delta\mathbf{W}_t = \beta \boldsymbol{\epsilon}_t \mathbf{k}_t^\top.$$

Assume the observation noise $\{\boldsymbol{\epsilon}_t\}_{t=1}^N$ is independent, zero-mean, and sub-Gaussian with variance proxy σ^2, and assume the inputs are bounded by $\|\mathbf{k}_t\|_2 \leq K$. Then $\mathbf{W}_m$ is a sum of independent, zero-mean increments. Moreover, for any scalar entry $[\mathbf{W}_m]_{p,q}$, its variance proxy satisfies

$$s^2 \leq \beta^2 \sigma^2 K^2 mC.$$

Proof. Let $N = mC$. The accumulated noise matrix is

$$\mathbf{W}_m = \beta \sum_{t=1}^{N} \boldsymbol{\epsilon}_t \mathbf{k}_t^\top,$$

with increment

$$\Delta\mathbf{W}_t = \beta \boldsymbol{\epsilon}_t \mathbf{k}_t^\top.$$

Since $\boldsymbol{\epsilon}_t$ is independent and zero-mean,

$$\mathbb{E}[\Delta\mathbf{W}_t] = \beta\, \mathbb{E}[\boldsymbol{\epsilon}_t]\mathbf{k}_t^\top = 0.$$

Let $\mathcal{F}_n = \sigma(\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_n)$ be the natural filtration. The partial sums $\{\mathbf{W}^{(n)}\}_{n=1}^N$ form a discrete-time martingale because

$$\mathbb{E}[\mathbf{W}^{(n)} \mid \mathcal{F}_{n-1}] = \mathbf{W}^{(n-1)} + \mathbb{E}[\Delta\mathbf{W}_n \mid \mathcal{F}_{n-1}] = \mathbf{W}^{(n-1)}.$$

For a single scalar element of $\mathbf{W}_m$ at row p and column q,

$$[\mathbf{W}_m]_{p,q} = \beta \sum_{t=1}^{mC} [\boldsymbol{\epsilon}_t]_p [\mathbf{k}_t]_q.$$

Since each $[\boldsymbol{\epsilon}_t]_p$ is an independent sub-Gaussian variable with variance proxy σ^2, the variance proxy of this linear combination is

$$s^2 = \sum_{t=1}^{mC} \big(\beta [\mathbf{k}_t]_q\big)^2 \sigma^2.$$

Using $\|\mathbf{k}_t\|_2 \leq K$, we have $[\mathbf{k}_t]_q^2 \leq K^2$, and hence

$$s^2 \leq \beta^2 \sigma^2 \sum_{t=1}^{mC} K^2 = \beta^2 \sigma^2 K^2 mC.$$

Thus, $\mathbf{W}_m$ is an accumulated sum of independent zero-mean increments, with coordinate-wise variance proxy growing linearly in mC. □

E Use of Large Language Models (LLMs)

During the preparation of this manuscript, we used Large Language Models (LLMs) for language assistance, including grammar checking and text polishing, to improve clarity and readability.

In addition, during the experimental phase, we used LLMs to assist in generating functional scripts (e.g., dataset preprocess, evaluation harnesses, and metric collection/statistics scripts). All such code was reviewed, tested, and, where necessary, modified by the authors. The authors take full responsibility for the correctness of the implementation, the experimental design, and the reported results and conclusions. LLMs were not used to generate experimental results or to make scientific decisions.