PLASH: Provably Linear-Time Attention with Selective Higher-Order Feature Sketching

Yuwen Huang [1] **Xiang Pan** [2]

Abstract

Attention selects information from long contexts, but standard softmax attention scales as $O(N_q N_k)$ in the number of queries N_q and keys N_k, making long-context training and inference expensive. We propose PLASH, an attention block with *provably linear-time* complexity in N_k that preserves the usual interface: each query still returns a data-dependent weighted combination of values. PLASH first compresses the key / value side into $M \ll N_k$ learned representatives, and then restores expressivity by enriching these representatives with *selective higher-order feature sketching* (e.g., TensorSketch), which approximates chosen polynomial interactions without explicit feature expansion. The final softmax readout from $\mathbf{Q}$ to the enriched $(\mathbf{K}_g, \mathbf{V}_g)$ is exact, so PLASH applies to both self- and cross-attention by treating N_q and N_k independently. We give a runtime analysis $O(N_k M d + N_q M d)$ (plus sketching costs), provide error bounds for the randomized sketches and an end-to-end deviation analysis relative to standard attention, and show strong long-context performance with favorable scaling versus efficient-attention baselines.

1. Introduction

Self-attention enables dense, content-dependent token interactions, but its standard form scales quadratically with sequence length (Vaswani et al., 2017). Given inputs, i.e., queries $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$, keys $\mathbf{K} \in \mathbb{R}^{N_k \times d_k}$, and values $\mathbf{V} \in \mathbb{R}^{N_k \times d_v}$, scaled dot-product attention operation, $\mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V})$ is:

$$\mathrm{softmax}\left(\tfrac{1}{\sqrt{d_k}}\mathbf{Q}\mathbf{K}^\top, \mathrm{dim} = -1\right)\mathbf{V},$$

whose dominant cost comes from the $N_q \times N_k$ logit matrix $\mathbf{Q}\mathbf{K}^\top$. In the common regime, $N_q \approx N_k \approx N$, this requires $\mathcal{O}(N^2 d_k)$ arithmetic and (typically) $\mathcal{O}(N^2)$ attention weights, making quadratic scaling the main obstacle to long-context training and inference.

Why better computing kernels are not enough. IO-aware implementations such as *FlashAttention* reduce memory overhead while preserving *exact* standard attention expressivity (Dao et al., 2022). These improvements extend feasible context lengths in the 200K–1M range (Anthropic, 2024). However, standard attention still requires $\Theta(N_q N_k)$ pairwise scores: at $N = 64\text{K}$ this is already $\approx 4 \times 10^9$, and at $N = 1\text{M}$ it is 10^{12}. Thus, long-context scaling ultimately requires an *algorithmic* route to sub-quadratic (ideally linear) interaction counts, not only faster implementations of the quadratic primitive.

A recurring tension in linear-time token mixing. A large literature reduces the quadratic cost of attention in two main ways: (i) restrict which token pairs interact, or (ii) replace softmax mixing with a more structured operator. Examples of (i) include sparse or block-pattern attention (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020), low-rank / Nyström approximations (Wang et al., 2020; Xiong et al., 2021), and hashing-based routing (Kitaev et al., 2020). Examples of (ii) include kernel / feature linearization (Katharopoulos et al., 2020; Choromanski et al., 2021), Fourier mixing layers (Lee-Thorp et al., 2022), and state-space or gated-recurrence mixers (Gu et al., 2022a; Gu & Dao, 2024; Poli et al., 2023; De et al., 2024; Yang et al., 2024). These methods scale to long contexts, but their accuracy–efficiency trade-offs are often baked into architectural choices and hyperparameters. As a result, it is typically nontrivial to obtain *layer-level* guarantees that remain faithful to the scaled dot-product *softmax* attention interface (Tay et al., 2022; 2021).

Our approach. We introduce PLASH (**P**rovably **L**inear-Time **A**ttention with **S**elective **H**igher-order feature **S**ketching), which makes two structural decisions explicit.

The work described in this paper was partially supported by Lingnan University under Grants (SDS24A16). [1]Thrust of Data Science and Analytics, The Hong Kong University of Science and Technology (Guangzhou), Guanzhou, China [2]Division of Industrial Data Science, Lingnan University, Hong Kong, China. Correspondence to: Xiang Pan <first2.last2@link.cuhk.edu.hk>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

First, PLASH compresses keys and values into matrices of column length $M \ll N_k$. Second, it retains an *exact* scaled dot-product readout from queries to compressed keys and values. All approximation is localized to a single enrichment operator applied *after* compression:

$$(\mathbf{K}, \mathbf{V}) \xrightarrow{\text{compressing}} (\widetilde{\mathbf{K}}, \widetilde{\mathbf{V}}) \xrightarrow[\text{enrichment}]{\text{feature}} (\mathbf{K}_g, \mathbf{V}_g) \xrightarrow{\text{readout}} \mathrm{Atten}(\mathbf{Q}; \mathbf{K}_g, \mathbf{V}_g).$$

This separation is central: the attention *interface* is preserved at the end of the block, while the only approximation occurs inside the intermediate feature extraction.

Comparison with related work. Most *token-level* randomized attention approximations inject stochasticity *before* any positions of the bottleneck (i.e., $\mathbf{QK}^\top$), so the random map (or sampling rule) directly perturbs the surrogate for the dense logit operator $\mathbf{QK}^\top \in \mathbb{R}^{N_q \times N_k}$ across the full $N_q \times N_k$ interface. This includes random-feature / kernel linearizations of softmax attention (e.g., Performer / FAVOR+ (Choromanski et al., 2021) and related random-feature designs (Peng et al., 2021; Zheng et al., 2023)) as well as polynomial-sketch constructions (Kacham et al., 2024) and stochastic estimators (Peng et al., 2021; Zheng et al., 2023) that operate at the original token resolution, where randomness changes the effective query–token interaction weights for *all* positions. In contrast, PLASH first forms a *deterministic* compressed $(\mathbf{K}, \mathbf{V})$ of length $M \ll N_k$ (Stage I), injects randomness *only* in the enrichment map (Stage II), and then performs an *exact* scaled dot-product softmax readout from $\mathbf{Q}$ to $(\mathbf{K}_g, \mathbf{V}_g)$ (Stage III). Consequently, once the sketch is fixed, the remaining computation is fully deterministic. This allows us to rigorously bound the sketching error in Stage II and control its propagation through the deterministic subsequent steps.

Why higher-order interactions, with explicit budgets. KV compression discards token-level detail, so many distinct tokens can map to the same compressed key. If we process the compressed keys / values only linearly, the block can become a low-rank bottleneck. PLASH restores expressivity by adding *controlled higher-order (polynomial) interactions* on the compressed keys / values, so that information lost by linear compression can reappear in feature cross-terms (Schölkopf & Smola, 2002; Gao et al., 2016). We implement these interactions efficiently via TensorSketch (Pham & Pagh, 2013), using sketching results for polynomial kernels (Avron et al., 2014; Ahle et al., 2020). This yields two budgetable knobs: the degree set $\mathcal{K}$ and sketch sizes $\{D_k\}_{k\in\mathcal{K}}$ (larger D_k gives higher fidelity at higher cost) (Pham & Pagh, 2013; Gao et al., 2016; Fukui et al., 2016).

Why we can certify end-to-end deviation per input. As discussed, the only algorithmic randomness is localized to the TensorSketch enrichment, we can certify its effect using forward-pass quantities and then propagate the bound through the deterministic subsequent pipeline using local stability bounds. This yields an input-dependent forward-pass certificate for the final-layer output without forming the dense $N_q \times N_k$ logits.

1.1. Contributions

Our primary contributions are threefolds:

1. **KV-side compression with localized randomness and explicit budgets.** PLASH compresses only $(\mathbf{K}, \mathbf{V})$ to length $M \ll N_k$ and keeps an *exact* softmax readout from $\mathbf{Q}$ to $(\mathbf{K}_g, \mathbf{V}_g)$. All approximation is confined to one TensorSketch enrichment step, with explicit knobs: the degree set $\mathcal{K}$ and sketch sizes $\{D_k\}_{k\in\mathcal{K}}$. Our design excels in flexibility where the compression can be flexibly configured to apply to any pair within the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ triplet.
2. **Forward-pass, per-input deviation certificates (accessible).** We give a high-probability (ϵ, δ) guarantee on how far PLASH's output can deviate from standard softmax attention on the *same* input. The bound is computable from the forward pass: we first bound the sketching error introduced in Stage II and then propagate it through the remaining deterministic operations (mixer, projections, exact readout), without forming the $N_q \times N_k$ attention matrix. Moving beyond the empirical validation in prior work, our certificate enables designing provably reliable randomized attention.
3. **Evaluation on long-sequence forecasting.** We evaluate PLASH on long-sequence time-series forecasting benchmarks, including Electricity Transformer Temperature (ETT), Electricity Consuming Load (ECL), and Weather (WTH), under the Informer protocol (Zhou et al., 2021; Zhang et al., 2023), using Informer as the backbone and replacing only the attention module for fair comparisons. We compare to strong efficient-attention baselines and report Mean Squared Error (MSE) / Mean Absolute Error (MAE), latency, and peak storage. PLASH is consistently competitive and achieves the best results on Weather (WTH) and ETTm1, while exhibiting milder increases in latency and memory usage for long sequence length.

2. PLASH: A Provably Linear-Time Attention Mechanism

This section presents PLASH, an efficient attention mechanism that addresses the quadratic complexity of standard self-attention. At its core, PLASH achieves *provable linear-time complexity* by re-expressing the costly pairwise interaction through a selective, higher-order feature mapping

scheme. This approach reduces the computational footprint while maintaining and generalizing the expressive power of standard dot-product attention. The remainder of this section is organized as follows:

- **Section 2.1:** We review the standard self-attention mechanism (Vaswani et al., 2017), formalizing its quadratic scaling with sequence length and the efficiency challenge that PLASH addresses.
- **Section 2.2:** We introduce the PLASH architecture. The key innovation is a compression step guided by learnable prototypes, followed by an expressive, higher-order, randomized feature transformation that enriches the representation before the final output.

2.1. Revisiting Standard Self-attention Mechanism

This section recalls scaled dot-product attention and its main cost. Self-attention computes a weighted sum of values, where weights come from query–key similarities passed through a row-wise softmax. For $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$, $\mathbf{K} \in \mathbb{R}^{N_k \times d_k}$, $\mathbf{V} \in \mathbb{R}^{N_k \times d_v}$, $\mathbf{z} \in \mathbb{R}^m$, and $j \in [m]$, we define $\mathrm{softmax}(\mathbf{z})_j \triangleq \exp(\mathbf{z}_j)/\big(\sum_{\ell=1}^m \exp(\mathbf{z}_\ell)\big)$ and apply it row-wise to a matrix. The scaled dot-product attention operator is (Vaswani et al., 2017):

$$\mathrm{Atten}(\mathbf{Q};\mathbf{K},\mathbf{V}) \triangleq \mathrm{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}, \mathrm{dim} = -1\Big)\mathbf{V}. \quad (1)$$

In the Transformer architecture, this single operator serves both self-attention and cross-attention. In self-attention, $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are all derived from the same input sequence (with $N_q = N_k$). In cross-attention, $\mathbf{Q}$ typically comes from a target sequence (e.g., in a decoder), while $\mathbf{K}$ and $\mathbf{V}$ come from a source sequence.

The main cost in (1) is forming the logit matrix $\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N_q \times N_k}$, which takes $\mathcal{O}(N_q N_k d_k)$ time and typically stores $\mathcal{O}(N_q N_k)$ weights. In self-attention, $N_q \approx N_k \approx N$, so this becomes $\mathcal{O}(N^2 d_k)$ time and $\mathcal{O}(N^2)$ storage, which limits long contexts. Prior work reduces this cost by replacing dense interactions with structured mixing, such as $\mathcal{O}(N \log N)$ Fourier token mixing (Lee-Thorp et al., 2022; Cooley & Tukey, 1965) or $\mathcal{O}(N)$ kernel / random-feature attention (Katharopoulos et al., 2020; Choromanski et al., 2021). These methods are effective, but they often introduce extra structure (kernel choices, fixed mixing, or specific approximations), making layer-level comparisons to the softmax attention inaccessible, which motivates the error analysis in Section 3.

2.2. PLASH: A Linear-Time Attention

PLASH achieves linear time by compressing only the key / value side. We keep $\mathbf{Q}$ at full resolution and map $(\mathbf{K},\mathbf{V})$ to a shorter pair $(\mathbf{K}_\mathrm{g},\mathbf{V}_\mathrm{g})$ of length $M \ll N_k$. We apply standard attention unchanged: $\mathbf{Y}_{\mathrm{PLASH}} \triangleq \mathrm{Atten}(\mathbf{Q};\mathbf{K}_\mathrm{g},\mathbf{V}_\mathrm{g})$. Thus, PLASH is a drop-in replacement for the usual $(\mathbf{Q},\mathbf{K},\mathbf{V})$ block. see Figure 1 for an overview of the three-stage PLASH block.

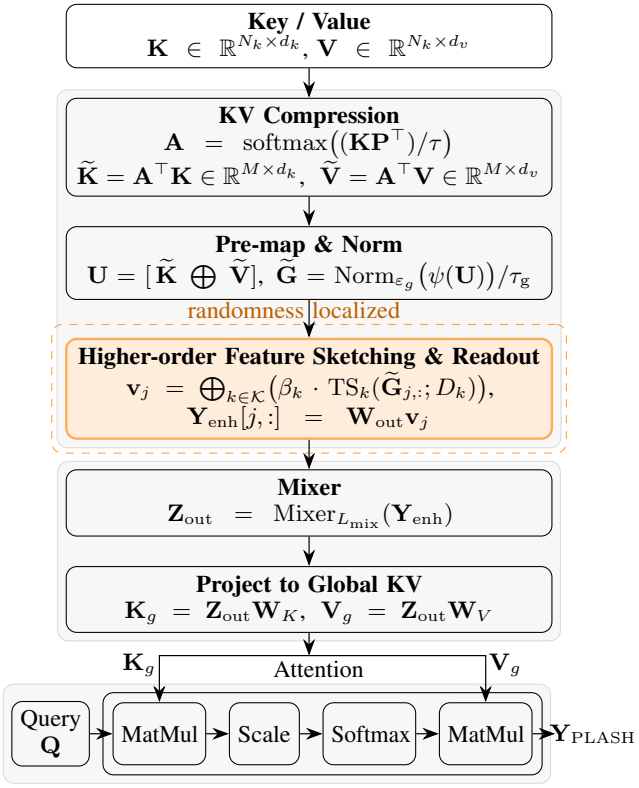


Figure 1. PLASH block in the $(\mathbf{Q},\mathbf{K},\mathbf{V})$ interface.

Knob 1: compressed length M. Replacing $\mathbf{K}$ by $\mathbf{K}_\mathrm{g}$ changes the computation from $\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N_q \times N_k}$ to $\mathbf{Q}\mathbf{K}_\mathrm{g}^\top \in \mathbb{R}^{N_q \times M}$, reducing the cost from $N_q N_k$ to $N_q M$.

Knob 2: sketch sizes $\{D_k\}_{k\in\mathcal{K}}$. After compression, we enrich each compressed item with selective higher-order feature sketches. The degree set $\mathcal{K}$ chooses which interaction orders are included, and D_k controls the approximation fidelity at degree k: larger D_k yields a more accurate higher-order representation. Since randomness appears only in this enrichment step, we can certify its error and propagate it through the deterministic subsequent maps.

2.2.1. Stage I: KV Compression

This stage compresses the original $(\mathbf{K},\mathbf{V})$ matrices of column lengths N_k and N_v, respectively, into a compact pair $(\widetilde{\mathbf{K}},\widetilde{\mathbf{V}})$ of size $M \ll N_k$ via prototype-based routing and weighted pooling. The goal is to form a compact representation while preserving the query–key–value relationship: $\widetilde{\mathbf{K}}$ stores representative *keys*, and $\widetilde{\mathbf{V}}$ stores the corresponding aggregated *values* from the cluster.

The compression uses M learnable prototype vectors $\mathbf{P} \in$

$\mathbb{R}^{M\times d_k}$, which is updated via standard training gradients (not randomized). A soft assignment (routing) is computed as

$$\mathbf{S} \triangleq \mathbf{K}\mathbf{P}^\top \in \mathbb{R}^{N_k\times M}, \tag{2}$$

$$\mathbf{A} \triangleq \mathrm{softmax}(\mathbf{S}/\tau,\ \mathrm{dim}=-1) \in \mathbb{R}^{N_k\times M}, \tag{3}$$

where $\tau >$ is a fixed temperature parameter that controls the sharpness of the assignment. Importantly, Stage I introduces *no algorithmic randomness*: once $(\mathbf{K}, \mathbf{P}, \tau)$ are fixed, $\mathbf{A}$ is fully deterministic. We then use $\mathbf{A}$ to *compress* both $\mathbf{K}$ and $\mathbf{V}$ in one pass:

$$\widetilde{\mathbf{K}} \triangleq \mathbf{A}^\top\mathbf{K} \in \mathbb{R}^{M\times d_k}, \tag{4}$$

$$\widetilde{\mathbf{V}} \triangleq \mathbf{A}^\top\mathbf{V} \in \mathbb{R}^{M\times d_v}. \tag{5}$$

Applying $\mathbf{A}$ identically to both $\mathbf{K}$ and $\mathbf{V}$ is the core of the method. It ensures that each row $(\widetilde{\mathbf{K}}_j, \widetilde{\mathbf{V}}_j)$ is a semantically consistent, reduced-dimension representation of a cluster of original key–value pairs. The resulting compression $(\widetilde{\mathbf{K}}, \widetilde{\mathbf{V}})$ is a drop-in replacement for the originals, enabling subsequent attention to operate on a linearly-sized representation. We remark that the mechanism can be adapted to compress any two of the $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ triple.

2.2.2. Stage II: High-order Feature Interaction

While Stage I yields a compressed sketch $(\widetilde{\mathbf{K}}, \widetilde{\mathbf{V}})$ of length M, this representation may lack the expressivity needed to model complex relationships. Stage II addresses this limitation through two complementary enhancements: (i) enriching each item in the sketch with higher-order interactions via a randomized feature map, and (ii) enabling communication between the M sketch items using a short-sequence mixer. Only the enrichment step employs randomization; the mixer and the final readout remain deterministic components.

Enrichment via Randomized Feature Maps. To jointly process the address $(\widetilde{\mathbf{K}})$ and content $(\widetilde{\mathbf{V}})$ information, we first concatenate them row-wise into a combined representation $\mathbf{U} \in \mathbb{R}^{M\times d_m}$, where $d_m = d_k + d_v$. This matrix undergoes a preprocessing and normalization step:

$$\mathbf{G} \triangleq \psi(\mathbf{U}) \in \mathbb{R}^{M\times d'}, \tag{6}$$

$$\widetilde{\mathbf{G}} \triangleq \mathrm{Norm}_{\varepsilon_g}(\mathbf{G})/\tau_{\mathrm{g}} \in \mathbb{R}^{M\times d'}, \tag{7}$$

where ψ is a row-wise feature map, $\mathrm{Norm}_{\varepsilon_g}$, as defined in Definition A.4, is a stabilized normalization (similar to LayerNorm in Definition A.5), and $\tau_{\mathrm{g}} > 0$ is a temperature parameter. This step ensures numerical stability by bounding the row norms of $\widetilde{\mathbf{G}}$, which is critical for controlling the variance of subsequent randomized sketches.

The core enrichment is achieved by applying a *randomized, higher-order feature map*. For a predefined set of polynomial degrees $\mathcal{K}$ with corresponding learned weights $\{\beta_k\}$ and sketch dimensions $\{D_k\}_{k\in\mathcal{K}}$, we compute for each row $j \in [M]$:

$$\mathbf{v}_j \triangleq \bigoplus_{k\in\mathcal{K}} \big(\beta_k \cdot \mathrm{TS}_k(\widetilde{\mathbf{G}}_{j,:}; D_k)\big) \in \mathbb{R}^{D_{\mathrm{tot}}}, \tag{8}$$

$$D_{\mathrm{tot}} = \sum_k D_k, \tag{9}$$

$$\mathbf{Y}_{\mathrm{enh}}[j,:] \triangleq \mathbf{W}_{\mathrm{out}}\mathbf{v}_j \in \mathbb{R}^d, \tag{10}$$

where TS_k is a TensorSketch transformation (defined in Appendix C) that implicitly approximates a degree-k polynomial kernel, and $\mathbf{W}_{\mathrm{out}}$ is a learnable linear readout. The dimensions $\{D_k\}_{k\in\mathcal{K}}$ serve as per-degree *accuracy knobs*: larger D_k yields a more faithful approximation of k-th order interactions at a modest increase in computational cost.

Interaction via a Short-Sequence Mixer. To let the M enriched items in $\mathbf{Y}_{\mathrm{enh}}$ exchange information, we apply a lightweight mixer that operates *across* the length-M axis (global mixing):

$$\mathbf{Z}_{\mathrm{out}} \triangleq \mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{Y}_{\mathrm{enh}}) \in \mathbb{R}^{M\times d}. \tag{11}$$

Because $M \ll N_k$, this global mixing adds only a small overhead, yet it lets the compressed keys / values coordinate before the final attention readout.

2.2.3. Stage III: Final Output

In the final stage, the compressed representation from Stage II is projected back into the key and value spaces and then passed through the standard attention operator without further approximation: $\mathbf{K}_g \triangleq \mathbf{Z}_{\mathrm{out}}\mathbf{W}_K \in \mathbb{R}^{M\times d_k}$ and $\mathbf{V}_g \triangleq \mathbf{Z}_{\mathrm{out}}\mathbf{W}_V \in \mathbb{R}^{M\times d_v}$. where $\mathbf{W}_K$ and $\mathbf{W}_V$ are trainable projection matrices. The output is computed as:

$$\mathbf{Y}_{\mathrm{PLASH}} = \mathrm{softmax}\Big(\tfrac{1}{\sqrt{d_k}}\mathbf{Q}\mathbf{K}_g^\top,\ \mathrm{dim}=-1\Big)\mathbf{V}_g. \tag{12}$$

Overall, PLASH reduces the dominant cost from $N_q \times N_k$ to $N_q \times M$, achieving linear time complexity while confining all randomness to earlier stages, enabling rigorous error analysis as shown in Section 3. The pseudocode for the whole PLASH block is presented in Algorithm 1 in Appendix D.

3. Error and Complexity Analysis on PLASH

PLASH is designed to concentrate *all* algorithmic randomness into one specific operation in Stage II. This focused design allows us to analyze the impact of randomness and track it through the rest of the deterministic computation, without forming the full $N_q \times N_k$ attention matrix.

Reference. Given $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, the reference is standard scaled dot-product attention $\mathbf{Y}_{\mathrm{soft}} \triangleq \mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V})$.

Where the End-to-end Deviation Comes From. The error $\|\mathbf{Y}_{\text{soft}} - \mathbf{Y}_{\text{PLASH}}\|_F$ has three parts: (i) a deterministic Stage I compression error (Appendix F), (ii) a deterministic Stage II reference mismatch (bias), and (iii) the randomized Stage II sketching error after deterministic post-processing. The main text certifies (iii) and states the combined bound; the appendix provides the Stage I bound and full proofs.

3.1. Stage II: Certifying the Randomized Feature Map

Stage II is the only source of randomness in PLASH. We therefore certify Stage II at the level of Stage II features and Stage II embeddings, and then propagate the bound through the deterministic post-processing pipeline.

Implemented sketch features and a deterministic reference. Let $\widetilde{\mathbf{G}} \in \mathbb{R}^{M\times d'}$ be the normalized features produced in Stage II (cf. (7)). For each $k \in \mathcal{K}$ we interpret $\widetilde{\mathbf{G}}_{j,:}$ as an element of $\mathbb{R}^{D_k}$ using a fixed deterministic convention (e.g., zero-padding when $D_k \geq d'$); this introduces no randomness. The implemented (randomized) degree-k feature is $\mathbf{z}^{\text{ts}}_{j,k} \triangleq TS_k(\widetilde{\mathbf{G}}_{j,:}; D_k) \in \mathbb{R}^{D_k}$. The deterministic reference replaces sketching by exact k-fold circular convolution: $\mathbf{z}^{\text{det}}_{j,k} \triangleq \widetilde{\mathbf{G}}_{j,:} * \cdots * \widetilde{\mathbf{G}}_{j,:} \in \mathbb{R}^{D_k}$. Define the degree-mixture vectors $\mathbf{v}^{\text{ts}}_j \triangleq \bigoplus_{k\in\mathcal{K}}(\beta_k \cdot \mathbf{z}^{\text{ts}}_{j,k})$ and $\mathbf{v}^{\text{det}}_j \triangleq \bigoplus_{k\in\mathcal{K}}(\beta_k \cdot \mathbf{z}^{\text{det}}_{j,k})$,. Also, define the corresponding Stage II embeddings: $\mathbf{Y}_{\text{enh}}[j,:] = \mathbf{W}_{\text{out}}\mathbf{v}^{\text{ts}}_j$, and $\mathbf{Y}^{\text{det}}_{\text{enh}}[j,:] \triangleq \mathbf{W}_{\text{out}}\mathbf{v}^{\text{det}}_j$.

What is certified, and why τ_{g} is a certification knob. The certificate controls $\|\mathbf{z}^{\text{ts}}_{j,k} - \mathbf{z}^{\text{det}}_{j,k}\|_2$ uniformly over all aggregated items j and degrees k. Two parameters appear with distinct roles: the parameter D_k controls the *probability* of the sketch event (larger D_k yields tighter concentration), and the parameter τ_{g} is the Stage II norm-control temperature in (7), which is *deterministic* and monotonically damps *both* the sketched features and the deterministic reference features, thereby tightening the discrepancy bound. This monotone dependence on τ_{g} is exactly what we visualize in the heatmaps in Figure 2.

Theorem 3.1 (Stage II feature discrepancy (uniform, high probability))**.** *Fix $\eta \in (0,1)$ and $\delta \in (0,1)$. Assume the limited-independence condition in Assumption C.2, with the checkable instantiation given in Construction C.3. Choose per-degree budgets $\{\delta_k\}_{k\in\mathcal{K}}$ such that $\sum_{k\in\mathcal{K}}\delta_k \leq \delta$, and choose sketch sizes $D_k \geq 2M/(\eta^2\delta_k)$ for all $k \in \mathcal{K}$. Then, with probability at least $1-\delta$, simultaneously for all $j \in [M]$ and $k \in \mathcal{K}$,*

$$\|\mathbf{z}^{\text{ts}}_{j,k} - \mathbf{z}^{\text{det}}_{j,k}\|_2 \;\leq\; \left(\sqrt{1+\eta} + D_k^{\frac{k-1}{2}}\right)\cdot \tau_{\text{g}}^{-k}. \qquad (13)$$

Consequently, for any target tolerance $\epsilon_{\text{feat}} > 0$, the choice $\tau_{\text{g}} \geq \max_{k\in\mathcal{K}}\left(\sqrt{1+\eta} + D_k^{(k-1)/2}/\epsilon_{\text{feat}}\right)^{1/k}$ implies $\|\mathbf{z}^{\text{ts}}_{j,k} - \mathbf{z}^{\text{det}}_{j,k}\|_2 \leq \epsilon_{\text{feat}}$ uniformly over all (j,k) on the same probability-$1-\delta$ event. □

How to use the Stage II theorem in practice. The proof in Appendix G separates (i) a concentration event controlled by D_k from (ii) deterministic magnitude control via τ_{g}. In the simulated $\mathcal{K} = \{1\}$ setting, this reduces to: choose D_1 once to fix the failure probability, and tune τ_{g} iteratively.

3.2. End-to-end certificate: from Stage II to the output

The randomness in PLASH is intentionally isolated to Stage II. Every computation that follows is fully deterministic, depending only on the given inputs and learned parameters. This clear separation allows us to view the model in two distinct parts: a randomized feature transformation in Stage II, followed by a deterministic sequence of operations.

Theorem 3.2 (Forward-pass computable $(\epsilon_{\text{out}}, \delta)$-certificate)**.** *Fix $\epsilon_{\text{out}} > 0$ and $\delta \in (0,1)$. Let $\mathbf{Y}_{\text{soft}} = \text{Atten}(\mathbf{Q};\mathbf{K},\mathbf{V})$ and let $\mathbf{Y}_{\text{PLASH}}$ be the* PLASH *output on the same $(\mathbf{Q},\mathbf{K},\mathbf{V})$ with degree set $\mathcal{K}$. Assume the Stage II sketch sizes satisfy the sizing rule in Theorem 3.1 (with budgets $\{\delta_k\}$ and total failure probability δ). Then, with probability at least $1-\delta$,*

$$\|\mathbf{Y}_{\text{soft}} - \mathbf{Y}_{\text{PLASH}}\|_F \;\leq\; \epsilon_{\text{I}} \;+\; \epsilon_{\text{det}} \;+\; \epsilon_{\text{II}},$$

where ϵ_{I} is the deterministic Stage I compression error, ϵ_{det} is the deterministic Stage II reference bias induced by the chosen feature set, and ϵ_{II} is the Stage II randomized sketch error after deterministic post-processing. A sufficient condition on τ_{g} that guarantees $\|\mathbf{Y}_{\text{soft}} - \mathbf{Y}_{\text{PLASH}}\|_F \leq \epsilon_{\text{out}}$ is given by (103) in Appendix H. □

Reading the theorem in the simulated regime $\mathcal{K} = \{1\}$. All experiments in Figure 2 use $\mathcal{K} = \{1\}$. In this case, Stage II uses only CountSketch (TensorSketch with $k = 1$), and the deterministic reference reduces to $\mathbf{z}^{\text{det}}_{j,1} = \widetilde{\mathbf{G}}_{j,:}$. Consequently, the end-to-end certificate in Theorem 3.2 can be read as an explicit approximation guarantee for $\mathbf{Y}_{\text{soft}} = \text{Atten}(\mathbf{Q};\mathbf{K},\mathbf{V})$. Moreover, since $D_1^{(1-1)/2} = 1$, (13) becomes

$$\|\mathbf{z}^{\text{ts}}_{j,1} - \widetilde{\mathbf{G}}_{j,:}\|_2 \;\leq\; \left(\sqrt{1+\eta}+1\right)\cdot\tau_{\text{g}}^{-1},\ \forall j \in [M], \quad (14)$$

with probability at least $1-\delta$ under the sizing rule $D_1 \geq 2M/(\eta^2\delta)$ (taking $\delta_1 = \delta$). This is the precise sense in which τ_{g} acts as a deterministic knob in the simulations: increasing τ_{g} monotonically tightens the Stage II discrepancy bound while leaving the downstream architecture unchanged. At the same time, increasing M typically reduces ϵ_{I} (compression becomes less lossy), increasing the room available for certification (see Appendix F for a quantitative analysis). These are the monotone trends tested empirically by the heatmaps in Figure 2.

3.3. Empirical prevalence of the certificate for $\mathcal{K} = \{1\}$

Certification rate. We set $\mathcal{K} = \{1\}$ and measure how often the forward-pass sufficient condition succeeds. Given

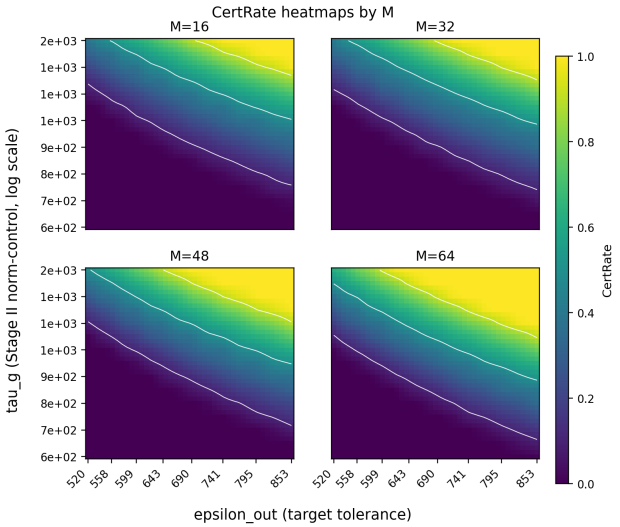


Figure 2. Empirical certification rate for PLASH under $\mathcal{K} = \{1\}$. Each cell is the fraction of trials certified at $(\tau_{\mathrm{g}}, \epsilon_{\mathrm{out}})$.

T independent and identically distributed (i.i.d.) draws of $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$, we define $\mathrm{CertRate}(\epsilon_{\mathrm{out}})$ as the fraction of trials for which Theorem 3.2 certifies $\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{PLASH}}\|_F \leq \epsilon_{\mathrm{out}}$. Concretely, a trial is *certified* when the forward-pass slack is positive and the sufficient inequality (103) in Appendix H holds; this follows the standard practice of reporting certified rates in robustness certification (e.g., randomized smoothing; (Cohen et al., 2019)). See Appendix H.5 for simulation details.

Figure 2 sweeps $(\tau_{\mathrm{g}}, \epsilon_{\mathrm{out}})$ and reports $\mathrm{CertRate}$ for several values of M. The monotone dependence on τ_{g} follows from (14): increasing τ_{g} deterministically shrinks the Stage II discrepancy bound and thus tightens the certified approximation error. The trend in M is driven by Stage I: larger M typically reduces the deterministic compression error (and thus ϵ_{I}), leaving more room at a fixed ϵ_{out}. Overall, these heatmaps serve as a *non-vacuity diagnostic*: they show, on representative inputs, how often the theory's explicit knobs yield a *nontrivial* (i.e., triggered) forward-pass certificate.

3.4. Complexity: bottleneck and overhead

PLASH divides the standard quadratic-cost interaction (of size $N_q \times N_k$) into two smaller, structured operations: one of size $N_k \times M$ (Stage I) and another of size $N_q \times M$ (Stage III). The intermediate Stage II computations depend only on M, which is much smaller than the sequence length. This separation makes the overall scaling explicit: the dominant cost grows linearly with sequence length, and any additional overhead from improving approximation accuracy is controlled only by the internal sketch dimensions.

Theorem 3.3 (Baseline complexity of Stages I–III)**.** *Consider Algorithm 1 with length $M \ll N_k$. Assume Stage II implements TensorSketch via CountSketch plus Fast Fourier Transform (FFT)-based convolutions. If $(d, d', d_k, d_v, \mathcal{K}, \{D_k\}_{k\in\mathcal{K}}, L_{\mathrm{mix}})$ are fixed independently of (N_q, N_k), then the run time satisfies $T_{\mathrm{PLASH}} = O\big((N_q + N_k)\cdot M\big)$. If we store $\mathbf{A} \in \mathbb{R}^{N_k \times M}$ and the readout weights in $\mathbb{R}^{N_q \times M}$, then the peak memory is $O\big((N_q + N_k)\cdot M\big)$; both matrices can be avoided by streaming.* □

The proof of Theorem 3.3 is in Appendix I. Imposing a $(\epsilon_{\mathrm{out}}, \delta)$-certificate in Theorem 3.2 does not alter the $O(N_k M)$ and $O(N_q M)$ structure of Stages I and III; it only constrains the sketch sizes $\{D_k\}_{k\in\mathcal{K}}$, which increases Stage II cost through the $D_k \log D_k$ FFT term and the D_{tot} readout term. The parameter τ_{g} tightens the certified error bounds but does not change arithmetic complexity.

4. Experiments with $\mathcal{K} = \{1, 2\}$

Task and benchmark protocol. We study long-sequence time-series forecasting (LSTF) on the standard ETT (ETTh1 / ETTh2 / ETTm1), ECL, and Weather benchmarks, using the multivariate setting and the evaluation protocol popularized in Informer and later adopted by comprehensive efficient-attention benchmarking suites (*same data splits and metrics*) (Zhou et al., 2021; Zhang et al., 2023). We report MSE and MAE on the test set, and we average results over all prediction horizons used by the protocol.

Backbone and replacement rule. We use Informer (Zhou et al., 2021) as the backbone and replace *only* its attention block with PLASH, keeping the embedding layers, feed-forward sublayers, normalization, and training recipe unchanged. This isolates the effect of the attention mechanism and makes the comparison directly attributable to the proposed $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ block.

Baselines. We compare against recent efficient attention families: (i) sparse / windowed attention (Local; cf. sparse / local designs in (Child et al., 2019; Beltagy et al., 2020; Zaheer et al., 2020)), (ii) random feature / kernel linearization (Performer) proposed in (Choromanski et al., 2021) and CosFormer (Qin et al., 2022)), respectively; (iii) randomized estimators designed for improved softmax fidelity (LARA) proposed in (Zheng et al., 2022)), (iv) low-rank approximations (Nystromformer) proposed in (Xiong et al., 2021)), (v) bounded-context attention (ABC) proposed in (Peng et al., 2022)), (vi) hybrid local + global attention (LongShort) proposed in (Zhu et al., 2021)), and (vii) state-space sequence layers (S4D) proposed in (Gu et al., 2022b)). We also include Vanilla (standard softmax attention) as a reference.

Hyper-parameters. Unless noted otherwise, we follow the Informer training setup (Zhou et al., 2021): batch size 32, 6 epochs, Adam (betas $= (0.9, 0.999)$), peak learning rate $= 1 \times 10^{-4}$ with exponential decay, attention dropout $= 0.05$, weight decay $= 0.0$, no gradient clipping, and no warm-up steps. For Weather, we cap the number of tokens

Table 1. Performance Comparison of Methods Across Multiple Datasets. The top-performing and the second-best methods in each dataset are highlighted in red and blue, respectively. Rankings are determined primarily by MSE, with MAE used as a secondary metric.

Method	ECL		WTH		ETTh1		ETTh2		ETTm1	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Vanilla	0.2370	0.3452	0.3121	0.3616	0.5553	0.5480	0.9789	0.7961	0.3318	0.3653
ABC	0.2458	0.3527	0.3192	0.3761	0.6870	0.6361	1.0734	0.8173	0.3636	0.4037
Performer	0.2548	0.3629	0.3075	0.3618	0.5065	0.5192	1.0104	0.8037	0.3124	0.3671
Local	0.2542	0.3654	0.3185	0.3591	0.6091	0.5824	1.3138	0.9227	0.3144	0.3673
Nystromformer	0.2474	0.3520	0.3141	0.3721	0.5752	0.5601	0.9888	0.8169	0.3691	0.3875
LARA	0.2839	0.3824	0.3104	0.3656	0.7711	0.6677	1.0763	0.8273	0.3906	0.4114
CosFormer	0.2775	0.3848	0.3394	0.3862	0.4965	0.5242	1.1316	0.8790	0.4014	0.4437
LongShort	0.2708	0.3844	0.3413	0.4007	0.4125	0.4399	0.5683	0.6067	0.3869	0.4248
S4D	0.2480	0.3559	0.3129	0.3570	0.5038	0.5080	1.2196	0.8994	0.3508	0.3879
PLASH (M=64)										
D_k= 64	0.2576	0.3625	0.3166	0.3642	0.5667	0.5606	0.6937	0.6821	0.3112	0.3855
D_k=128	0.2430	0.3527	0.3137	0.3650	0.5747	0.5585	1.0193	0.7976	0.3217	0.3669
D_k=256	0.2482	0.3561	0.3082	0.3595	0.5075	0.5027	0.9820	0.7935	0.3314	0.3896
D_k=512	0.2483	0.3561	0.3106	0.3570	0.5075	0.5027	0.9667	0.8158	0.3467	0.3919
PLASH (M=128)										
D_k=64	0.2435	0.3527	0.3057	0.3596	0.5636	0.5684	0.8993	0.7624	0.3292	0.3692
D_k=128	0.2473	0.3537	0.3061	0.3580	0.4974	0.5089	0.7668	0.7271	0.3226	0.3872
D_k=256	0.2453	0.3568	0.3096	0.3650	0.5074	0.5026	0.7328	0.6803	0.3083	0.3685
D_k=512	0.2444	0.3543	0.3009	0.3507	0.5508	0.5497	0.5236	0.5790	0.3280	0.3917
PLASH (M=256)										
D_k=64	0.2474	0.3520	0.3141	0.3721	0.5199	0.5236	0.9888	0.8169	0.3691	0.3875
D_k=128	0.2596	0.3700	0.3072	0.3621	0.4711	0.4872	0.7771	0.7045	0.3089	0.3657
D_k=256	0.2496	0.3565	0.3115	0.3648	0.5074	0.5027	0.8308	0.7167	0.3156	0.3816
D_k=512	0.2484	0.3590	0.3058	0.3638	0.4553	0.4960	0.6421	0.6312	0.3047	0.3649
PLASH (M=512)										
D_k=64	0.2546	0.3632	0.3067	0.3558	0.4769	0.5055	0.6989	0.6849	0.2945	0.3616
D_k=128	0.2478	0.3567	0.3009	0.3492	0.4873	0.5024	0.5140	0.5686	0.3057	0.3682
D_k=256	0.2578	0.3668	0.3015	0.3510	0.4722	0.5072	0.5987	0.6010	0.2896	0.3550
D_k=512	0.2632	0.3737	0.3081	0.3572	0.4725	0.4944	0.8952	0.7459	0.3452	0.4021

per batch iteration to 3 to fit the same compute budget.

4.1. Forecasting accuracy

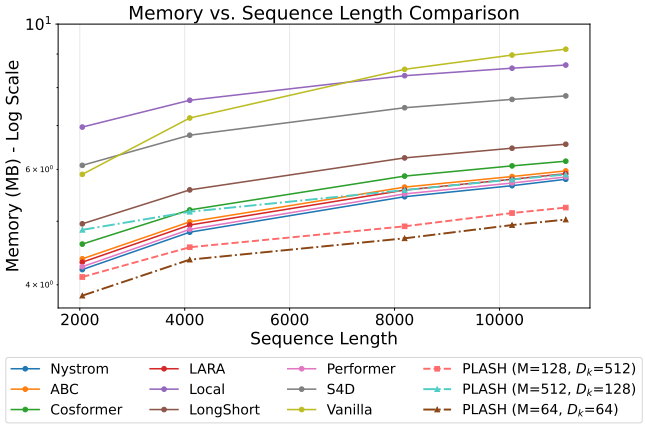
Main result. Table 1 shows that PLASH is consistently competitive with strong efficient-attention baselines, achieving the best or second-best result on most datasets. This supports the PLASH design: compress $(\mathbf{K}, \mathbf{V})$ to length $M \ll N_k$, enrich the compressed representation with selective higher-order feature sketches, and keep the final softmax readout from $\mathbf{Q}$ to $(\mathbf{K}_g, \mathbf{V}_g)$ exact.

Where PLASH wins. PLASH sets the best WTH score (MSE 0.3009, MAE 0.3492) and yields strong improvements on ETTh2 and ETTm1. On ECL and ETTh1, full attention / LongShort remain best, and PLASH is close and typically ranks second. Concretely: on WTH, PLASH with $M = 512$ and $D_k = 128$ is best overall (beating Performer); on ETTh2, PLASH with $M = 128$ and $D_k = 512$ improves over the best baseline; on ETTm1, PLASH with $M = 512$ and $D_k = 256$ is best overall. These gains are consistent with (i) compressed KV summaries that reduce irrelevant token mixing and (ii) higher-order enrichment that restores expressive interactions.

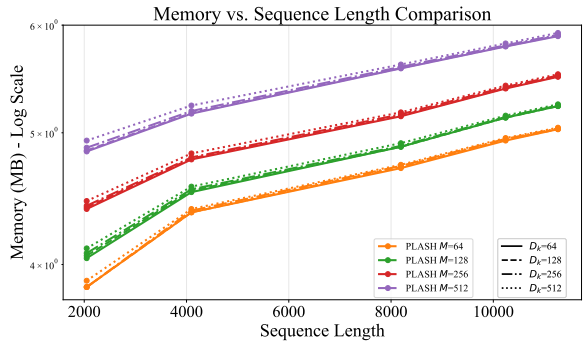
Why accuracy is not monotone in M and D_k. PLASH has two main knobs: M (compressed KV length) and D_k (sketch size). Test accuracy need not improve monotonically with either knob because they change both approximation error and training dynamics. Smaller D_k produces noisier sketches; this can help as implicit regularization or hurt by perturbing gradients, so larger D_k does not guarantee better generalization (Pham & Pagh, 2013; Woodruff, 2014). Increasing M reduces compression bias, but it also reduces averaging within each compressed item, which can make training more sensitive to noise and routing changes (Wang et al., 2020; Xiong et al., 2021). Under a fixed backbone and training schedule, (M, D_k) therefore behaves like a standard hyperparameter choice, and the best setting is dataset-dependent (Tay et al., 2022).

4.2. Storage and run-time scaling

Protocol. We evaluate efficiency using the standard attention-layer benchmark protocol used in prior efficient-attention comparisons (Zhang et al., 2023). We feed synthetic sequences of lengths $\{2048, 4096, 8192, 10240, 11264\}$ into each attention block, with embedding dimension 512, 4 heads, and batch size 1. We run 100 forward passes per configuration and report average latency and peak storage.



(a) Peak storage across methods.



(b) Peak storage of PLASH across M and D_k.

Figure 3. Peak storage comparisons.

Peak storage. Figure 3 shows that Vanilla storage grows quickly with sequence length, while PLASH stays substantially lower and is comparable to other efficient baselines at long contexts. This matches the PLASH design: it replaces the dense $N_q \times N_k$ interaction by two rectangular interfaces, $N_k \times M$ (Stage I) and $N_q \times M$ (Stage III), plus Stage II work that depends only on M and $\{D_k\}$ (Theorem 3.3). Within PLASH, M is the primary storage knob, while $\{D_k\}_{k\in\mathcal{K}}$ adds a smaller overhead from the feature sketches.

Run-time scaling. Table 2 shows the practical impact of removing the quadratic $N_q N_k$ term (Theorem 3.3). Note that the unit of the run-time is milliseconds. As sequence length grows from 2048 to 11264, Vanilla latency rises up from 2.43 to 58.95, and other length-sensitive baselines such as Local (4.72 $\rightarrow$ 25.58) and S4D (3.79 $\rightarrow$ 31.22) also grow sharply. In contrast, PLASH stays nearly flat for small M: with $M = 64$ (any D_k shown), latency stays around 2.4–2.8 across all lengths, giving an $\approx 25\times$ speedup over Vanilla at 11264. Even with $M = 512$, PLASH grows gently (about 2.3 $\rightarrow$ 4.9), and it is competitive with the fastest efficient-attention baselines at long context (e.g., comparable to ABC at 11264). These results indicate that PLASH delivers long-context speedups in wall-clock time while retaining the accuracy gains reported in Section 4.1.

Table 2. Run-time Comparison across Sequence Length. The fastest and the second-fastest methods at each sequence length are highlighted in red and blue, respectively.

Method	Sequence Length				
	2048	4096	8192	10240	11264
Nystrom	3.55	3.50	4.27	4.09	4.56
ABC	1.12	1.65	3.00	4.07	4.35
Cosformer	2.22	2.84	5.12	6.33	6.95
LARA	2.18	2.33	4.11	5.12	5.65
Local	4.72	9.34	18.60	23.24	25.58
LongShort	2.58	3.92	7.40	9.21	10.19
Performer	1.79	1.92	4.45	5.52	6.10
S4D	3.79	8.00	16.34	24.68	31.22
Vanilla	2.43	8.33	31.84	48.97	58.95
PLASH (M=64)					
D_k= 64	2.49	2.52	2.71	2.51	2.69
D_k=128	2.49	2.56	2.42	2.62	2.36
D_k=256	2.59	2.59	2.65	2.79	2.63
D_k=512	2.43	2.51	2.76	2.66	2.39
PLASH (M=128)					
D_k= 64	2.72	2.78	3.11	2.62	3.08
D_k=128	2.44	2.56	2.57	2.48	2.74
D_k=256	2.61	2.54	2.39	2.61	2.45
D_k=512	2.74	2.99	3.17	2.88	2.63
PLASH (M=256)					
D_k= 64	2.87	2.98	2.78	3.18	2.91
D_k=128	2.77	2.93	2.92	2.76	2.87
D_k=256	2.65	2.73	2.63	2.93	2.91
D_k=512	2.40	2.91	2.78	2.82	2.74
PLASH (M=512)					
D_k= 64	2.38	2.86	3.63	4.51	4.89
D_k=128	2.56	2.56	3.62	4.44	4.90
D_k=256	2.34	2.72	3.60	4.52	4.89
D_k=512	2.34	2.80	3.94	4.55	4.94

Small non-monotone fluctuations across (M, D_k) are expected from shape-dependent GPU kernel selection and FFT padding, even when arithmetic scaling is monotone (Dao et al., 2022; NVIDIA, 2023; 2024; 2025).

5. Conclusion

We presented PLASH, a drop-in $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ attention block that avoids the quadratic $O(N_q N_k)$ cost of softmax attention by compressing only the key / value side to $M \ll N_k$, enriching the compressed keys / values with selective higher-order feature sketches, and then performing an *exact* softmax readout from $\mathbf{Q}$ to $(\mathbf{K}_g, \mathbf{V}_g)$. By localizing all algorithmic randomness to a single enrichment module, PLASH enables instance-wise, forward-pass deviation certificates.

On long-sequence forecasting benchmarks, PLASH delivers strong long-context accuracy with substantially milder latency and storage growth than Vanilla attention, while remaining competitive with efficient-attention baselines. This complements faster exact-attention kernels such as FlashAttention (Dao et al., 2022) by providing an algorithmic route to long-context efficiency. More broadly, PLASH suggests a simple design principle: keep the final readout exact, expose clear accuracy–efficiency knobs, and certify the only randomized component.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning by improving the efficiency of attention mechanisms for long-context modeling. The primary expected benefit is reduced computation and energy use for training and inference, which may lower the cost of deploying long-context models. As with many advances in efficient modeling, the technique could also be used to scale up applications that process sensitive data; responsible use should therefore follow standard practices for privacy, security, and data governance. We do not anticipate additional ethical concerns beyond those already well established for deploying large-scale sequence models.

References

Ahle, T. D., Kapralov, M., Knudsen, J. B. T., Pagh, R., Velingker, A., Woodruff, D. P., and Zandieh, A. Oblivious sketching of high-degree polynomial kernels. In *Proceedings of the Thirty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*, pp. 141–160, Salt Lake City, UT, United States, 2020. SIAM.

Anthropic. Anthropic API documentation: Context window. Technical report, Anthropic, 2024.

Arthur, D. and Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, pp. 1027–1035, New Orleans, LA, United States, 2007. SIAM.

Avron, H., Nguyen, H. L., and Woodruff, D. P. Subspace embeddings for the polynomial kernel. In *Advances in Neural Information Processing Systems (Neural Information Processing Systems (NeurIPS) 2014)*, Montréal, Quebec, Canada, 2014. Curran Associates, Inc.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *CoRR*, abs/1607.06450, 2016.

Beltagy, I., Peters, M. E., and Cohan, A. Longformer: The long-document transformer. *CoRR*, abs/2004.05150, 2020.

Bishop, C. M. *Pattern Recognition and Machine Learning*. Springer, 2006.

Charikar, M., Chen, K., and Farach-Colton, M. Finding frequent items in data streams. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP 2002)*, Lecture Notes in Computer Science, pp. 693–703, Málaga, Spain, 2002. Springer.

Child, R., Gray, S., Radford, A., and Sutskever, I. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509, 2019.

Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlós, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, Ł., Belanger, D., Colwell, L., and Weller, A. Rethinking attention with performers. In *Proceedings of the International Conference on Learning Representations (ICLR 2021)*, Virtual, 2021.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320, Long Beach, CA, United States, 2019. PMLR.

Cooley, J. W. and Tukey, J. W. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301, 1965.

Dao, T., Fu, D. Y., Ermon, S., Rudra, A., and Ré, C. Flashattention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (Neural Information Processing Systems (NeurIPS) 2022)*, New Orleans, LA, United States, 2022. Curran Associates, Inc.

De, S., Smith, S. L., Fernando, A., Botev, A., Muraru, G., Gu, A., Haroun, R., Berrada, L., et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *CoRR*, abs/2402.19427, 2024.

Fukui, A., Park, D. H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP 2016)*, pp. 457–468, Austin, TX, United States, 2016. Association for Computational Linguistics.

Gao, Y., Beijbom, O., Zhang, N., and Darrel, T. Compact bilinear pooling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pp. 317–326, Las Vegas, NV, United States, 2016. IEEE Computer Society.

Gonzalez, T. F. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38: 293–306, 1985.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, Philadelphia, PA, United States, 2024.

Gu, A., Goel, K., and Ré, C. Efficiently modeling long sequences with structured state spaces. In *Proceedings of the International Conference on Learning Representations (ICLR 2022)*, Virtual, 2022a.

Gu, A., Gupta, A., Goel, K., and Ré, C. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems (Neural Information Processing Systems (NeurIPS) 2022)*, New Orleans, LA, United States, 2022b. Curran Associates, Inc.

Hinton, G. E., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

Kacham, P., Mirrokni, V., and Zhong, P. Polysketchformer: Fast transformers via sketching polynomial kernels. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, Proceedings of Machine Learning Research, Vienna, Austria, 2024. PMLR.

Katharopoulos, A., Vyas, A., Pappas, N., and Fleuret, F. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, pp. 5156–5165, Virtual, 2020. PMLR.

Kitaev, N., Kaiser, Ł., and Levskaya, A. Reformer: The efficient transformer. In *Proceedings of the International Conference on Learning Representations (ICLR 2020)*, Virtual, 2020.

Lee-Thorp, J., Ainslie, J., Eckstein, I., and Ontañón, S. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 4296–4313, Seattle, United States, July 2022. Association for Computational Linguistics.

Lloyd, S. P. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

Matoušek, J. *Lectures on Discrete Geometry*. Springer, 2002.

NVIDIA. Matrix multiplication background user's guide. Technical report, NVIDIA, 2023.

NVIDIA. cuFFT documentation. Technical report, NVIDIA, 2024.

NVIDIA. Cutlass profiler documentation. Technical report, NVIDIA, 2025.

Peng, H., Pappas, N., Yogatama, D., Schwartz, R., Smith, N., and Kong, L. Random feature attention. In *Proceedings of the International Conference on Learning Representations (ICLR 2021)*, Virtual, 2021.

Peng, H., Kasai, J., Pappas, N., Yogatama, D., Wu, Z., Kong, L., Schwartz, R., and Smith, N. A. ABC: Attention with bounded-memory control. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL 2022)*, Dublin, Ireland, 2022.

Pham, N. and Pagh, R. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2013)*, pp. 239–247, Chicago, IL, United States, 2013. ACM.

Poli, M., Massaroli, S., Nguyen, E., Fu, D. Y., Dao, T., Baccus, S., Bengio, Y., Ermon, S., and Ré, C. Hyena hierarchy: Towards larger convolutional language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, Proceedings of Machine Learning Research, pp. 28043–28078, Honolulu, HI, United States, 2023. PMLR.

Qin, Z., Sun, W., Deng, H., Li, D., Wei, Y., Lv, B., Yan, J., Kong, L., and Zhong, Y. cosformer: Rethinking softmax in attention. In *Proceedings of the International Conference on Learning Representations (ICLR 2022)*, Virtual, 2022.

Schölkopf, B. and Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, 2002.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

Stinson, D. R. Universal hashing and authentication codes. *Designs, Codes and Cryptography*, 4:369–380, 1994.

Tay, Y., Dehghani, M., Abnar, S., Shen, Y., Bahri, D., Pham, P., Rao, J., Yang, L., Ruder, S., and Metzler, D. Long range arena: A benchmark for efficient transformers. In *Proceedings of the International Conference on Learning Representations (ICLR 2021)*, Virtual, 2021.

Tay, Y., Dehghani, M., Bahri, D., and Metzler, D. Efficient transformers: A survey. *ACM Comput. Surv.*, 55(6), December 2022.

Tchebychef, P.-L. Des valeurs moyennes. *Journal de Mathématiques Pures et Appliquées*, 12:177–184, 1867.

van den Oord, A., Vinyals, O., and Kavukcuoglu, K. Neural discrete representation learning. In *Advances in Neural Information Processing Systems (Neural Information Processing Systems (NeurIPS) 2017)*, Long Beach, CA, United States, 2017. Curran Associates, Inc.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems (Neural Information Processing Systems (NeurIPS) 2017)*, Long Beach, CA, United States, 2017. Curran Associates, Inc.

Vershynin, R. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., and Ma, H. Linformer: Self-attention with linear complexity. *CoRR*, abs/2006.04768, 2020.

Woodruff, D. P. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Xiong, Y., Zeng, Z., Chakraborty, R., Tan, M., Fung, G., Chang, Y.-W., and Singh, V. Nyströmformer: A Nyström-based algorithm for approximating self-attention. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021)*, pp. 14138–14148, Virtual, 2021. AAAI Press.

Yang, S., Wang, B., Shen, Y., Panda, R., and Kim, Y. Gated linear attention transformers with hardware-efficient training. In *Proceedings of the 41st International Conference on Machine Learning (ICML 2024)*, Proceedings of Machine Learning Research, Vienna, Austria, 2024. PMLR.

Zaheer, M., Guruganesh, G., Dubey, K. A., Ainslie, J., Alberti, C., Ontanon, S., Pham, P., Ravula, A., Wang, Q., Yang, L., and Ahmed, A. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems (Neural Information Processing Systems (NeurIPS) 2020)*, pp. 17283–17297. Curran Associates, Inc., 2020.

Zhang, J., Jiang, S., Feng, J., Zheng, L., and Kong, L. CAB: Comprehensive attention benchmarking on long sequence modeling. In *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*, Proceedings of Machine Learning Research, Honolulu, HI, United States, 2023. PMLR.

Zheng, L., Wang, C., and Kong, L. Linear complexity randomized self-attention mechanism. In *Proceedings of the 39th International Conference on Machine Learning (ICML 2022)*, Proceedings of Machine Learning Research, pp. 27011–27041, Baltimore, MD, United States, 2022. PMLR.

Zheng, L., Yuan, J., Wang, C., and Kong, L. Efficient attention via control variates. In *The Eleventh International Conference on Learning Representations (ICLR 2023)*, Kigali, Rwanda, 2023.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2021)*, pp. 11106–11115, Virtual, 2021. AAAI Press.

Zhu, C., Ping, W., Xiao, C., Shoeybi, M., Goldstein, T., Anandkumar, A., and Catanzaro, B. Long-short transformer: Efficient transformers for language and vision. In *Advances in Neural Information Processing Systems (Neural Information Processing Systems (NeurIPS) 2021)*, Virtual, 2021. Curran Associates, Inc.

A. Basic operators and conventions

We collect basic notation and operators used throughout the paper. The goal is twofold: (i) to make the presentation self-contained and easy to scan, and (ii) to fix conventions so that later bounds can be read without ambiguity.

Index set. For any positive integer $M \in \mathbb{Z}_{>0}$, define

$$[M] \triangleq \{1, 2, \ldots, M\}.$$

Definition A.1 (Concatenation $\bigoplus$)**.** For vectors $\mathbf{a} \in \mathbb{R}^{m_1}$ and $\mathbf{b} \in \mathbb{R}^{m_2}$, define the concatenation operator

$$\mathbf{a} \bigoplus \mathbf{b} \triangleq \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \end{bmatrix} \in \mathbb{R}^{m_1+m_2}.$$

For matrices with the same number of rows, concatenation is applied row-wise: if $\mathbf{A} \in \mathbb{R}^{M \times m_1}$ and $\mathbf{B} \in \mathbb{R}^{M \times m_2}$, then $\mathbf{A} \bigoplus \mathbf{B} \in \mathbb{R}^{M \times (m_1+m_2)}$ has i-th row $(\mathbf{A}_{i,:} \bigoplus \mathbf{B}_{i,:})^\top$. △

Norm conventions. For a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$ and a vector $\mathbf{u} \in \mathbb{R}^m$, we use:

$$\|\mathbf{X}\|_{2,\infty} \triangleq \max_{i \in [m]} \|\mathbf{X}_{i,:}\|_2, \qquad \|\mathbf{u}\|_\infty \triangleq \max_{i \in [m]} |u_i|,$$

$$\|\mathbf{X}\|_F \triangleq \left(\sum_{i=1}^{m} \sum_{j=1}^{n} X_{ij}^2 \right)^{1/2}.$$

Here $\|\mathbf{X}\|_{2,\infty}$ is the maximum row ℓ_2 norm (useful for uniform, row-wise control), and $\|\mathbf{X}\|_F$ is the Frobenius norm.

Definition A.2 (Temperature-scaled row softmax)**.** Let $\mathrm{softmax}(\cdot)$ denote the *row-wise* softmax. For $\tau > 0$ and a matrix $\mathbf{S} \in \mathbb{R}^{n \times m}$, define the temperature-scaled row softmax by

$$\mathrm{softmax}(\mathbf{S}/\tau, \mathrm{dim} = -1) \triangleq \mathrm{softmax}(\mathbf{S}/\tau) \quad \text{(applied independently to each row)}.$$

The temperature τ controls assignment sharpness: smaller τ yields more peaked row distributions, while larger τ yields smoother assignments. Temperature scaling is standard in differentiable routing and distillation (Bishop, 2006; Goodfellow et al., 2016; Hinton et al., 2015). △

Definition A.3 (spectral/operator norm)**.** For any $\mathbf{W} \in \mathbb{R}^{N \times N}$ and any $N \in \mathbb{Z}_{>0}$, the spectral (operator) norm induced by the Euclidean norm is

$$\|\mathbf{W}\|_{\mathrm{op}} \triangleq \sup_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{W}\mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \sup_{\|\mathbf{x}\|_2 = 1} \|\mathbf{W}\mathbf{x}\|_2.$$

Equivalently,

$$\|\mathbf{W}\|_{\mathrm{op}} = \sqrt{\lambda_{\max}(\mathbf{W}^\top \mathbf{W})},$$

which is the largest singular value of $\mathbf{W}$. △

Definition A.4 (Pre-map and stabilized row normalization)**.** Fix $d_m, d' \in \mathbb{Z}_{\geq 1}$. A *pre-map* is a differentiable function $\psi : \mathbb{R}^{d_m} \to \mathbb{R}^{d'}$ applied row-wise to matrices in $\mathbb{R}^{M \times d_m}$.

Given $\varepsilon_g > 0$, define the stabilized normalization map $\mathrm{Norm}_{\varepsilon_g} : \mathbb{R}^{d'} \to \mathbb{R}^{d'}$ by

$$\mathrm{Norm}_{\varepsilon_g}(\mathbf{g}) \triangleq \frac{\mathbf{g}}{\max\{\|\mathbf{g}\|_2, \varepsilon_g\}}.$$

Thus $\mathrm{Norm}_{\varepsilon_g}$ rescales $\mathbf{g}$ to have ℓ_2 norm at most 1, and it avoids division by very small norms by clipping at ε_g. This uniform norm control is used to bound the magnitude (and hence the variance) of downstream randomized sketches. △

Definition A.5 (Layer normalization)**.** Let $\gamma, \beta \in \mathbb{R}^d$ be trainable parameters and let $\varepsilon_{\ln} > 0$. For $\mathbf{u} \in \mathbb{R}^d$, define

$$\mu(\mathbf{u}) \triangleq \frac{1}{d}\sum_{r=1}^{d} u_r, \qquad \mathrm{Var}(\mathbf{u}) \triangleq \frac{1}{d}\sum_{r=1}^{d}\big(u_r - \mu(\mathbf{u})\big)^2,$$

$$\mathrm{LayerNorm}(\mathbf{u}) \triangleq \gamma \odot \frac{\mathbf{u} - \mu(\mathbf{u})\cdot \mathbf{1}}{\sqrt{\mathrm{Var}(\mathbf{u}) + \varepsilon_{\ln}}} + \beta,$$

where $\odot$ denotes elementwise multiplication and $\mathbf{1} \in \mathbb{R}^d$ is the all-ones vector. For $\mathbf{U} \in \mathbb{R}^{M\times d}$, $\mathrm{LayerNorm}(\mathbf{U})$ is applied row-wise. This is the standard LayerNorm (LN) operator (Ba et al., 2016). △

Definition A.6 (TensorSketch primitive (abstract form))**.** Fix $k \in \mathbb{Z}_{\geq 1}$ and $D_k \in \mathbb{Z}_{\geq 1}$. The degree-$k$ TensorSketch map is a randomized feature map

$$\mathrm{TS}_k(\cdot; D_k) : \mathbb{R}^{d'} \to \mathbb{R}^{D_k},$$

implemented via CountSketch hashing and FFT-based convolution (Charikar et al., 2002; Cooley & Tukey, 1965; Pham & Pagh, 2013). Its explicit construction and the guarantees used in this paper are stated in Section C. △

Definition A.7 ($\mathrm{Mixer}_{L_{\mathrm{mix}}}$ (Transformer encoder on length M))**.** Let $\mathbf{Y}^{(0)} \in \mathbb{R}^{M\times d}$ be the input length-M sequence. Fix the number of heads $H \in \mathbb{Z}_{\geq 1}$ and the head width $d_h \in \mathbb{Z}_{\geq 1}$ such that $Hd_h = d$.

For each layer $\ell \in \{1, \ldots, L_{\mathrm{mix}}\}$ and head $h \in \{1, \ldots, H\}$, let $\mathbf{W}_Q^{(\ell,h)}, \mathbf{W}_K^{(\ell,h)}, \mathbf{W}_V^{(\ell,h)} \in \mathbb{R}^{d\times d_h}$ and $\mathbf{W}_O^{(\ell)} \in \mathbb{R}^{d\times d}$ be learnable parameters. Define the multi-head self-attention map $\mathrm{MHSA}^{(\ell)} : \mathbb{R}^{M\times d} \to \mathbb{R}^{M\times d}$ by

$$\mathrm{MHSA}^{(\ell)}(\mathbf{U}) \triangleq \left(\bigoplus_{h=1}^{H} \mathrm{softmax}\left(\frac{1}{\sqrt{d_h}}\cdot\left(\mathbf{U}\cdot\mathbf{W}_Q^{(\ell,h)}\right)\cdot\left(\mathbf{U}\cdot\mathbf{W}_K^{(\ell,h)}\right)^{\top},\ \mathrm{dim} = -1\right)\cdot\left(\mathbf{U}\cdot\mathbf{W}_V^{(\ell,h)}\right)\right)\cdot\mathbf{W}_O^{(\ell)}.$$

Here $\bigoplus_{h=1}^{H}$ denotes concatenation along the feature dimension (so the result has width $Hd_h = d$).

Let $d_{\mathrm{ff}} \in \mathbb{Z}_{\geq 1}$ be the FFN hidden width, and let $\mathbf{W}_1^{(\ell)} \in \mathbb{R}^{d\times d_{\mathrm{ff}}}$, $\mathbf{W}_2^{(\ell)} \in \mathbb{R}^{d_{\mathrm{ff}}\times d}$, $\mathbf{b}_1^{(\ell)} \in \mathbb{R}^{d_{\mathrm{ff}}}$, and $\mathbf{b}_2^{(\ell)} \in \mathbb{R}^{d}$ be learnable parameters. Define the position-wise feed-forward map $\mathrm{FFN}^{(\ell)} : \mathbb{R}^{M\times d} \to \mathbb{R}^{M\times d}$ by

$$\mathrm{FFN}^{(\ell)}(\mathbf{U}) \triangleq \sigma\Big(\mathbf{U}\cdot\mathbf{W}_1^{(\ell)} + \mathbf{1}\cdot\big(\mathbf{b}_1^{(\ell)}\big)^{\top}\Big)\cdot\mathbf{W}_2^{(\ell)} + \mathbf{1}\cdot\big(\mathbf{b}_2^{(\ell)}\big)^{\top}, \tag{15}$$

where $\sigma : \mathbb{R} \to \mathbb{R}$ is applied elementwise.

Activation regularity used later. In the stability lemmas, we will use that σ is Lipschitz (globally or on a bounded interval). Common choices satisfy this: ReLU is globally 1-Lipschitz, $\tanh$ is globally 1-Lipschitz, and $\mathrm{sigmoid}$ is globally $(1/4)$-Lipschitz (Lemma A.11; see also standard deep-learning references such as (Goodfellow et al., 2016; Bishop, 2006)). Smooth activations such as GELU or SiLU are locally Lipschitz on any bounded interval; the corresponding constants will be invoked explicitly in Section A.1.

The post-LayerNorm encoder-layer update uses $\mathrm{LayerNorm}$ from Definition A.5:

$$\hat{\mathbf{Y}}^{(\ell)} \triangleq \mathrm{LayerNorm}\Big(\mathbf{Y}^{(\ell-1)} + \mathrm{MHSA}^{(\ell)}\big(\mathbf{Y}^{(\ell-1)}\big)\Big), \tag{16}$$

$$\mathbf{Y}^{(\ell)} \triangleq \mathrm{LayerNorm}\Big(\hat{\mathbf{Y}}^{(\ell)} + \mathrm{FFN}^{(\ell)}\big(\hat{\mathbf{Y}}^{(\ell)}\big)\Big). \tag{17}$$

The mixer output is

$$\mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{Y}^{(0)}) \triangleq \mathbf{Y}^{(L_{\mathrm{mix}})}.$$

This is the standard Transformer-encoder architecture (Vaswani et al., 2017; Ba et al., 2016). △

A.1. Lipschitz facts for σ

Lemma A.8 (Bounded derivative implies Lipschitzness (scalar))**.** *Let $\sigma : \mathbb{R} \to \mathbb{R}$ be differentiable on an interval $\mathcal{I} \subseteq \mathbb{R}$ and suppose*

$$\sup_{x\in\mathcal{I}} \big|\sigma'(x)\big| \ \leq\ L_\sigma(\mathcal{I}).$$

Then σ is $L_\sigma(\mathcal{I})$-Lipschitz on $\mathcal{I}$, i.e., for all $a, b \in \mathcal{I}$,

$$\big|\sigma(a) - \sigma(b)\big| \;\le\; L_\sigma(\mathcal{I}) \cdot |a - b|.$$

Proof. Fix $a, b \in \mathcal{I}$. If $a = b$, then the inequality holds with equality. Assume $a \neq b$. Since σ is differentiable on $\mathcal{I}$, it is continuous on $\mathcal{I}$. Apply the mean-value theorem to σ on the interval with endpoints a and b: there exists ξ between a and b such that

$$\sigma(a) - \sigma(b) \;=\; \sigma'(\xi) \cdot (a - b).$$

Take absolute values and use the derivative bound (noting that $\xi \in \mathcal{I}$):

$$\big|\sigma(a) - \sigma(b)\big| \;=\; \big|\sigma'(\xi)\big| \cdot |a - b| \;\le\; \Big(\sup_{x \in \mathcal{I}} |\sigma'(x)|\Big) \cdot |a - b| \;\le\; L_\sigma(\mathcal{I}) \cdot |a - b|.$$

□

Lemma A.9 (Elementwise activation is row-wise Lipschitz in $\|\cdot\|_{2,\infty}$)**.** *Assume $\sigma : \mathbb{R} \to \mathbb{R}$ is L_σ-Lipschitz on $\mathbb{R}$. Extend σ to $\mathbb{R}^{M \times d}$ elementwise by $\big(\sigma(\mathbf{U})\big)_{i,r} \triangleq \sigma\big(\mathbf{U}_{i,r}\big)$. Then for all $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{M \times d}$,*

$$\|\sigma(\mathbf{U}) - \sigma(\mathbf{V})\|_{2,\infty} \;\le\; L_\sigma \cdot \|\mathbf{U} - \mathbf{V}\|_{2,\infty}.$$

Proof. Fix a row index $i \in [M]$. For each coordinate $r \in [d]$, Lipschitzness gives

$$\big|\sigma(\mathbf{U}_{i,r}) - \sigma(\mathbf{V}_{i,r})\big| \;\le\; L_\sigma \cdot |\mathbf{U}_{i,r} - \mathbf{V}_{i,r}|.$$

Square both sides and sum over $r = 1, \ldots, d$:

$$\begin{aligned}
\|\sigma(\mathbf{U})_{i,:} - \sigma(\mathbf{V})_{i,:}\|_2^2 &= \sum_{r=1}^{d} \big|\sigma(\mathbf{U}_{i,r}) - \sigma(\mathbf{V}_{i,r})\big|^2 \\
&\le \sum_{r=1}^{d} \Big(L_\sigma \cdot |\mathbf{U}_{i,r} - \mathbf{V}_{i,r}|\Big)^2 \\
&= L_\sigma^2 \cdot \|\mathbf{U}_{i,:} - \mathbf{V}_{i,:}\|_2^2.
\end{aligned}$$

Take square roots to obtain $\|\sigma(\mathbf{U})_{i,:} - \sigma(\mathbf{V})_{i,:}\|_2 \le L_\sigma \cdot \|\mathbf{U}_{i,:} - \mathbf{V}_{i,:}\|_2$. Finally, take the maximum over $i \in [M]$ to get the $\|\cdot\|_{2,\infty}$ claim. □

Lemma A.10 (ReLU is 1-Lipschitz)**.** *Let $\mathrm{ReLU}(x) \triangleq \max\{0, x\}$. Then for all $a, b \in \mathbb{R}$,*

$$\big|\mathrm{ReLU}(a) - \mathrm{ReLU}(b)\big| \;\le\; |a - b|.$$

Consequently, the elementwise row-wise ReLU map is 1-Lipschitz in $\|\cdot\|_{2,\infty}$ by Lemma A.9.

Proof. We prove the scalar inequality by cases.

Case 1: $a \ge 0$ and $b \ge 0$. Then $\mathrm{ReLU}(a) = a$ and $\mathrm{ReLU}(b) = b$, so $|\mathrm{ReLU}(a) - \mathrm{ReLU}(b)| = |a - b|$.

Case 2: $a \le 0$ and $b \le 0$. Then $\mathrm{ReLU}(a) = \mathrm{ReLU}(b) = 0$, so the left-hand side is $0 \le |a - b|$.

Case 3: $a \ge 0 \ge b$. Then $\mathrm{ReLU}(a) = a$ and $\mathrm{ReLU}(b) = 0$, hence $|\mathrm{ReLU}(a) - \mathrm{ReLU}(b)| = |a| = a$. Also $|a - b| = a - b \ge a$ since $b \le 0$. Thus $|\mathrm{ReLU}(a) - \mathrm{ReLU}(b)| \le |a - b|$.

Case 4: $b \ge 0 \ge a$. This is symmetric to Case 3. □

Lemma A.11 (Common smooth activations: global Lipschitz constants)**.** *The following global Lipschitz bounds hold:*

$$\begin{aligned}
\big|\tanh(a) - \tanh(b)\big| &\le |a - b|, \\
\big|\mathrm{sigmoid}(a) - \mathrm{sigmoid}(b)\big| &\le \frac{1}{4} \cdot |a - b|, \qquad \mathrm{sigmoid}(x) \triangleq \frac{1}{1 + e^{-x}}.
\end{aligned}$$

Proof. Step 1 (reduce to a derivative bound). By Lemma A.8, it suffices to bound $\sup_{x\in\mathbb{R}} |\tanh'(x)|$ and $\sup_{x\in\mathbb{R}} |\mathrm{sigmoid}'(x)|$.

*Step 2 (*tanh*).* We have $\tanh'(x) = \mathrm{sech}^2(x) = 1/\cosh^2(x)$. Since $\cosh(x) \geq 1$ for all x, we get $0 < \mathrm{sech}^2(x) \leq 1$, hence $\sup_{x\in\mathbb{R}} |\tanh'(x)| \leq 1$. Applying Lemma A.8 with $\mathcal{I} = \mathbb{R}$ yields $|\tanh(a) - \tanh(b)| \leq 1 \cdot |a - b|$.

Step 3 (sigmoid). Let $s(x) \triangleq \mathrm{sigmoid}(x) = 1/(1 + e^{-x})$. Differentiate:

$$s'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) = s(x) \cdot (1 - s(x)).$$

Now let $t \triangleq s(x) \in (0, 1)$. Then $s'(x) = t(1 - t)$. The quadratic $t(1 - t) = t - t^2$ attains its maximum over $t \in [0, 1]$ at $t = 1/2$ with value $1/4$. Therefore $\sup_{x\in\mathbb{R}} |s'(x)| \leq 1/4$. Applying Lemma A.8 with $\mathcal{I} = \mathbb{R}$ yields $|\mathrm{sigmoid}(a) - \mathrm{sigmoid}(b)| \leq (1/4) \cdot |a - b|$.

Remark (standard facts). The derivative formulas and the $1/4$ bound for sigmoid are standard; see, e.g., (Goodfellow et al., 2016; Bishop, 2006). □

B. Local Lipschitzness of the KV mixer in $\|\cdot\|_{2,\infty}$

We study the post-LayerNorm Transformer-encoder mixer $\mathrm{Mixer}_{L_{\mathrm{mix}}}$ from Definition A.7 under the matrix norm

$$\|\mathbf{U}\|_{2,\infty} \triangleq \max_{i\in[M]} \|\mathbf{U}_{i,:}\|_2,$$

i.e., the maximum row-wise ℓ_2 magnitude. This norm is convenient for PLASH because our Stage II certificate naturally controls per-row perturbations, and the mixer is deterministic given inputs and parameters. Hence we can: (i) certify the Stage II perturbation, and (ii) propagate it through the mixer via Lipschitz stability.

Throughout this section we work in inference mode (dropout disabled). If dropout is enabled, all statements below hold conditional on a fixed dropout mask.

The only segment we need. To compare the realized Stage II embedding $\mathbf{Y}_{\mathrm{enh}}$ and its deterministic comparator $\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}$, it suffices to control the mixer on the line segment

$$\mathcal{S} = \left\{\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}} + t \cdot \left(\mathbf{Y}_{\mathrm{enh}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}\right) : \ t \in [0, 1]\right\} \ \subseteq \ \mathbb{R}^{M\times d},$$

which is the same as (82).

Lemma B.1 (Right multiplication is Lipschitz in $\|\cdot\|_{2,\infty}$)**.** *For any compatible matrices $\mathbf{X}, \mathbf{X}'$ and any matrix $\mathbf{W}$,*

$$\left\|\mathbf{X}\cdot\mathbf{W} - \mathbf{X}'\cdot\mathbf{W}\right\|_{2,\infty} \ \leq \ \|\mathbf{W}\|_{\mathrm{op}} \cdot \|\mathbf{X} - \mathbf{X}'\|_{2,\infty},$$

where $\|\mathbf{W}\|_{\mathrm{op}}$ is defined in Definition A.3.

Proof. For each $i \in [M]$,

$$\left\|(\mathbf{X}\cdot\mathbf{W} - \mathbf{X}'\cdot\mathbf{W})_{i,:}\right\|_2 = \left\|(\mathbf{X}_{i,:} - \mathbf{X}'_{i,:})\cdot\mathbf{W}\right\|_2 \ \leq \ \|\mathbf{W}\|_{\mathrm{op}} \cdot \|\mathbf{X}_{i,:} - \mathbf{X}'_{i,:}\|_2.$$

Taking $\max_{i\in[M]}$ yields the claim. □

Lemma B.2 (Softmax is 1-Lipschitz from ℓ_∞ to ℓ_1)**.** *For any $\mathbf{s}, \mathbf{s}' \in \mathbb{R}^N$,*

$$\|\,\mathrm{softmax}(\mathbf{s}) - \mathrm{softmax}(\mathbf{s}')\|_1 \ \leq \ \|\mathbf{s} - \mathbf{s}'\|_\infty. \tag{18}$$

Proof. Let $\mathbf{s}(t) = \mathbf{s}' + t(\mathbf{s} - \mathbf{s}')$ for $t \in [0, 1]$, and define $\mathbf{a}(t) = \mathrm{softmax}(\mathbf{s}(t))$. By the fundamental theorem of calculus,

$$\mathrm{softmax}(\mathbf{s}) - \mathrm{softmax}(\mathbf{s}') = \int_0^1 \mathbf{J}(t)\,(\mathbf{s} - \mathbf{s}')\,dt,$$

where $\mathbf{J}(t)$ is the Jacobian of softmax at $\mathbf{s}(t)$. Fix t and write $\mathbf{a} = \mathbf{a}(t)$. The Jacobian satisfies $\mathbf{J} = \mathrm{Diag}(\mathbf{a}) - \mathbf{a}\mathbf{a}^\top$, so for any $\mathbf{h} \in \mathbb{R}^N$,

$$\mathbf{J}\mathbf{h} = \mathbf{a} \odot \mathbf{h} - (\mathbf{a}^\top \mathbf{h})\mathbf{a} = \mathbf{a} \odot (\mathbf{h} - m\mathbf{1}), \qquad m \triangleq \mathbf{a}^\top \mathbf{h}.$$

Therefore,

$$\|\mathbf{J}\mathbf{h}\|_1 = \sum_{j=1}^{N} \mathbf{a}_j \, |h_j - m|.$$

Assume w.l.o.g. $\|\mathbf{h}\|_\infty \le 1$ (otherwise scale). Define a random variable H that takes value h_j with probability $\mathbf{a}_j$. Then $m = \mathbb{E}[H]$ and $\|\mathbf{J}\mathbf{h}\|_1 = \mathbb{E}[|H - \mathbb{E}[H]|]$. Since $H \in [-1, 1]$, we have $|H - \mathbb{E}[H]| \le 2$ and, more sharply,

$$\mathbb{E}[|H - \mathbb{E}[H]|] \le 1,$$

with equality achieved by a two-point distribution on $\{-1, 1\}$ with mean 0. Hence $\|\mathbf{J}\mathbf{h}\|_1 \le 1$ whenever $\|\mathbf{h}\|_\infty \le 1$, i.e., $\|\mathbf{J}\|_{\infty\to 1} \le 1$. Integrating along t yields (18).

□

Lemma B.3 (Scaled dot-product attention is Lipschitz in each argument under $\|\cdot\|_{2,\infty}$)**.** *Recall*

$$\mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\Big(\tfrac{1}{\sqrt{d_k}}\, \mathbf{Q}\mathbf{K}^\top,\ \mathrm{dim} = -1\Big)\mathbf{V} \in \mathbb{R}^{N_q \times d_v}.$$

Fix $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$ *and* $\mathbf{V}, \mathbf{V}' \in \mathbb{R}^{N \times d_v}$ *and* $\mathbf{K}, \mathbf{K}' \in \mathbb{R}^{N \times d_k}$*. Define*

$$\Gamma_Q \triangleq \frac{\|\mathbf{Q}\|_{2,\infty}}{\sqrt{d_k}}, \qquad \Gamma_V \triangleq \max\{\|\mathbf{V}\|_{2,\infty},\ \|\mathbf{V}'\|_{2,\infty}\}.$$

Then the following deterministic bounds hold:

$$\big\|\mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V}) - \mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V}')\big\|_{2,\infty} \le \|\mathbf{V} - \mathbf{V}'\|_{2,\infty},$$
$$\big\|\mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V}) - \mathrm{Atten}(\mathbf{Q}; \mathbf{K}', \mathbf{V})\big\|_{2,\infty} \le \Gamma_Q \, \Gamma_V \, \|\mathbf{K} - \mathbf{K}'\|_{2,\infty}.$$

Consequently,

$$\big\|\mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V}) - \mathrm{Atten}(\mathbf{Q}; \mathbf{K}', \mathbf{V}')\big\|_{2,\infty} \ \le\ \Gamma_Q \, \Gamma_V \, \|\mathbf{K} - \mathbf{K}'\|_{2,\infty} + \|\mathbf{V} - \mathbf{V}'\|_{2,\infty}.$$

Proof. Write

$$\mathbf{S} = \tfrac{1}{\sqrt{d_k}}\mathbf{Q}\mathbf{K}^\top \in \mathbb{R}^{N_q \times N}, \qquad \mathbf{A} = \mathrm{softmax}(\mathbf{S}, \mathrm{dim} = -1) \in \mathbb{R}^{N_q \times N},$$

and analogously $\mathbf{S}'$ and $\mathbf{A}'$ for $\mathbf{K}'$. Each row of $\mathbf{A}$ and $\mathbf{A}'$ lies in the probability simplex (nonnegative entries summing to 1).

Step 1: Lipschitz in $\mathbf{V}$ *(with* $\mathbf{K}$ *fixed).* Fix $i \in [N_q]$. Since $\mathbf{A}_{i,:}$ is a convex weight vector,

$$\begin{aligned}
\|(\mathbf{A}\mathbf{V} - \mathbf{A}\mathbf{V}')_{i,:}\|_2 &= \Big\|\sum_{j=1}^{N} \mathbf{A}_{i,j}\, (\mathbf{V}_{j,:} - \mathbf{V}'_{j,:})\Big\|_2 \\
&\le \sum_{j=1}^{N} \mathbf{A}_{i,j}\, \|\mathbf{V}_{j,:} - \mathbf{V}'_{j,:}\|_2 \\
&\le\ \|\mathbf{V} - \mathbf{V}'\|_{2,\infty}.
\end{aligned}$$

Taking $\max_{i \in [N_q]}$ gives the first bound.

Step 2: Lipschitz in $\mathbf{K}$ *(with* $\mathbf{V}$ *fixed).* Fix $i \in [N_q]$ and let $\mathbf{s} = \mathbf{S}_{i,:}$ and $\mathbf{s}' = \mathbf{S}'_{i,:}$. For any $j \in [N]$,

$$|\mathbf{s}_j - \mathbf{s}'_j| = \Big|\tfrac{1}{\sqrt{d_k}}\langle \mathbf{Q}_{i,:}, \mathbf{K}_{j,:} - \mathbf{K}'_{j,:}\rangle\Big| \le \tfrac{1}{\sqrt{d_k}}\|\mathbf{Q}_{i,:}\|_2 \, \|\mathbf{K}_{j,:} - \mathbf{K}'_{j,:}\|_2,$$

so

$$\|\mathbf{s}-\mathbf{s}'\|_\infty \le \Gamma_Q \, \|\mathbf{K}-\mathbf{K}'\|_{2,\infty}.$$

Using $\mathbf{AV}-\mathbf{A}'\mathbf{V} = (\mathbf{A}-\mathbf{A}')\mathbf{V}$, Lemma B.2, and $\|\mathbf{V}\|_{2,\infty} \le \Gamma_V$,

$$\begin{aligned}\|((\mathbf{A}-\mathbf{A}')\mathbf{V})_{i,:}\|_2 &\le \|\mathbf{A}_{i,:}-\mathbf{A}'_{i,:}\|_1 \, \|\mathbf{V}\|_{2,\infty} \\ &\le \|\mathbf{s}-\mathbf{s}'\|_\infty \, \Gamma_V \\ &\le \; \Gamma_Q \, \Gamma_V \, \|\mathbf{K}-\mathbf{K}'\|_{2,\infty}.\end{aligned}$$

Taking $\max_{i\in[N_q]}$ yields the second bound. The joint bound follows by triangle inequality. □

Lemma B.4 (Row-wise LayerNorm is locally Lipschitz on a bounded-variance set)**.** *Let* LayerNorm *be as in Definition A.5 with $\varepsilon_{\ln} > 0$ and trainable $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^d$. Fix a set $\mathcal{T} \subseteq \mathbb{R}^d$ and define*

$$m_{\mathcal{T}} \triangleq \inf_{\mathbf{u}\in\mathcal{T}} \big(\mathrm{Var}(\mathbf{u}) + \varepsilon_{\ln}\big) \;\ge\; \varepsilon_{\ln}.$$

Then there exists a deterministic constant $L_{\mathrm{LN}}(\mathcal{T}) > 0$ such that for all $\mathbf{u}, \mathbf{v} \in \mathcal{T}$,

$$\|\mathrm{LayerNorm}(\mathbf{u}) - \mathrm{LayerNorm}(\mathbf{v})\|_2 \;\le\; L_{\mathrm{LN}}(\mathcal{T}) \, \|\mathbf{u}-\mathbf{v}\|_2.$$

One admissible choice is

$$L_{\mathrm{LN}}(\mathcal{T}) \triangleq \|\boldsymbol{\gamma}\|_\infty \cdot \left(\frac{2}{\sqrt{m_{\mathcal{T}}}} + \frac{2}{d} \cdot \frac{\big(\sup_{\mathbf{u}\in\mathcal{T}} \|\mathbf{u}-\mu(\mathbf{u})\mathbf{1}\|_2\big)^2}{m_{\mathcal{T}}^{3/2}} \right).$$

Consequently, if LayerNorm *is applied row-wise to matrices and each row lies in $\mathcal{T}$, then*

$$\|\mathrm{LayerNorm}(\mathbf{U}) - \mathrm{LayerNorm}(\mathbf{V})\|_{2,\infty} \;\le\; L_{\mathrm{LN}}(\mathcal{T}) \, \|\mathbf{U}-\mathbf{V}\|_{2,\infty}.$$

Proof. **Step 0: Rewrite LayerNorm.** Define the centering and inverse-stdev maps

$$\mathbf{c}(\mathbf{u}) \triangleq \mathbf{u} - \mu(\mathbf{u})\mathbf{1}, \qquad \alpha(\mathbf{u}) \triangleq \big(\mathrm{Var}(\mathbf{u}) + \varepsilon_{\ln}\big)^{-1/2}.$$

Then Definition A.5 becomes

$$\mathrm{LayerNorm}(\mathbf{u}) = \boldsymbol{\gamma} \odot \big(\alpha(\mathbf{u})\, \mathbf{c}(\mathbf{u})\big) + \boldsymbol{\beta}.$$

The bias cancels under subtraction and $\|\boldsymbol{\gamma} \odot \mathbf{x}\|_2 \le \|\boldsymbol{\gamma}\|_\infty \|\mathbf{x}\|_2$, so it suffices to bound

$$\big\|\alpha(\mathbf{u})\mathbf{c}(\mathbf{u}) - \alpha(\mathbf{v})\mathbf{c}(\mathbf{v})\big\|_2.$$

Step 1: Centering is 2-Lipschitz. For any $\mathbf{u}, \mathbf{v}$,

$$\mathbf{c}(\mathbf{u}) - \mathbf{c}(\mathbf{v}) = (\mathbf{u}-\mathbf{v}) - \frac{1}{d}\langle \mathbf{1}, \mathbf{u}-\mathbf{v}\rangle \mathbf{1}.$$

Hence, using Cauchy–Schwarz and $\|\mathbf{1}\|_2^2 = d$,

$$\begin{aligned}\|\mathbf{c}(\mathbf{u}) - \mathbf{c}(\mathbf{v})\|_2 &\le \|\mathbf{u}-\mathbf{v}\|_2 + \frac{1}{d} \, |\langle \mathbf{1}, \mathbf{u}-\mathbf{v}\rangle| \, \|\mathbf{1}\|_2 \\ &\le \|\mathbf{u}-\mathbf{v}\|_2 + \frac{1}{d} \, \|\mathbf{1}\|_2^2 \, \|\mathbf{u}-\mathbf{v}\|_2 \\ &= 2\|\mathbf{u}-\mathbf{v}\|_2.\end{aligned}$$

Step 2: Decompose the difference. Add and subtract $\alpha(\mathbf{u})\mathbf{c}(\mathbf{v})$:

$$\|\alpha(\mathbf{u})\mathbf{c}(\mathbf{u}) - \alpha(\mathbf{v})\mathbf{c}(\mathbf{v})\|_2 \le \alpha(\mathbf{u})\|\mathbf{c}(\mathbf{u}) - \mathbf{c}(\mathbf{v})\|_2 + |\alpha(\mathbf{u}) - \alpha(\mathbf{v})| \, \|\mathbf{c}(\mathbf{v})\|_2. \tag{19}$$

Step 3: Bound the first term using $m_{\mathcal{T}}$. For any $\mathbf{u} \in \mathcal{T}$, $\alpha(\mathbf{u}) \le 1/\sqrt{m_{\mathcal{T}}}$ by definition of $m_{\mathcal{T}}$. Combining with Step 1 gives

$$\alpha(\mathbf{u})\|\mathbf{c}(\mathbf{u}) - \mathbf{c}(\mathbf{v})\|_2 \le \frac{2}{\sqrt{m_{\mathcal{T}}}}\|\mathbf{u} - \mathbf{v}\|_2.$$

Step 4: Bound $|\alpha(\mathbf{u}) - \alpha(\mathbf{v})|$ via variance differences. Let $f(x) = x^{-1/2}$ on $[m_{\mathcal{T}}, \infty)$. Then $|f'(x)| = \frac{1}{2}x^{-3/2} \le \frac{1}{2}m_{\mathcal{T}}^{-3/2}$. By the mean value theorem,

$$|\alpha(\mathbf{u}) - \alpha(\mathbf{v})| \le \frac{1}{2}m_{\mathcal{T}}^{-3/2}\,|\operatorname{Var}(\mathbf{u}) - \operatorname{Var}(\mathbf{v})|. \tag{20}$$

Next, since $\operatorname{Var}(\mathbf{u}) = \frac{1}{d}\|\mathbf{c}(\mathbf{u})\|_2^2$,

$$\begin{aligned}
|\operatorname{Var}(\mathbf{u}) - \operatorname{Var}(\mathbf{v})| &= \frac{1}{d}\big|\|\mathbf{c}(\mathbf{u})\|_2^2 - \|\mathbf{c}(\mathbf{v})\|_2^2\big| \\
&= \frac{1}{d}\big|\langle \mathbf{c}(\mathbf{u}) + \mathbf{c}(\mathbf{v}),\ \mathbf{c}(\mathbf{u}) - \mathbf{c}(\mathbf{v})\rangle\big| \\
&\le \frac{1}{d}\|\mathbf{c}(\mathbf{u}) + \mathbf{c}(\mathbf{v})\|_2\,\|\mathbf{c}(\mathbf{u}) - \mathbf{c}(\mathbf{v})\|_2 \\
&\le \frac{2}{d}\Big(\sup_{\mathbf{w}\in\mathcal{T}}\|\mathbf{c}(\mathbf{w})\|_2\Big)\cdot 2\|\mathbf{u} - \mathbf{v}\|_2 \\
&= \frac{4}{d}\Big(\sup_{\mathbf{w}\in\mathcal{T}}\|\mathbf{c}(\mathbf{w})\|_2\Big)\|\mathbf{u} - \mathbf{v}\|_2,
\end{aligned}$$

where we used Step 1 and $\|\mathbf{c}(\mathbf{u}) + \mathbf{c}(\mathbf{v})\|_2 \le 2\sup_{\mathbf{w}\in\mathcal{T}}\|\mathbf{c}(\mathbf{w})\|_2$. Substituting this into (20) yields

$$|\alpha(\mathbf{u}) - \alpha(\mathbf{v})| \le \frac{2}{d}\cdot\frac{\sup_{\mathbf{w}\in\mathcal{T}}\|\mathbf{c}(\mathbf{w})\|_2}{m_{\mathcal{T}}^{3/2}}\,\|\mathbf{u} - \mathbf{v}\|_2. \tag{21}$$

Step 5: Bound the second term and combine. Since $\|\mathbf{c}(\mathbf{v})\|_2 \le \sup_{\mathbf{w}\in\mathcal{T}}\|\mathbf{c}(\mathbf{w})\|_2$, combining with (21) gives

$$|\alpha(\mathbf{u}) - \alpha(\mathbf{v})|\,\|\mathbf{c}(\mathbf{v})\|_2 \le \frac{2}{d}\cdot\frac{\big(\sup_{\mathbf{w}\in\mathcal{T}}\|\mathbf{c}(\mathbf{w})\|_2\big)^2}{m_{\mathcal{T}}^{3/2}}\,\|\mathbf{u} - \mathbf{v}\|_2.$$

Substitute the bounds from Steps 3 and 5 into (19), then multiply by $\|\boldsymbol{\gamma}\|_\infty$. This yields the claimed Lipschitz inequality with the displayed admissible choice of $L_{\mathrm{LN}}(\mathcal{T})$.

Row-wise matrix version. If each row of $\mathbf{U}$ and $\mathbf{V}$ lies in $\mathcal{T}$, apply the vector bound to each row and take the maximum over row indices. This is exactly the $\|\cdot\|_{2,\infty}$ norm. □

Theorem B.5 (Local Lipschitzness of $\mathrm{Mixer}_{L_{\mathrm{mix}}}$ on $\mathcal{S}$ in $\|\cdot\|_{2,\infty}$)**.** *Let $\mathrm{Mixer}_{L_{\mathrm{mix}}}$ be as in Definition A.7. Assume the pointwise nonlinearity σ used in $\mathrm{FFN}^{(\ell)}$ is L_σ-Lipschitz and $\varepsilon_{\mathrm{ln}} > 0$ (see Section A.1). Let $\mathcal{S} \subseteq \mathbb{R}^{M\times d}$ be the segment defined in (82). Then there exists a deterministic constant $L_{\mathrm{mix}}(\mathcal{S}) > 0$ such that for all $\mathbf{U}, \mathbf{U}' \in \mathcal{S}$,*

$$\big\|\mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U}) - \mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U}')\big\|_{2,\infty} \ \le\ L_{\mathrm{mix}}(\mathcal{S})\cdot\|\mathbf{U} - \mathbf{U}'\|_{2,\infty}.$$

Moreover, one admissible choice is

$$L_{\mathrm{mix}}(\mathcal{S}) \triangleq \prod_{\ell=1}^{L_{\mathrm{mix}}} L_\ell(\mathcal{S}),$$

where $L_\ell(\mathcal{S})$ is any Lipschitz constant of the ℓth post-LN encoder layer restricted to the corresponding input segment induced by $\mathcal{S}$.

Proof. **Step 0: Compact segments induced by $\mathcal{S}$.** For $\ell \in \{0, 1, \ldots, L_{\mathrm{mix}}\}$, let $\mathbf{Y}^{(\ell)}$ be the ℓth-layer hidden state produced by the update equations in Definition A.7 starting from $\mathbf{Y}^{(0)} \in \mathcal{S}$, and define

$$\mathcal{S}_0 \triangleq \mathcal{S}, \qquad \mathcal{S}_\ell \triangleq \{\mathbf{Y}^{(\ell)} :\ \mathbf{Y}^{(0)} \in \mathcal{S}\} \subseteq \mathbb{R}^{M\times d}.$$

Each layer map in Definition A.7 is continuous (finite compositions of affine maps, $\mathrm{softmax}(\cdot, \mathrm{dim} = -1)$, σ, and LayerNorm with $\varepsilon_{\mathrm{ln}} > 0$), so $\mathcal{S}_\ell$ is the continuous image of the compact set $\mathcal{S}$ and is therefore compact.

Fix a layer index $\ell \in \{1, \dots, L_{\mathrm{mix}}\}$. Write the ℓth post-LN encoder layer as (cf. Definition A.7)

$$\mathbf{H}^{(\ell)}(\mathbf{Y}) \triangleq \underbrace{\mathrm{LayerNorm}\Big(\mathbf{A} + \mathrm{FFN}^{(\ell)}(\mathbf{A})\Big)}_{\mathbf{G}^{(\ell)}(\mathbf{A})}\Big|_{\mathbf{A}=\mathbf{F}^{(\ell)}(\mathbf{Y})}, \qquad \mathbf{F}^{(\ell)}(\mathbf{Y}) \triangleq \mathrm{LayerNorm}\Big(\mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y})\Big).$$

Then $\mathbf{Y}^{(\ell)} = \mathbf{H}^{(\ell)}(\mathbf{Y}^{(\ell-1)})$.

Step 1: The map $\mathbf{F}^{(\ell)}$ is Lipschitz on $\mathcal{S}_{\ell-1}$. Define the set of row vectors that occur as inputs to the first LayerNorm in layer ℓ:

$$\mathcal{T}_{\ell,F} \triangleq \Big\{(\mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y}))_{i,:} : \ \mathbf{Y} \in \mathcal{S}_{\ell-1}, \ i \in [M]\Big\} \subseteq \mathbb{R}^d.$$

Since $\mathcal{S}_{\ell-1}$ is compact and $\mathbf{Y} \mapsto \mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y})$ is continuous, $\mathcal{T}_{\ell,F}$ is compact; moreover $\mathrm{Var}(\cdot) + \varepsilon_{\mathrm{ln}} \geq \varepsilon_{\mathrm{ln}} > 0$. Thus Lemma B.4 applies row-wise on $\mathcal{T}_{\ell,F}$:

$$\begin{aligned}\|\mathbf{F}^{(\ell)}(\mathbf{Y}) - \mathbf{F}^{(\ell)}(\mathbf{Y}')\|_{2,\infty} &\leq L_{\mathrm{LN}}(\mathcal{T}_{\ell,F}) \left\|(\mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y})) - (\mathbf{Y}' + \mathrm{MHSA}^{(\ell)}(\mathbf{Y}'))\right\|_{2,\infty} \\ &\leq L_{\mathrm{LN}}(\mathcal{T}_{\ell,F}) \Big(\|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty} + \|\mathrm{MHSA}^{(\ell)}(\mathbf{Y}) - \mathrm{MHSA}^{(\ell)}(\mathbf{Y}')\|_{2,\infty}\Big). \end{aligned} \tag{22}$$

To control the MHSA term, decompose it into: (i) linear projections to $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ per head, (ii) head-level attention, and (iii) the output projection. Apply Lemma B.1 to the linear maps $\mathbf{Y} \mapsto \mathbf{Y}\mathbf{W}_Q^{(\ell,h)}$, $\mathbf{Y} \mapsto \mathbf{Y}\mathbf{W}_K^{(\ell,h)}$, and $\mathbf{Y} \mapsto \mathbf{Y}\mathbf{W}_V^{(\ell,h)}$. Then apply Lemma B.3 to each head-level attention map (with $N_q = N = M$ inside the mixer and $d_k = d_h$ per head), and finally apply Lemma B.1 to the output projection $\mathbf{W}_O^{(\ell)}$. This yields a deterministic constant $L_{\mathrm{MHSA}}^{(\ell)}(\mathcal{S}_{\ell-1})$ such that for all $\mathbf{Y}, \mathbf{Y}' \in \mathcal{S}_{\ell-1}$,

$$\|\mathrm{MHSA}^{(\ell)}(\mathbf{Y}) - \mathrm{MHSA}^{(\ell)}(\mathbf{Y}')\|_{2,\infty} \leq L_{\mathrm{MHSA}}^{(\ell)}(\mathcal{S}_{\ell-1}) \, \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}. \tag{23}$$

Substituting (23) into (22) gives

$$\|\mathbf{F}^{(\ell)}(\mathbf{Y}) - \mathbf{F}^{(\ell)}(\mathbf{Y}')\|_{2,\infty} \leq L_{\mathrm{LN}}(\mathcal{T}_{\ell,F}) \left(1 + L_{\mathrm{MHSA}}^{(\ell)}(\mathcal{S}_{\ell-1})\right) \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}. \tag{24}$$

Step 2: The map $\mathbf{G}^{(\ell)}$ is Lipschitz on $\mathbf{F}^{(\ell)}(\mathcal{S}_{\ell-1})$. Define the set of row vectors that occur as inputs to the second LayerNorm in layer ℓ:

$$\mathcal{T}_{\ell,G} \triangleq \Big\{(\mathbf{A} + \mathrm{FFN}^{(\ell)}(\mathbf{A}))_{i,:} : \ \mathbf{A} \in \mathbf{F}^{(\ell)}(\mathcal{S}_{\ell-1}), \ i \in [M]\Big\} \subseteq \mathbb{R}^d.$$

The set $\mathbf{F}^{(\ell)}(\mathcal{S}_{\ell-1})$ is compact by continuity of $\mathbf{F}^{(\ell)}$, so $\mathcal{T}_{\ell,G}$ is compact. Hence Lemma B.4 applies row-wise on $\mathcal{T}_{\ell,G}$.

Using Lemma B.1 and L_σ-Lipschitzness of σ (recall (15)), we have for all $\mathbf{A}, \mathbf{A}'$,

$$\|\mathrm{FFN}^{(\ell)}(\mathbf{A}) - \mathrm{FFN}^{(\ell)}(\mathbf{A}')\|_{2,\infty} \leq \|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}} \, L_\sigma \, \|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}} \, \|\mathbf{A} - \mathbf{A}'\|_{2,\infty}. \tag{25}$$

Therefore, by Lemma B.4 and the residual addition,

$$\|\mathbf{G}^{(\ell)}(\mathbf{A}) - \mathbf{G}^{(\ell)}(\mathbf{A}')\|_{2,\infty} \leq L_{\mathrm{LN}}(\mathcal{T}_{\ell,G}) \left(1 + \|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}} \, L_\sigma \, \|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}}\right) \|\mathbf{A} - \mathbf{A}'\|_{2,\infty}. \tag{26}$$

Step 3: One layer is Lipschitz on $\mathcal{S}_{\ell-1}$. Let $\mathbf{A} = \mathbf{F}^{(\ell)}(\mathbf{Y})$ and $\mathbf{A}' = \mathbf{F}^{(\ell)}(\mathbf{Y}')$ in (26) and then apply (24). This yields, for all $\mathbf{Y}, \mathbf{Y}' \in \mathcal{S}_{\ell-1}$,

$$\|\mathbf{H}^{(\ell)}(\mathbf{Y}) - \mathbf{H}^{(\ell)}(\mathbf{Y}')\|_{2,\infty} \leq L_\ell(\mathcal{S}) \, \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty},$$

with the admissible choice

$$L_\ell(\mathcal{S}) \triangleq L_{\mathrm{LN}}(\mathcal{T}_{\ell,G}) \left(1 + \|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}} \, L_\sigma \, \|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}}\right) L_{\mathrm{LN}}(\mathcal{T}_{\ell,F}) \left(1 + L_{\mathrm{MHSA}}^{(\ell)}(\mathcal{S}_{\ell-1})\right).$$

Step 4: Propagate across layers. Fix $\mathbf{U}, \mathbf{U}' \in \mathcal{S}$ and let $\mathbf{Y}^{(0)} = \mathbf{U}$ and $\mathbf{Y}'^{(0)} = \mathbf{U}'$. Iterating the one-layer Lipschitz bound over $\ell = 1, \ldots, L_{\text{mix}}$ gives

$$\|\mathbf{Y}^{(L_{\text{mix}})} - \mathbf{Y}'^{(L_{\text{mix}})}\|_{2,\infty} \leq \left(\prod_{\ell=1}^{L_{\text{mix}}} L_\ell(\mathcal{S})\right) \|\mathbf{U} - \mathbf{U}'\|_{2,\infty}.$$

Since $\text{Mixer}_{L_{\text{mix}}}(\mathbf{U}) = \mathbf{Y}^{(L_{\text{mix}})}$ and $\text{Mixer}_{L_{\text{mix}}}(\mathbf{U}') = \mathbf{Y}'^{(L_{\text{mix}})}$, the theorem follows with $L_{\text{mix}}(\mathcal{S}) \triangleq \prod_{\ell=1}^{L_{\text{mix}}} L_\ell(\mathcal{S})$. □

B.1. From row-wise LayerNorm stability to a checkable layer constant

This subsection makes explicit how the row-wise LayerNorm stability constant $L_{\text{LN}}(\mathcal{T})$ (Lemma B.4) enters the per-layer encoder constant $L_\ell(\mathcal{S})$ used in Theorem B.5. Concretely:

- Lemma B.7 decomposes $L_\ell(\mathcal{S})$ into *two* LayerNorm constants evaluated on the relevant *pre-LN row-sets* and standard Lipschitz constants of the MHSA and FFN sublayers.
- Theorem B.8 shows how to upper bound each factor by quantities computed from a single forward pass on the segment endpoints (plus operator norms of weights). This yields a practical, conservative coefficient for the certificate, without sacrificing rigor.

Definition B.6 (Pre-LN row-sets induced by a segment)**.** Fix a segment $\mathcal{S} \subseteq \mathbb{R}^{M\times d}$. For encoder layer $\ell \in \{1, \ldots, L_{\text{mix}}\}$, let $\mathcal{S}_{\ell-1}$ denote the set of possible inputs to layer ℓ induced by $\mathcal{S}$ (i.e., the image of $\mathcal{S}$ under the first $\ell - 1$ layers of $\text{Mixer}_{L_{\text{mix}}}$, with $\mathcal{S}_0 \triangleq \mathcal{S}$).

Define the two pre-LN row-sets:

$$\mathcal{T}_{\ell,F} \triangleq \left\{\mathbf{u} \in \mathbb{R}^d : \exists\, \mathbf{Y} \in \mathcal{S}_{\ell-1},\ \exists\, i \in [M] \text{ s.t. } \mathbf{u} = \Big(\mathbf{Y} + \text{MHSA}^{(\ell)}(\mathbf{Y})\Big)_{i,:}\right\},$$

$$\mathcal{T}_{\ell,G} \triangleq \left\{\mathbf{u} \in \mathbb{R}^d : \exists\, \mathbf{A} \in \mathbf{F}^{(\ell)}(\mathcal{S}_{\ell-1}),\ \exists\, i \in [M] \text{ s.t. } \mathbf{u} = \Big(\mathbf{A} + \text{FFN}^{(\ell)}(\mathbf{A})\Big)_{i,:}\right\},$$

where $\mathbf{F}^{(\ell)}(\mathbf{Y}) = \text{LayerNorm}\Big(\mathbf{Y} + \text{MHSA}^{(\ell)}(\mathbf{Y})\Big)$ as in the proof of Theorem B.5. △

Lemma B.7 (Per-layer Lipschitz constant decomposes into LayerNorm and sublayer constants)**.** *Fix $\ell \in \{1, \ldots, L_{\text{mix}}\}$ and a segment $\mathcal{S} \subseteq \mathbb{R}^{M\times d}$. Recall σ is L_σ-Lipschitz and $\varepsilon_{\text{ln}} > 0$. (See also Section A.1 for details.) Suppose that both pre-LN row-sets $\mathcal{T}_{\ell,F}$ and $\mathcal{T}_{\ell,G}$ from Definition B.6 have positive stabilized variance lower bounds:*

$$m_{\ell,F} \triangleq \inf_{\mathbf{u}\in\mathcal{T}_{\ell,F}} \Big(\text{Var}(\mathbf{u}) + \varepsilon_{\text{ln}}\Big) > 0, \qquad m_{\ell,G} \triangleq \inf_{\mathbf{u}\in\mathcal{T}_{\ell,G}} \Big(\text{Var}(\mathbf{u}) + \varepsilon_{\text{ln}}\Big) > 0.$$

Then the ℓth post-LN encoder layer map $\text{Layer}^{(\ell)} : \mathbb{R}^{M\times d} \to \mathbb{R}^{M\times d}, \mathbf{Y}^{(\ell-1)} \mapsto \mathbf{Y}^{(\ell)}$, as defined in (17), is Lipschitz on $\mathcal{S}_{\ell-1}$ under $\|\cdot\|_{2,\infty}$ with constant

$$L_\ell(\mathcal{S}) \leq L_{\text{LN}}\Big(\mathcal{T}_{\ell,G}\Big) \cdot \Big(1 + L_{\text{FFN}}^{(\ell)}\Big) \cdot L_{\text{LN}}\Big(\mathcal{T}_{\ell,F}\Big) \cdot \Big(1 + L_{\text{MHSA}}^{(\ell)}(\mathcal{S}_{\ell-1})\Big),$$

where

$$L_{\text{FFN}}^{(\ell)} \triangleq \|\mathbf{W}_2^{(\ell)}\|_{\text{op}} \cdot L_\sigma \cdot \|\mathbf{W}_1^{(\ell)}\|_{\text{op}},$$

$$L_{\text{MHSA}}^{(\ell)}(\mathcal{S}_{\ell-1}) \triangleq \|\mathbf{W}_O^{(\ell)}\|_{\text{op}} \cdot \sum_{h=1}^{H} \left(\left(\Gamma_Q^{(\ell,h)} \cdot \Gamma_V^{(\ell,h)} \cdot \|\mathbf{W}_K^{(\ell,h)}\|_{\text{op}}\right) + \left(\Gamma_K^{(\ell,h)} \cdot \Gamma_V^{(\ell,h)} \cdot \|\mathbf{W}_Q^{(\ell,h)}\|_{\text{op}}\right) + \|\mathbf{W}_V^{(\ell,h)}\|_{\text{op}}\right),$$

and the head-wise row-norm envelopes are

$$\Gamma_Q^{(\ell,h)} \triangleq \frac{1}{\sqrt{d_h}} \cdot \sup_{\mathbf{Y}\in\mathcal{S}_{\ell-1}} \big\|\mathbf{Y} \cdot \mathbf{W}_Q^{(\ell,h)}\big\|_{2,\infty},$$

$$\Gamma_K^{(\ell,h)} \triangleq \sup_{\mathbf{Y}\in\mathcal{S}_{\ell-1}} \big\|\mathbf{Y} \cdot \mathbf{W}_K^{(\ell,h)}\big\|_{2,\infty},$$

$$\Gamma_V^{(\ell,h)} \triangleq \max\left\{\sup_{\mathbf{Y}\in\mathcal{S}_{\ell-1}} \big\|\mathbf{Y} \cdot \mathbf{W}_V^{(\ell,h)}\big\|_{2,\infty}, \sup_{\mathbf{Y}\in\mathcal{S}_{\ell-1}} \big\|\mathbf{Y}' \cdot \mathbf{W}_V^{(\ell,h)}\big\|_{2,\infty}\right\} = \sup_{\mathbf{Y}\in\mathcal{S}_{\ell-1}} \big\|\mathbf{Y} \cdot \mathbf{W}_V^{(\ell,h)}\big\|_{2,\infty}.$$

In particular, $L_\ell(\mathcal{S})$ is controlled by two LayerNorm constants evaluated on the pre-LN row-sets *and by row-norm envelopes and operator norms of weight matrices.*

Proof. Let $\mathbf{Y}, \mathbf{Y}' \in \mathcal{S}_{\ell-1}$. Define the residual-LN maps

$$\mathbf{F}^{(\ell)}(\mathbf{Y}) \triangleq \mathrm{LayerNorm}\Big(\mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y})\Big),$$
$$\mathbf{G}^{(\ell)}(\mathbf{A}) \triangleq \mathrm{LayerNorm}\Big(\mathbf{A} + \mathrm{FFN}^{(\ell)}(\mathbf{A})\Big),$$

so that the layer map is the composition $\mathrm{Layer}^{(\ell)} = \mathbf{G}^{(\ell)} \circ \mathbf{F}^{(\ell)}$.

Step 1: bound $\mathbf{F}^{(\ell)}$ on $\mathcal{S}_{\ell-1}$. By Definition B.6, for any $\mathbf{Y} \in \mathcal{S}_{\ell-1}$ each row of $\mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y})$ lies in $\mathcal{T}_{\ell,F}$. Applying Lemma B.4 row-wise yields

$$\begin{aligned}\big\|\mathbf{F}^{(\ell)}(\mathbf{Y}) - \mathbf{F}^{(\ell)}(\mathbf{Y}')\big\|_{2,\infty} &\le L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,F}\Big) \cdot \big\|\big(\mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y})\big) - \big(\mathbf{Y}' + \mathrm{MHSA}^{(\ell)}(\mathbf{Y}')\big)\big\|_{2,\infty} \\ &\le L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,F}\Big) \cdot \Big(\big\|\mathbf{Y} - \mathbf{Y}'\big\|_{2,\infty} + \big\|\mathrm{MHSA}^{(\ell)}(\mathbf{Y}) - \mathrm{MHSA}^{(\ell)}(\mathbf{Y}')\big\|_{2,\infty}\Big).\end{aligned}$$

Step 1a: bound the MHSA difference head-by-head. Fix a head h. Define projected queries / keys / values

$$\mathbf{Q}^{(h)}(\mathbf{Y}) \triangleq \mathbf{Y} \cdot \mathbf{W}_Q^{(\ell,h)}, \qquad \mathbf{K}^{(h)}(\mathbf{Y}) \triangleq \mathbf{Y} \cdot \mathbf{W}_K^{(\ell,h)}, \qquad \mathbf{V}^{(h)}(\mathbf{Y}) \triangleq \mathbf{Y} \cdot \mathbf{W}_V^{(\ell,h)}.$$

By Lemma B.1,

$$\begin{aligned}\big\|\mathbf{Q}^{(h)}(\mathbf{Y}) - \mathbf{Q}^{(h)}(\mathbf{Y}')\big\|_{2,\infty} &\le \|\mathbf{W}_Q^{(\ell,h)}\|_{\mathrm{op}} \cdot \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}, \\ \big\|\mathbf{K}^{(h)}(\mathbf{Y}) - \mathbf{K}^{(h)}(\mathbf{Y}')\big\|_{2,\infty} &\le \|\mathbf{W}_K^{(\ell,h)}\|_{\mathrm{op}} \cdot \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}, \\ \big\|\mathbf{V}^{(h)}(\mathbf{Y}) - \mathbf{V}^{(h)}(\mathbf{Y}')\big\|_{2,\infty} &\le \|\mathbf{W}_V^{(\ell,h)}\|_{\mathrm{op}} \cdot \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}.\end{aligned}$$

Moreover, by definition of $\Gamma_Q^{(\ell,h)}, \Gamma_K^{(\ell,h)}, \Gamma_V^{(\ell,h)}$ we have, uniformly over $\mathbf{Y}, \mathbf{Y}' \in \mathcal{S}_{\ell-1}$,

$$\frac{1}{\sqrt{d_h}} \cdot \big\|\mathbf{Q}^{(h)}(\mathbf{Y})\big\|_{2,\infty} \le \Gamma_Q^{(\ell,h)}, \qquad \big\|\mathbf{K}^{(h)}(\mathbf{Y})\big\|_{2,\infty} \le \Gamma_K^{(\ell,h)}, \qquad \max\Big\{\big\|\mathbf{V}^{(h)}(\mathbf{Y})\big\|_{2,\infty}, \big\|\mathbf{V}^{(h)}(\mathbf{Y}')\big\|_{2,\infty}\Big\} \le \Gamma_V^{(\ell,h)}.$$

Applying Lemma B.3 (with $N_q = N_k = M$ and $d_k = d_h$) gives

$$\begin{aligned}&\Big\|\mathrm{Atten}\big(\mathbf{Q}^{(h)}(\mathbf{Y}); \mathbf{K}^{(h)}(\mathbf{Y}), \mathbf{V}^{(h)}(\mathbf{Y})\big) - \mathrm{Atten}\big(\mathbf{Q}^{(h)}(\mathbf{Y}'); \mathbf{K}^{(h)}(\mathbf{Y}'), \mathbf{V}^{(h)}(\mathbf{Y}')\big)\Big\|_{2,\infty} \\ &\le \left(\Gamma_Q^{(\ell,h)} \cdot \Gamma_V^{(\ell,h)} \cdot \|\mathbf{W}_K^{(\ell,h)}\|_{\mathrm{op}} + \Gamma_K^{(\ell,h)} \cdot \Gamma_V^{(\ell,h)} \cdot \|\mathbf{W}_Q^{(\ell,h)}\|_{\mathrm{op}} + \|\mathbf{W}_V^{(\ell,h)}\|_{\mathrm{op}}\right) \cdot \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}.\end{aligned}$$

Step 1b: combine heads and output projection. Let $\mathbf{U}^{(h)}(\mathbf{Y}) \in \mathbb{R}^{M \times d_h}$ denote the output of head $h \in \{1, \dots, H\}$ before the final output projection. Define the concatenation operator

$$\mathrm{Concat}_h\, \mathbf{U}^{(h)}(\mathbf{Y}) \triangleq \big[\mathbf{U}^{(1)}(\mathbf{Y}) \,\big|\, \mathbf{U}^{(2)}(\mathbf{Y}) \,\big|\, \cdots \,\big|\, \mathbf{U}^{(H)}(\mathbf{Y})\big] = \bigoplus_h \mathbf{U}^{(h)}(\mathbf{Y}) \in \mathbb{R}^{M \times (H d_h)},$$

i.e., concatenation along the column (feature) dimension. Equivalently, for each row $i \in [M]$,

$$\big(\mathrm{Concat}_h\, \mathbf{U}^{(h)}(\mathbf{Y})\big)_{i,:} = \big[\mathbf{U}^{(1)}(\mathbf{Y})_{i,:},\ \mathbf{U}^{(2)}(\mathbf{Y})_{i,:},\ \dots,\ \mathbf{U}^{(H)}(\mathbf{Y})_{i,:}\big].$$

Difference of concatenations. For $\mathbf{Y}, \mathbf{Y}'$, define

$$\mathrm{Concat}_h\, \mathbf{U}^{(h)}(\mathbf{Y}) - \mathrm{Concat}_h\, \mathbf{U}^{(h)}(\mathbf{Y}') \triangleq \big[\mathbf{U}^{(1)}(\mathbf{Y}) - \mathbf{U}^{(1)}(\mathbf{Y}') \,\big|\, \cdots \,\big|\, \mathbf{U}^{(H)}(\mathbf{Y}) - \mathbf{U}^{(H)}(\mathbf{Y}')\big] \in \mathbb{R}^{M\times(Hd_h)}.$$

The multi-head concatenation satisfies, for each row i,

$$\big\|[\mathbf{U}^{(1)}_{i,:}, \ldots, \mathbf{U}^{(H)}_{i,:}]\big\|_2 = \sqrt{\sum_{h=1}^{H} \|\mathbf{U}^{(h)}_{i,:}\|_2^2} \;\le\; \sum_{h=1}^{H} \|\mathbf{U}^{(h)}_{i,:}\|_2,$$

hence $\big\|\mathrm{Concat}_h\, \mathbf{U}^{(h)}(\mathbf{Y}) - \mathrm{Concat}_h\, \mathbf{U}^{(h)}(\mathbf{Y}')\big\|_{2,\infty} \le \sum_{h=1}^{H} \|\mathbf{U}^{(h)}(\mathbf{Y}) - \mathbf{U}^{(h)}(\mathbf{Y}')\|_{2,\infty}$. Finally, applying the output projection and Lemma B.1 gives

$$\big\|\mathrm{MHSA}^{(\ell)}(\mathbf{Y}) - \mathrm{MHSA}^{(\ell)}(\mathbf{Y}')\big\|_{2,\infty} \;\le\; L^{(\ell)}_{\mathrm{MHSA}}(\mathcal{S}_{\ell-1}) \cdot \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}.$$

Substituting into the bound for $\mathbf{F}^{(\ell)}$ yields

$$\big\|\mathbf{F}^{(\ell)}(\mathbf{Y}) - \mathbf{F}^{(\ell)}(\mathbf{Y}')\big\|_{2,\infty} \;\le\; L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,F}\Big) \cdot \Big(1 + L^{(\ell)}_{\mathrm{MHSA}}(\mathcal{S}_{\ell-1})\Big) \cdot \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}.$$

Step 2: bound $\mathbf{G}^{(\ell)}$ on $\mathbf{F}^{(\ell)}(\mathcal{S}_{\ell-1})$. Let $\mathbf{A}, \mathbf{A}' \in \mathbf{F}^{(\ell)}(\mathcal{S}_{\ell-1})$. By Definition B.6, the rows of $\mathbf{A} + \mathrm{FFN}^{(\ell)}(\mathbf{A})$ lie in $\mathcal{T}_{\ell,G}$. Applying Lemma B.4 row-wise gives

$$\big\|\mathbf{G}^{(\ell)}(\mathbf{A}) - \mathbf{G}^{(\ell)}(\mathbf{A}')\big\|_{2,\infty} \le L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,G}\Big) \cdot \Big(\big\|\mathbf{A} - \mathbf{A}'\big\|_{2,\infty} + \big\|\mathrm{FFN}^{(\ell)}(\mathbf{A}) - \mathrm{FFN}^{(\ell)}(\mathbf{A}')\big\|_{2,\infty}\Big).$$

For the FFN, apply Lemma B.1 twice and Lipschitzness of σ:

$$\begin{aligned}
\big\|\mathrm{FFN}^{(\ell)}(\mathbf{A}) - \mathrm{FFN}^{(\ell)}(\mathbf{A}')\big\|_{2,\infty} &= \big\|\sigma\big(\mathbf{A}\cdot\mathbf{W}_1^{(\ell)} + \mathbf{b}_1^{(\ell)}\big)\cdot\mathbf{W}_2^{(\ell)} - \sigma\big(\mathbf{A}'\cdot\mathbf{W}_1^{(\ell)} + \mathbf{b}_1^{(\ell)}\big)\cdot\mathbf{W}_2^{(\ell)}\big\|_{2,\infty} \\
&\le \|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}} \cdot L_\sigma \cdot \|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}} \cdot \|\mathbf{A} - \mathbf{A}'\|_{2,\infty} \\
&= L^{(\ell)}_{\mathrm{FFN}} \cdot \|\mathbf{A} - \mathbf{A}'\|_{2,\infty}.
\end{aligned}$$

Thus

$$\big\|\mathbf{G}^{(\ell)}(\mathbf{A}) - \mathbf{G}^{(\ell)}(\mathbf{A}')\big\|_{2,\infty} \;\le\; L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,G}\Big) \cdot \Big(1 + L^{(\ell)}_{\mathrm{FFN}}\Big) \cdot \|\mathbf{A} - \mathbf{A}'\|_{2,\infty}.$$

Step 3: compose. Set $\mathbf{A} = \mathbf{F}^{(\ell)}(\mathbf{Y})$ and $\mathbf{A}' = \mathbf{F}^{(\ell)}(\mathbf{Y}')$ and combine Step 1 and Step 2:

$$\big\|\mathrm{Layer}^{(\ell)}(\mathbf{Y}) - \mathrm{Layer}^{(\ell)}(\mathbf{Y}')\big\|_{2,\infty} \;\le\; L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,G}\Big) \cdot \Big(1 + L^{(\ell)}_{\mathrm{FFN}}\Big) \cdot L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,F}\Big) \cdot \Big(1 + L^{(\ell)}_{\mathrm{MHSA}}(\mathcal{S}_{\ell-1})\Big) \cdot \|\mathbf{Y} - \mathbf{Y}'\|_{2,\infty}.$$

Any per-layer Lipschitz constant $L_\ell(\mathcal{S})$ may be chosen as the right-hand side coefficient, proving the claim. □

Theorem B.8 (Forward-pass checkability of a practical upper bound for $L_\ell(\mathcal{S})$)**.** *Fix $\ell \in \{1, \ldots, L_{\mathrm{mix}}\}$ and a segment $\mathcal{S} \subseteq \mathbb{R}^{M\times d}$. Assume inference mode (dropout disabled). Recall that σ is L_σ-Lipschitz. (See also Section A.1 for details.) Let $L_\ell(\mathcal{S})$ be any Lipschitz constant of the ℓth encoder layer on $\mathcal{S}_{\ell-1}$. Then the quantity*

$$\overline{L}_\ell(\mathcal{S}) \triangleq L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,G}\Big) \cdot \Big(1 + L^{(\ell)}_{\mathrm{FFN}}\Big) \cdot L_{\mathrm{LN}}\Big(\mathcal{T}_{\ell,F}\Big) \cdot \Big(1 + L^{(\ell)}_{\mathrm{MHSA}}(\mathcal{S}_{\ell-1})\Big)$$

satisfies $L_\ell(\mathcal{S}) \le \overline{L}_\ell(\mathcal{S})$ and is forward-pass computable *in the following sense.*

Assume we can evaluate the layer at the two endpoints of $\mathcal{S}_{\ell-1}$, i.e., $\mathbf{Y}^{\mathrm{det}}_{\ell-1}$ and $\mathbf{Y}_{\ell-1}$, and we can compute operator norms of weight matrices (e.g., by power iteration). Then each factor in $\overline{L}_\ell(\mathcal{S})$ admits an explicit conservative *upper bound computed from these endpoint activations:*

1. *$\Gamma_Q^{(\ell,h)}, \Gamma_K^{(\ell,h)}, \Gamma_V^{(\ell,h)}$ can be upper bounded by evaluating $\big\|\mathbf{Y}^{\mathrm{det}}_{\ell-1}\cdot\mathbf{W}^{(\ell,h)}_{\{Q,K,V\}}\big\|_{2,\infty}$ and $\big\|\mathbf{Y}_{\ell-1}\cdot\mathbf{W}^{(\ell,h)}_{\{Q,K,V\}}\big\|_{2,\infty}$ and taking the maximum;*

2. $L_{\mathrm{FFN}}^{(\ell)}$ *depends only on* $\|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}}$, $\|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}}$, *and* L_σ;
3. $L_{\mathrm{LN}}(\mathcal{T}_{\ell,F})$ *and* $L_{\mathrm{LN}}(\mathcal{T}_{\ell,G})$ *admit endpoint-based upper bounds by (i) bounding the possible deviation of pre-LN rows away from the endpoint rows using sublayer Lipschitz constants and (ii) translating this deviation into conservative bounds on the variance and centered norm terms that control* $L_{\mathrm{LN}}(\cdot)$ *(as described in the proof).*

Hence $\overline{L}_\ell(\mathcal{S})$ *is a practical, checkable coefficient that upper bounds* $L_\ell(\mathcal{S})$.

Proof. We split the argument into two parts: (i) why $L_\ell(\mathcal{S}) \le \overline{L}_\ell(\mathcal{S})$ holds, and (ii) why each factor in $\overline{L}_\ell(\mathcal{S})$ can be bounded from endpoint information.

Part I: why $L_\ell(\mathcal{S}) \le \overline{L}_\ell(\mathcal{S})$. The inequality is the direct product-form bound from Lemma B.7: each residual sublayer contributes a $(1 + \mathrm{Lip})$ factor and each LayerNorm contributes a $L_{\mathrm{LN}}(\cdot)$ factor. We therefore focus on checkability of the four factors in $\overline{L}_\ell(\mathcal{S})$.

Part II: forward-pass checkability of each factor.

Step 1: checkability of the head envelopes. Fix any matrix $\mathbf{W}$. Define $f(\mathbf{Y}) \triangleq \|\mathbf{Y}\mathbf{W}\|_{2,\infty}$. We claim that

$$\sup_{\mathbf{Y}\in\mathcal{S}_{\ell-1}} \|\mathbf{Y}\mathbf{W}\|_{2,\infty} \;=\; \max\Big\{\|\mathbf{Y}_{\ell-1}^{\mathrm{det}}\mathbf{W}\|_{2,\infty},\; \|\mathbf{Y}_{\ell-1}\mathbf{W}\|_{2,\infty}\Big\}. \tag{27}$$

Proof of (27). First, f is convex: $\mathbf{Y} \mapsto \mathbf{Y}\mathbf{W}$ is linear and $\|\cdot\|_{2,\infty}$ is a norm, hence their composition is convex. Next, parametrize the segment as

$$\mathbf{Y}(t) \triangleq \mathbf{Y}_{\ell-1}^{\mathrm{det}} + t(\mathbf{Y}_{\ell-1} - \mathbf{Y}_{\ell-1}^{\mathrm{det}}), \qquad t \in [0,1].$$

Then $g(t) \triangleq f(\mathbf{Y}(t))$ is convex in t as the composition of a convex function with an affine map. A convex function on a compact interval attains its maximum at an endpoint, so $\sup_{t\in[0,1]} g(t) = \max\{g(0), g(1)\}$, which is exactly (27). □

Applying (27) with $\mathbf{W} \in \{\mathbf{W}_Q^{(\ell,h)}, \mathbf{W}_K^{(\ell,h)}, \mathbf{W}_V^{(\ell,h)}\}$ yields endpoint-computable upper bounds for $\Gamma_Q^{(\ell,h)}, \Gamma_K^{(\ell,h)}, \Gamma_V^{(\ell,h)}$.

Step 2: checkability of $L_{\mathrm{FFN}}^{(\ell)}$. By definition,

$$L_{\mathrm{FFN}}^{(\ell)} = \|\mathbf{W}_2^{(\ell)}\|_{\mathrm{op}} \cdot L_\sigma \cdot \|\mathbf{W}_1^{(\ell)}\|_{\mathrm{op}}.$$

Thus $L_{\mathrm{FFN}}^{(\ell)}$ depends only on operator norms of the FFN weight matrices and the known constant L_σ. Since $\|\cdot\|_{\mathrm{op}}$ is the spectral norm, it can be upper bounded in practice by standard estimators such as power iteration (or exact SVD for small matrices). This makes $L_{\mathrm{FFN}}^{(\ell)}$ forward-pass checkable.

Step 3: endpoint bounds for LayerNorm constants on induced row-sets. Lemma B.4 bounds $L_{\mathrm{LN}}(\mathcal{T})$ in terms of two scalars that depend only on the pre-LN row-set $\mathcal{T} \subseteq \mathbb{R}^d$:

$$m_{\mathcal{T}} = \inf_{\mathbf{u}\in\mathcal{T}} \Big(\mathrm{Var}(\mathbf{u}) + \varepsilon_{\mathrm{ln}}\Big), \qquad C_{\mathcal{T}} = \sup_{\mathbf{u}\in\mathcal{T}} \big\|\mathbf{u} - \mu(\mathbf{u})\mathbf{1}\big\|_2.$$

Therefore, to upper bound $L_{\mathrm{LN}}(\mathcal{T})$, it suffices to compute: (a) a *lower* bound on $m_{\mathcal{T}}$ and (b) an *upper* bound on $C_{\mathcal{T}}$. We now show how to do this from endpoint activations, separately for $\mathcal{T}_{\ell,F}$ and $\mathcal{T}_{\ell,G}$.

Two row-wise Lipschitz facts. For any $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$, the mean map $\mu(\cdot)$ is $1/\sqrt{d}$-Lipschitz in ℓ_2:

$$\big|\mu(\mathbf{u}) - \mu(\mathbf{v})\big| = \frac{1}{d}\big|\mathbf{1}^\top(\mathbf{u} - \mathbf{v})\big| \le \frac{1}{d}\|\mathbf{1}\|_2 \cdot \|\mathbf{u} - \mathbf{v}\|_2 = \frac{1}{\sqrt{d}}\|\mathbf{u} - \mathbf{v}\|_2. \tag{28}$$

Moreover, the centering map $\mathbf{u} \mapsto \mathbf{u} - \mu(\mathbf{u})\mathbf{1}$ is 2-Lipschitz:

$$\big\|\mathbf{u} - \mu(\mathbf{u})\mathbf{1} - \big(\mathbf{v} - \mu(\mathbf{v})\mathbf{1}\big)\big\|_2 \le \|\mathbf{u} - \mathbf{v}\|_2 + \sqrt{d}\big|\mu(\mathbf{u}) - \mu(\mathbf{v})\big| \le 2\|\mathbf{u} - \mathbf{v}\|_2, \tag{29}$$

where the second inequality uses (28).

Step 3a: bound $L_{\mathrm{LN}}(\mathcal{T}_{\ell,F})$ from endpoints. Define the pre-LN map $\mathbf{H}_{\ell,F}(\mathbf{Y}) \triangleq \mathbf{Y} + \mathrm{MHSA}^{(\ell)}(\mathbf{Y})$. By Lemma B.7, $\mathrm{MHSA}^{(\ell)}$ is Lipschitz on $\mathcal{S}_{\ell-1}$ with constant $L^{(\ell)}_{\mathrm{MHSA}}(\mathcal{S}_{\ell-1})$. Hence $\mathbf{H}_{\ell,F}$ is Lipschitz on $\mathcal{S}_{\ell-1}$ with

$$L^{\mathrm{pre}}_{\ell,F} \triangleq 1 + L^{(\ell)}_{\mathrm{MHSA}}(\mathcal{S}_{\ell-1}). \tag{30}$$

Let the segment radius be $r_{\ell-1} \triangleq \|\mathbf{Y}_{\ell-1} - \mathbf{Y}^{\mathrm{det}}_{\ell-1}\|_{2,\infty}$. Then, for any $\mathbf{Y} \in \mathcal{S}_{\ell-1}$,

$$\big\|\mathbf{H}_{\ell,F}(\mathbf{Y}) - \mathbf{H}_{\ell,F}(\mathbf{Y}^{\mathrm{det}}_{\ell-1})\big\|_{2,\infty} \le L^{\mathrm{pre}}_{\ell,F} \cdot \|\mathbf{Y} - \mathbf{Y}^{\mathrm{det}}_{\ell-1}\|_{2,\infty} \le L^{\mathrm{pre}}_{\ell,F} \cdot r_{\ell-1}. \tag{31}$$

Set $\Delta_{\ell,F} \triangleq L^{\mathrm{pre}}_{\ell,F} \cdot r_{\ell-1}$. Fix any $\mathbf{u} \in \mathcal{T}_{\ell,F}$. By definition of $\mathcal{T}_{\ell,F}$, there exist $\mathbf{Y} \in \mathcal{S}_{\ell-1}$ and $i \in [M]$ such that $\mathbf{u} = \mathbf{H}_{\ell,F}(\mathbf{Y})_{i,:}$. Let $\mathbf{u}_0 \triangleq \mathbf{H}_{\ell,F}(\mathbf{Y}^{\mathrm{det}}_{\ell-1})_{i,:}$. Then (31) implies

$$\|\mathbf{u} - \mathbf{u}_0\|_2 \le \|\mathbf{u} - \mathbf{u}_0\|_{2,\infty} \le \Delta_{\ell,F}. \tag{32}$$

Now use (29) with $(\mathbf{u}, \mathbf{u}_0)$:

$$\Big|\, \|\mathbf{u} - \mu(\mathbf{u})\mathbf{1}\|_2 - \|\mathbf{u}_0 - \mu(\mathbf{u}_0)\mathbf{1}\|_2 \,\Big| \le \big\|\mathbf{u} - \mu(\mathbf{u})\mathbf{1} - \big(\mathbf{u}_0 - \mu(\mathbf{u}_0)\mathbf{1}\big)\big\|_2 \le 2\|\mathbf{u} - \mathbf{u}_0\|_2. \tag{33}$$

Combining (32) and (33) yields

$$\|\mathbf{u} - \mu(\mathbf{u})\mathbf{1}\|_2 \ge \|\mathbf{u}_0 - \mu(\mathbf{u}_0)\mathbf{1}\|_2 - 2\Delta_{\ell,F}, \qquad \|\mathbf{u} - \mu(\mathbf{u})\mathbf{1}\|_2 \le \|\mathbf{u}_0 - \mu(\mathbf{u}_0)\mathbf{1}\|_2 + 2\Delta_{\ell,F}. \tag{34}$$

Define endpoint row statistics

$$c^{\min}_{\ell,F} \triangleq \min\left\{ \min_{i\in[M]} \big\|\mathbf{H}_{\ell,F}(\mathbf{Y}^{\mathrm{det}}_{\ell-1})_{i,:} - \mu(\mathbf{H}_{\ell,F}(\mathbf{Y}^{\mathrm{det}}_{\ell-1})_{i,:})\mathbf{1}\big\|_2,\ \min_{i\in[M]} \big\|\mathbf{H}_{\ell,F}(\mathbf{Y}_{\ell-1})_{i,:} - \mu(\mathbf{H}_{\ell,F}(\mathbf{Y}_{\ell-1})_{i,:})\mathbf{1}\big\|_2 \right\},$$

$$c^{\max}_{\ell,F} \triangleq \max\left\{ \max_{i\in[M]} \big\|\mathbf{H}_{\ell,F}(\mathbf{Y}^{\mathrm{det}}_{\ell-1})_{i,:} - \mu(\mathbf{H}_{\ell,F}(\mathbf{Y}^{\mathrm{det}}_{\ell-1})_{i,:})\mathbf{1}\big\|_2,\ \max_{i\in[M]} \big\|\mathbf{H}_{\ell,F}(\mathbf{Y}_{\ell-1})_{i,:} - \mu(\mathbf{H}_{\ell,F}(\mathbf{Y}_{\ell-1})_{i,:})\mathbf{1}\big\|_2 \right\}.$$

Since $\mathbf{u}_0$ is a row from the endpoint $\mathbf{H}_{\ell,F}(\mathbf{Y}^{\mathrm{det}}_{\ell-1})$, we have $\|\mathbf{u}_0 - \mu(\mathbf{u}_0)\mathbf{1}\|_2 \in [c^{\min}_{\ell,F}, c^{\max}_{\ell,F}]$, and thus (34) implies

$$\|\mathbf{u} - \mu(\mathbf{u})\mathbf{1}\|_2 \ge c^{\min}_{\ell,F} - 2\Delta_{\ell,F}, \qquad \|\mathbf{u} - \mu(\mathbf{u})\mathbf{1}\|_2 \le c^{\max}_{\ell,F} + 2\Delta_{\ell,F}.$$

Finally, using $\mathrm{Var}(\mathbf{u}) = \frac{1}{d}\|\mathbf{u} - \mu(\mathbf{u})\mathbf{1}\|_2^2$, we obtain conservative bounds

$$m^{\mathrm{check}}_{\ell,F} \triangleq \frac{1}{d}\Big(\max\{0,\, c^{\min}_{\ell,F} - 2\Delta_{\ell,F}\}\Big)^2 + \varepsilon_{\mathrm{ln}}, \qquad C^{\mathrm{check}}_{\ell,F} \triangleq c^{\max}_{\ell,F} + 2\Delta_{\ell,F}. \tag{35}$$

By construction, $m_{\mathcal{T}_{\ell,F}} \ge m^{\mathrm{check}}_{\ell,F}$ and $C_{\mathcal{T}_{\ell,F}} \le C^{\mathrm{check}}_{\ell,F}$. Plugging these into Lemma B.4 yields an endpoint-computable upper bound on $L_{\mathrm{LN}}(\mathcal{T}_{\ell,F})$.

Step 3b: bound $L_{\mathrm{LN}}(\mathcal{T}_{\ell,G})$ from endpoints. First compute the post-MHSA / pre-FFN activations at endpoints:

$$\mathbf{A}^{\mathrm{det}}_{\ell} \triangleq \mathbf{F}^{(\ell)}(\mathbf{Y}^{\mathrm{det}}_{\ell-1}), \qquad \mathbf{A}_{\ell} \triangleq \mathbf{F}^{(\ell)}(\mathbf{Y}_{\ell-1}).$$

Define the pre-LN FFN map $\mathbf{H}_{\ell,G}(\mathbf{A}) \triangleq \mathbf{A} + \mathrm{FFN}^{(\ell)}(\mathbf{A})$. By Lemma B.7, $\mathrm{FFN}^{(\ell)}$ is Lipschitz with constant $L^{(\ell)}_{\mathrm{FFN}}$. Hence $\mathbf{H}_{\ell,G}$ is Lipschitz with

$$L^{\mathrm{pre}}_{\ell,G} \triangleq 1 + L^{(\ell)}_{\mathrm{FFN}}. \tag{36}$$

Set $r_{\ell,A} \triangleq \|\mathbf{A}_{\ell} - \mathbf{A}^{\mathrm{det}}_{\ell}\|_{2,\infty}$ and $\Delta_{\ell,G} \triangleq L^{\mathrm{pre}}_{\ell,G} \cdot r_{\ell,A}$. Repeating the same argument as in Step 3a (now with $\mathbf{H}_{\ell,G}$ and endpoint matrices $\mathbf{A}^{\mathrm{det}}_{\ell}, \mathbf{A}_{\ell}$) yields conservative bounds

$$m^{\mathrm{check}}_{\ell,G} \triangleq \frac{1}{d}\Big(\max\{0,\, c^{\min}_{\ell,G} - 2\Delta_{\ell,G}\}\Big)^2 + \varepsilon_{\mathrm{ln}}, \qquad C^{\mathrm{check}}_{\ell,G} \triangleq c^{\max}_{\ell,G} + 2\Delta_{\ell,G}, \tag{37}$$

where

$$c_{\ell,G}^{\min} \triangleq \min\left\{ \min_{i\in[M]} \left\|\mathbf{H}_{\ell,G}(\mathbf{A}_\ell^{\det})_{i,:} - \mu(\mathbf{H}_{\ell,G}(\mathbf{A}_\ell^{\det})_{i,:})\mathbf{1}\right\|_2,\ \min_{i\in[M]} \left\|\mathbf{H}_{\ell,G}(\mathbf{A}_\ell)_{i,:} - \mu(\mathbf{H}_{\ell,G}(\mathbf{A}_\ell)_{i,:})\mathbf{1}\right\|_2 \right\},$$

$$c_{\ell,G}^{\max} \triangleq \max\left\{ \max_{i\in[M]} \left\|\mathbf{H}_{\ell,G}(\mathbf{A}_\ell^{\det})_{i,:} - \mu(\mathbf{H}_{\ell,G}(\mathbf{A}_\ell^{\det})_{i,:})\mathbf{1}\right\|_2,\ \max_{i\in[M]} \left\|\mathbf{H}_{\ell,G}(\mathbf{A}_\ell)_{i,:} - \mu(\mathbf{H}_{\ell,G}(\mathbf{A}_\ell)_{i,:})\mathbf{1}\right\|_2 \right\}.$$

By construction, $m_{\mathcal{T}_{\ell,G}} \geq m_{\ell,G}^{\text{check}}$ and $C_{\mathcal{T}_{\ell,G}} \leq C_{\ell,G}^{\text{check}}$. Plugging (37) into Lemma B.4 yields an endpoint-computable upper bound on $L_{\text{LN}}(\mathcal{T}_{\ell,G})$.

Step 4: conclude checkability of $\overline{L}_\ell(\mathcal{S})$. Step 1 provides endpoint-computable upper bounds for the MHSA head envelopes and hence $L_{\text{MHSA}}^{(\ell)}(\mathcal{S}_{\ell-1})$. Step 2 provides a checkable upper bound for $L_{\text{FFN}}^{(\ell)}$. Step 3 provides endpoint-computable upper bounds for $L_{\text{LN}}(\mathcal{T}_{\ell,F})$ and $L_{\text{LN}}(\mathcal{T}_{\ell,G})$. Therefore, each factor in $\overline{L}_\ell(\mathcal{S})$ can be bounded using endpoint activations and weight-matrix operator norms, so $\overline{L}_\ell(\mathcal{S})$ is forward-pass checkable. □

Remark B.9 (Interpretation: $L_{\text{LN}}(\mathcal{T})$ vs. $L_\ell(\mathcal{S})$). The constant $L_{\text{LN}}(\mathcal{T})$ quantifies how sensitive the *row-wise normalization operator* is on a specified pre-LN row-set $\mathcal{T} \subseteq \mathbb{R}^d$. In contrast, $L_\ell(\mathcal{S})$ quantifies how sensitive the *entire encoder layer* is on an input segment $\mathcal{S}_{\ell-1} \subseteq \mathbb{R}^{M\times d}$. Lemma B.7 makes this relationship explicit: two LayerNorm applications contribute the two factors $L_{\text{LN}}(\mathcal{T}_{\ell,F})$ and $L_{\text{LN}}(\mathcal{T}_{\ell,G})$, while the remaining factors account for the residual branches $\text{MHSA}^{(\ell)}$ and $\text{FFN}^{(\ell)}$. Theorem B.8 shows that $L_\ell(\mathcal{S})$ admits an explicit, conservative, forward-pass computable upper bound $\overline{L}_\ell(\mathcal{S})$; hence it can be used as a practical certificate coefficient. △

C. TensorSketch compressed core: definitions and kernel-estimation facts used

TensorSketch restores expressivity after KV compression with a standard and controllable mechanism. Stage I compresses length-N_k keys / values into $M \ll N_k$ summaries. This is efficient, but if we process the M summaries only linearly, the block can behave like a low-rank bottleneck. We therefore enrich each compressed item with higher-order interactions.

The clean target is the degree-k polynomial feature map: for $\mathbf{g} \in \mathbb{R}^{d'}$, the exact lift $\text{vec}(\mathbf{g}^{\otimes k}) \in \mathbb{R}^{(d')^k}$ induces the kernel $\langle \mathbf{g}, \mathbf{g}' \rangle^k$, but explicit expansion is infeasible for moderate d' and k. TensorSketch avoids this blow-up: it produces a D_k-dimensional sketch whose inner products provide an unbiased estimator of $\langle \mathbf{g}, \mathbf{g}' \rangle^k$, with distortion improving as D_k grows (Charikar et al., 2002; Pham & Pagh, 2013). This gives an explicit accuracy–compute knob $\{D_k\}_{k\in\mathcal{K}}$ that is separate from the sequence-length knob M: M controls how many compressed summaries are exposed to attention, while $\{D_k\}$ controls how accurately each summary is enriched.

We also keep all algorithmic randomness inside TensorSketch. This localization makes the approximation easy to analyze and tune: D_k controls concentration, and the preceding norm control stabilizes sketch variance (which scales with $\|\mathbf{g}\|_2^{2k}$ for degree k) and makes degree mixtures reliable in practice (Pham & Pagh, 2013).

All definitions below are stated for generic vectors in $\mathbb{R}^{d'}$. In PLASH, these vectors are the rows $\widetilde{\mathbf{G}}_{j,:}$ produced by the stabilized normalization in (7). The only assumption carried forward is that $\|\widetilde{\mathbf{G}}_{j,:}\|_2$ is uniformly controlled.

Definition C.1 (CountSketch and TensorSketch (explicit maps))**.** Fix integers $d' \geq 1$ and $D \geq 1$, and write $[D] \triangleq \{0, 1, \ldots, D-1\}$ and $[d'] \triangleq \{1, 2, \ldots, d'\}$. Let $h : [d'] \to [D]$ and $s : [d'] \to \{\pm 1\}$ be random hash functions. As in (Charikar et al., 2002; Pham & Pagh, 2013), we assume h is 2-wise independent and uniform on $[D]$, and s is 2-wise independent and uniform on $\{\pm 1\}$.

The CountSketch matrix $\mathbf{C} \in \mathbb{R}^{D\times d'}$ is defined entrywise by

$$\mathbf{C}_{b,i} \triangleq s(i)\,\mathbf{1}\{h(i) = b\}, \qquad b \in [D],\ i \in [d']. \tag{38}$$

For $\mathbf{g} \in \mathbb{R}^{d'}$, its CountSketch is $\mathbf{c}(\mathbf{g}) \triangleq \mathbf{C}\mathbf{g} \in \mathbb{R}^D$, i.e.,

$$\mathbf{c}(\mathbf{g})_b = \sum_{i:\, h(i)=b} s(i)\, g_i, \qquad b \in [D]. \tag{39}$$

Fix $k \geq 1$ and $D_k \geq 1$. For each $t \in \{1, \ldots, k\}$, sample an independent pair $h_t : [d'] \to [D_k]$ and $s_t : [d'] \to \{\pm 1\}$ with the same assumptions, and let $\mathbf{C}_t$ be the associated CountSketch matrix. Define $\mathbf{c}_t(\mathbf{g}) \triangleq \mathbf{C}_t \mathbf{g} \in \mathbb{R}^{D_k}$. TensorSketch is the k-fold circular convolution of these sketches (Pham & Pagh, 2013):

$$\mathrm{TS}_k(\mathbf{g}; D_k) \triangleq \mathbf{c}_1(\mathbf{g}) * \mathbf{c}_2(\mathbf{g}) * \cdots * \mathbf{c}_k(\mathbf{g}) \in \mathbb{R}^{D_k}, \tag{40}$$

$$\mathrm{TensorSketch}_k(\mathbf{g}; D_k, \{(h_t, s_t)\}_{t=1}^k) \triangleq \mathrm{TS}_k(\mathbf{g}; D_k) \in \mathbb{R}^{D_k}, \tag{41}$$

where $*$ denotes circular convolution on $[D_k]$. Using the FFT convolution identity (Cooley & Tukey, 1965),

$$\mathrm{TS}_k(\mathbf{g}; D_k) = \mathrm{IFFT}\Big(\mathrm{FFT}(\mathbf{c}_1(\mathbf{g})) \odot \cdots \odot \mathrm{FFT}(\mathbf{c}_k(\mathbf{g}))\Big). \tag{42}$$

TensorSketch can also be viewed as a CountSketch applied to the degree-k tensor lifting. Define the combined hash / sign on $[d']^k$ by

$$H(i_1, \ldots, i_k) \triangleq \Big(\sum_{t=1}^k h_t(i_t)\Big) \bmod D_k, \tag{43}$$

$$S(i_1, \ldots, i_k) \triangleq \prod_{t=1}^k s_t(i_t). \tag{44}$$

Let $\mathbf{C}^{(\otimes k)} \in \mathbb{R}^{D_k \times (d')^k}$ be the corresponding CountSketch matrix on tensor indices,

$$\mathbf{C}^{(\otimes k)}_{b,(i_1,\ldots,i_k)} \triangleq S(i_1, \ldots, i_k)\, \mathbf{1}\{H(i_1, \ldots, i_k) = b\}. \tag{45}$$

Then one obtains the identity (proved in (Pham & Pagh, 2013) and included here as a direct consequence of the convolution construction):

$$\mathrm{TS}_k(\mathbf{g}; D_k) = \mathbf{C}^{(\otimes k)}\, \mathrm{vec}(\mathbf{g}^{\otimes k}). \tag{46}$$

△

TensorSketch is valuable here because it preserves the lifted-feature viewpoint: the implemented enriched feature is a randomized linear projection of $\mathrm{vec}(\mathbf{g}^{\otimes k})$. This is precisely what lets us compare the implemented operator to an explicit deterministic reference using standard sketching theory.

Assumption C.2 (TensorSketch randomness with limited independence)**.** Fix a finite degree set $\mathcal{K} \subset \mathbb{Z}_{\geq 1}$. For each $k \in \mathcal{K}$ and each factor index $t \in [k]$, TensorSketch uses hash / sign maps

$$h_t^{(k)} : [d'] \to [D_k], \qquad s_t^{(k)} : [d'] \to \{\pm 1\},$$

such that:

1. $h_t^{(k)}$ is uniform on $[D_k]$ and 3-wise independent;
2. $s_t^{(k)}$ is uniform on $\{\pm 1\}$ and 4-wise independent;
3. the family $\{(h_t^{(k)}, s_t^{(k)})\}_{k \in \mathcal{K},\, t \in [k]}$ is mutually independent across all pairs (k, t).

This limited-independence regime is sufficient for (i) the second-moment calculations used in Proposition C.4 and (ii) polynomial-kernel OSE / AMP guarantees for learned readouts (Pham & Pagh, 2013; Avron et al., 2014). △

Construction C.3 (A checkable instantiation of Assumption C.2)**.** Fix $k \in \mathcal{K}$ and a target sketch length D_k.

Step 1 (3-wise uniform hashing). Choose a prime $p_k \geq \max\{d', D_k\}$, identify $[d'] \subset \mathbb{F}_{p_k}$, sample $a_0, a_1, a_2 \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(\mathbb{F}_{p_k})$, and define

$$\tilde{h}(x) = a_0 + a_1 x + a_2 x^2 \pmod{p_k}, \qquad x \in \mathbb{F}_{p_k}.$$

If $D_k = p_k$, set $h(i) \triangleq \tilde{h}(i) \in [D_k]$. If $D_k < p_k$, obtain an *exactly uniform* hash into $[D_k]$ by rejection sampling: output $\tilde{h}(i) \bmod D_k$ only when $\tilde{h}(i) < D_k \lfloor p_k / D_k \rfloor$, otherwise resample (a_0, a_1, a_2). The resulting h is uniform and 3-wise independent on $[D_k]$.

Step 2 (4-wise independent Rademacher signs). Work over $\mathbb{F}_{2^w}$ with $2^w \geq d'$, identify $[d'] \subset \mathbb{F}_{2^w}$, sample $b_0, b_1, b_2, b_3 \overset{\text{i.i.d.}}{\sim} \mathrm{Unif}(\mathbb{F}_{2^w})$, define

$$\tilde{s}(x) = b_0 + b_1 x + b_2 x^2 + b_3 x^3 \in \mathbb{F}_{2^w},$$

fix any nonzero $\mathbb{F}_2$-linear functional $\ell : \mathbb{F}_{2^w} \to \mathbb{F}_2$, and set

$$s(i) \triangleq (-1)^{\ell(\tilde{s}(i))} \in \{\pm 1\}.$$

Then s is uniform on $\{\pm 1\}$ and 4-wise independent.

Step 3 (independence across (k,t)). Instantiate $(h_t^{(k)}, s_t^{(k)})$ independently for each (k,t).

Justification. Degree-d random polynomials over a field yield $(d+1)$-wise independence by interpolation: given any $(d+1)$ distinct inputs, the polynomial values are jointly uniform because the coefficient vector is uniform and the Vandermonde system is invertible. Rejection sampling preserves exact uniformity when mapping $\mathbb{F}_{p_k}$ to $[D_k]$. The sign construction is the analogous argument over $\mathbb{F}_{2^w}$ composed with a nontrivial linear functional. See standard universal hashing references (e.g., (Stinson, 1994)). △

Proposition C.4 (Inner-product estimation for the degree-k polynomial kernel)**.** *Fix $k \geq 1$ and $D \geq 1$, and let $\mathrm{TS}_k(\cdot; D)$ be as in Definition C.1. Define*

$$\widehat{\kappa}_k(\mathbf{x}, \mathbf{y}) \triangleq \big\langle \mathrm{TS}_k(\mathbf{x}; D), \mathrm{TS}_k(\mathbf{y}; D) \big\rangle.$$

Under the same limited-independence regime used in (Pham & Pagh, 2013) (in particular, sufficient for the moment computations below), for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{d'}$,

$$\mathbb{E}\big[\widehat{\kappa}_k(\mathbf{x}, \mathbf{y})\big] = \langle \mathbf{x}, \mathbf{y} \rangle^k, \tag{47}$$

and

$$\mathrm{Var}\big[\widehat{\kappa}_k(\mathbf{x}, \mathbf{y})\big] \leq \frac{1}{D}\Big(\langle \mathbf{x}, \mathbf{y} \rangle^{2k} + \|\mathbf{x}\|_2^{2k} \|\mathbf{y}\|_2^{2k} \Big). \tag{48}$$

Proof. **Step 1 (reduce to CountSketch on the lifted space).** By (46), we can write

$$\mathrm{TS}_k(\mathbf{x}; D) = \mathbf{C}\,\mathbf{u}, \qquad \mathrm{TS}_k(\mathbf{y}; D) = \mathbf{C}\,\mathbf{v},$$

where $\mathbf{u} \triangleq \mathrm{vec}(\mathbf{x}^{\otimes k}) \in \mathbb{R}^{(d')^k}$, $\mathbf{v} \triangleq \mathrm{vec}(\mathbf{y}^{\otimes k}) \in \mathbb{R}^{(d')^k}$, and $\mathbf{C}$ is the corresponding CountSketch matrix $\mathbf{C}^{(\otimes k)}$ (we suppress the superscript for readability). Therefore

$$\widehat{\kappa}_k(\mathbf{x}, \mathbf{y}) = \langle \mathbf{C}\mathbf{u}, \mathbf{C}\mathbf{v} \rangle.$$

Step 2 (unbiasedness of CountSketch inner products). For a CountSketch matrix with entries as in (38),

$$\langle \mathbf{C}\mathbf{u}, \mathbf{C}\mathbf{v} \rangle = \sum_{b \in [D]} \Big(\sum_{p:\, h(p) = b} s(p) u_p \Big) \Big(\sum_{q:\, h(q) = b} s(q) v_q \Big) = \sum_{p,q} s(p) s(q)\, u_p v_q\, \mathbf{1}\{h(p) = h(q)\}.$$

Split the sum into diagonal and off-diagonal parts:

$$\langle \mathbf{C}\mathbf{u}, \mathbf{C}\mathbf{v} \rangle = \sum_{p} u_p v_p + \sum_{p \neq q} s(p) s(q)\, u_p v_q\, \mathbf{1}\{h(p) = h(q)\}.$$

The first term is $\langle \mathbf{u}, \mathbf{v} \rangle$. For the second term, fix $p \neq q$. Since $s(p)$ and $s(q)$ are independent Rademacher variables, $\mathbb{E}[s(p)s(q)] = \mathbb{E}[s(p)]\mathbb{E}[s(q)] = 0$, and this expectation is independent of $\mathbf{1}\{h(p) = h(q)\}$. Hence each off-diagonal term has zero expectation, so

$$\mathbb{E}\big[\langle \mathbf{C}\mathbf{u}, \mathbf{C}\mathbf{v} \rangle\big] = \langle \mathbf{u}, \mathbf{v} \rangle.$$

Step 3 (identify the lifted inner product). By the tensor-product identity,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle \mathrm{vec}(\mathbf{x}^{\otimes k}), \mathrm{vec}(\mathbf{y}^{\otimes k}) \rangle = \langle \mathbf{x}, \mathbf{y} \rangle^k.$$

Combining Steps 2–3 gives (47).

Step 4 (variance bound via second-moment expansion). Let

$$\Delta \triangleq \langle \mathbf{Cu}, \mathbf{Cv} \rangle - \langle \mathbf{u}, \mathbf{v} \rangle = \sum_{p \neq q} s(p)s(q)\, u_p v_q\, \mathbf{1}\{h(p) = h(q)\}.$$

Then $\mathbb{E}[\Delta] = 0$, so $\mathrm{Var}[\widehat{\kappa}_k(\mathbf{x}, \mathbf{y})] = \mathbb{E}[\Delta^2]$. Expand:

$$\mathbb{E}[\Delta^2] = \sum_{p \neq q} \sum_{r \neq t} u_p v_q u_r v_t\, \mathbb{E}\Big[s(p)s(q)s(r)s(t)\, \mathbf{1}\{h(p) = h(q)\}\mathbf{1}\{h(r) = h(t)\} \Big].$$

Under 4-wise independence of $s(\cdot)$ and 2-wise independence of $h(\cdot)$, the expectation above is zero unless each sign variable appears an even number of times. The only index patterns that can contribute are:

$$(p, q) = (r, t) \quad \text{or} \quad (p, q) = (t, r).$$

(Other patterns force at least one sign to appear exactly once, giving zero expectation.) We handle these two cases.

Case 1: $(p, q) = (r, t)$. Then $s(p)s(q)s(r)s(t) = s(p)^2 s(q)^2 = 1$, and $\mathbb{E}[\mathbf{1}\{h(p) = h(q)\}] = \mathbb{P}[h(p) = h(q)] = 1/D$ for $p \neq q$. Thus the contribution from this case is

$$\sum_{p \neq q} u_p^2 v_q^2 \cdot \frac{1}{D} \;\leq\; \frac{1}{D} \sum_p u_p^2 \sum_q v_q^2 = \frac{1}{D} \|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2.$$

Case 2: $(p, q) = (t, r)$. Then $s(p)s(q)s(r)s(t) = s(p)s(q)s(q)s(p) = 1$, and again $\mathbb{E}[\mathbf{1}\{h(p) = h(q)\}] = 1/D$ for $p \neq q$. The contribution from this case is

$$\sum_{p \neq q} u_p v_q u_q v_p \cdot \frac{1}{D} = \frac{1}{D} \Big(\big(\sum_p u_p v_p \big)^2 - \sum_p u_p^2 v_p^2 \Big) \;\leq\; \frac{1}{D} \langle \mathbf{u}, \mathbf{v} \rangle^2.$$

Combining the two cases yields the standard CountSketch second-moment bound:

$$\mathrm{Var}[\widehat{\kappa}_k(\mathbf{x}, \mathbf{y})] = \mathbb{E}[\Delta^2] \;\leq\; \frac{1}{D} \Big(\|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2 + \langle \mathbf{u}, \mathbf{v} \rangle^2 \Big).$$

Step 5 (translate back to $\mathbf{x}, \mathbf{y}$). For tensor lifts, $\|\mathbf{u}\|_2^2 = \|\mathrm{vec}(\mathbf{x}^{\otimes k})\|_2^2 = \|\mathbf{x}\|_2^{2k}$ and $\|\mathbf{v}\|_2^2 = \|\mathbf{y}\|_2^{2k}$, while $\langle \mathbf{u}, \mathbf{v} \rangle^2 = \langle \mathbf{x}, \mathbf{y} \rangle^{2k}$. Substituting these identities gives (48). □

Corollary C.5 (Uniform variance control under norm bounds)**.** *If $\|\mathbf{x}\|_2 \leq R$ and $\|\mathbf{y}\|_2 \leq R$, then*

$$\mathrm{Var}\big[\widehat{\kappa}_k(\mathbf{x}, \mathbf{y})\big] \;\leq\; \frac{2R^{4k}}{D}.$$

In particular, if $\|\widetilde{\mathbf{G}}_{j,:}\|_2 \leq 1/\tau_{\mathrm{g}}$ for all $j \in [M]$, then the degree-dependent factor in (48) is uniformly controlled over all compressed items.

Proof. By Cauchy–Schwarz, $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\|_2 \|\mathbf{y}\|_2 \leq R^2$, hence $\langle \mathbf{x}, \mathbf{y} \rangle^{2k} \leq R^{4k}$ and $\|\mathbf{x}\|_2^{2k} \|\mathbf{y}\|_2^{2k} \leq R^{4k}$. Substitute these bounds into (48) to obtain $\mathrm{Var}[\widehat{\kappa}_k(\mathbf{x}, \mathbf{y})] \leq \frac{1}{D}(R^{4k} + R^{4k}) = 2R^{4k}/D$. The final statement follows by taking $R = 1/\tau_{\mathrm{g}}$. □

D. Algorithmic Summary

Algorithm 1 summarizes the PLASH block in the standard $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ interface. It implements the three-stage pipeline used throughout the paper: *(Stage I)* compress $(\mathbf{K}, \mathbf{V})$ from length N_k to length $M \ll N_k$ using key-based routing; *(Stage II)* enrich each compressed item using selective higher-order feature sketches and a linear readout; then apply a short-sequence mixer across the M items; *(Stage III)* project to $(\mathbf{K}_g, \mathbf{V}_g)$ and perform an *exact* scaled dot-product softmax readout.

What is randomized. All randomness is confined to the TensorSketch calls $\mathrm{TensorSketch}_k(\widetilde{\mathbf{G}}_{j,:}; D_k, \cdot)$ in Stage II. The routing, normalization, mixer, projections, and final attention readout are deterministic given inputs and parameters. This separation is the reason our deviation analysis can certify Stage II and then propagate the bound through deterministic post-processing.

Two invariants that guide reading. The algorithm is easiest to read by tracking two invariants:

- **Shape invariant.** After Stage I, every object used in the expensive interaction with $\mathbf{Q}$ has length M (not N_k). Hence Stage III forms only $N_q \times M$ logits.
- **Norm-control invariant.** Each Stage II feature vector $\widetilde{\mathbf{G}}_{j,:}$ is deterministically bounded in norm (up to the ε_g floor and the τ_{g} scaling). This is what makes the sketch discrepancy bounds depend monotonically on τ_{g}.

Correctness of the deterministic parts. We state two simple lemmas that justify the deterministic steps used by Algorithm 1.

Lemma D.1 (Row-stochastic routing matrix)**.** *Let $\mathbf{A} \triangleq \mathrm{softmax}(\mathbf{S}/\tau, \mathrm{dim} = -1)$ be defined row-wise as in Algorithm 1. Then each row of $\mathbf{A}$ is a probability vector: for all $i \in [N_k]$, we have $\mathbf{A}_{i,j} \geq 0$ for all $j \in [M]$ and $\sum_{j=1}^{M} \mathbf{A}_{i,j} = 1$.*

Proof. Fix any row i. By definition of softmax, $\mathbf{A}_{i,j} = \exp(\mathbf{S}_{i,j}/\tau)\big/\big(\sum_{\ell=1}^{M} \exp(\mathbf{S}_{i,\ell}/\tau)\big)$. Exponentials are nonnegative, so $\mathbf{A}_{i,j} \geq 0$. Summing over j yields $\sum_{j=1}^{M} \mathbf{A}_{i,j} = \sum_{j=1}^{M} \exp(\mathbf{S}_{i,j}/\tau)\big/\big(\sum_{\ell=1}^{M} \exp(\mathbf{S}_{i,\ell}/\tau)\big) = 1$. □

Lemma D.2 (Deterministic norm control in Stage II)**.** *In Algorithm 1, for every $j \in [M]$,*

$$\widetilde{\mathbf{G}}_{j,:} = \frac{\mathbf{G}_{j,:}}{\max\{\|\mathbf{G}_{j,:}\|_2, \varepsilon_g\} \cdot \tau_{\mathrm{g}}}.$$

Then $\|\widetilde{\mathbf{G}}_{j,:}\|_2 \leq 1/\tau_{\mathrm{g}}$. Moreover, if $\|\mathbf{G}_{j,:}\|_2 \geq \varepsilon_g$, then $\|\widetilde{\mathbf{G}}_{j,:}\|_2 = 1/\tau_{\mathrm{g}}$.

Proof. Let $c_j \triangleq \max\{\|\mathbf{G}_{j,:}\|_2, \varepsilon_g\} \cdot \tau_{\mathrm{g}}$. By construction, $c_j > 0$. Then

$$\|\widetilde{\mathbf{G}}_{j,:}\|_2 = \frac{\|\mathbf{G}_{j,:}\|_2}{\max\{\|\mathbf{G}_{j,:}\|_2, \varepsilon_g\} \cdot \tau_{\mathrm{g}}} \leq \frac{1}{\tau_{\mathrm{g}}}.$$

If $\|\mathbf{G}_{j,:}\|_2 \geq \varepsilon_g$, then $\max\{\|\mathbf{G}_{j,:}\|_2, \varepsilon_g\} = \|\mathbf{G}_{j,:}\|_2$, so the fraction equals $1/\tau_{\mathrm{g}}$. □

Why the TensorSketch call is the only randomness. The following proposition formalizes the "randomness localized" statement used throughout the paper.

Proposition D.3 (Randomness localization in Algorithm 1)**.** *Fix all parameters and inputs in Algorithm 1, including $\mathbf{P}$, τ, τ_{g}, ε_g, ψ, $\mathbf{W}_{\mathrm{out}}$, $\mathrm{Mixer}_{L_{\mathrm{mix}}}$, $\mathbf{W}_K$, and $\mathbf{W}_V$. Assume the TensorSketch seeds $\{(h_t^{(k)}, s_t^{(k)})\}$ are the only random objects. Then every intermediate quantity produced by Stages I and III and the mixer at Stage II, is a deterministic function of $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ and the fixed parameters; the only random intermediate variables are the Stage II sketch outputs $\{\mathbf{z}_k\}$ (and thus $\{\mathbf{v}_j\}$ and $\mathbf{Y}_{\mathrm{enh}}$) generated by $\mathrm{TensorSketch}_k(\cdot)$.*

Proof. We proceed stage by stage and show that each line is deterministic unless it calls TensorSketch.

Stage I. The matrix product $\mathbf{S} = \mathbf{K}\mathbf{P}^\top$ is deterministic. Applying softmax row-wise with a fixed temperature τ is deterministic, so $\mathbf{A}$ is deterministic. Matrix multiplications $\widetilde{\mathbf{K}} = \mathbf{A}^\top\mathbf{K}$ and $\widetilde{\mathbf{V}} = \mathbf{A}^\top\mathbf{V}$ are deterministic.

Stage II (before sketching). Concatenation $\mathbf{U} = [\widetilde{\mathbf{K}} \bigoplus \widetilde{\mathbf{V}}]$ is deterministic. Applying the fixed pre-map ψ row-wise yields deterministic $\mathbf{G}$. The normalization step computes $\widetilde{\mathbf{G}}_{j,:}$ via fixed arithmetic operations (max, norm, division) with fixed ε_g and τ_{g}, hence it is deterministic.

Stage II (sketching). The only place where the algorithm references the seeds $\{(h_t^{(k)}, s_t^{(k)})\}$ is the call $\mathbf{z}_k \leftarrow \mathrm{TensorSketch}_k(\widetilde{\mathbf{G}}_{j,:}; D_k, \{(h_t^{(k)}, s_t^{(k)})\}_{t=1}^{k})$. Therefore $\mathbf{z}_k$ is the only random output at this point. The subsequent concatenation $\mathbf{v}_j \leftarrow \mathbf{v}_j \bigoplus(\beta_k\, \mathbf{z}_k)$ and linear map $\mathbf{Y}_{\mathrm{enh}}[j,:] \leftarrow \mathbf{W}_{\mathrm{out}}\mathbf{v}_j$ are deterministic given $\mathbf{z}_k$.

The mixer at Stage II and Stage III. The mixer is deterministic given $\mathbf{Y}_{\mathrm{enh}}$, so $\mathbf{Z}_{\mathrm{out}}$ is deterministic conditional on the Stage II sketches. The projections $\mathbf{K}_g = \mathbf{Z}_{\mathrm{out}}\mathbf{W}_K$ and $\mathbf{V}_g = \mathbf{Z}_{\mathrm{out}}\mathbf{W}_V$ are deterministic. Finally, the attention readout

Algorithm 1 PLASH block in $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$

Input: $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$, $\mathbf{K} \in \mathbb{R}^{N_k \times d_k}$, $\mathbf{V} \in \mathbb{R}^{N_k \times d_v}$.
Input: length $M \ll N_k$; prototypes $\mathbf{P} \in \mathbb{R}^{M \times d_k}$; temperatures $\tau, \tau_{\mathrm{g}} > 0$; stability $\varepsilon_g > 0$.
Input: pre-map $\psi : \mathbb{R}^{d_k + d_v} \to \mathbb{R}^{d'}$ (row-wise).
Input: degrees $\mathcal{K} \subset \mathbb{Z}_{\geq 1}$; weights $\{\beta_k\}_{k \in \mathcal{K}}$; sketch sizes $\{D_k\}_{k \in \mathcal{K}}$; fixed TensorSketch seeds $\{(h_t^{(k)}, s_t^{(k)})\}$.
Input: $\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{d \times D_{\mathrm{tot}}}$ with $D_{\mathrm{tot}} = \sum_{k \in \mathcal{K}} D_k$; mixer $\mathrm{Mixer}_{L_{\mathrm{mix}}}$.
Input: projections $\mathbf{W}_K \in \mathbb{R}^{d \times d_k}$, $\mathbf{W}_V \in \mathbb{R}^{d \times d_v}$.
Output: $\mathbf{Y} \in \mathbb{R}^{N_q \times d_v}$.

Stage I: K-V Compressing
$\mathbf{S} \leftarrow \mathbf{K}\mathbf{P}^\top$ // $\mathbf{S} \in \mathbb{R}^{N_k \times M}$
$\mathbf{A} \leftarrow \mathrm{softmax}(\mathbf{S}/\tau, \mathrm{dim} = -1)$ // row-wise over M
$\widetilde{\mathbf{K}} \leftarrow \mathbf{A}^\top \mathbf{K}; \quad \widetilde{\mathbf{V}} \leftarrow \mathbf{A}^\top \mathbf{V}$ // both length M

Stage II: Randomized High-order Feature Lifting
$\mathbf{U} \leftarrow [\widetilde{\mathbf{K}} \bigoplus \widetilde{\mathbf{V}}]$ // row-wise concat, $\mathbf{U} \in \mathbb{R}^{M \times (d_k + d_v)}$
$\mathbf{G} \leftarrow \psi(\mathbf{U})$ // $\mathbf{G} \in \mathbb{R}^{M \times d'}$
for $j = 1$ **to** M **do**
  $\widetilde{\mathbf{G}}_{j,:} \leftarrow \mathbf{G}_{j,:} \big/ \big(\max\{\|\mathbf{G}_{j,:}\|_2, \varepsilon_g\} \cdot \tau_{\mathrm{g}} \big)$ // stabilized norm control
  for each $k \in \mathcal{K}$ **do**
    $\mathbf{z}_k \leftarrow \mathrm{TensorSketch}_k(\widetilde{\mathbf{G}}_{j,:}; D_k, \{(h_t^{(k)}, s_t^{(k)})\}_{t=1}^k) \in \mathbb{R}^{D_k}$
    $\mathbf{v}_j \leftarrow \mathbf{v}_j \bigoplus (\beta_k \, \mathbf{z}_k)$
  end for
  $\mathbf{Y}_{\mathrm{enh}}[j, :] \leftarrow \mathbf{W}_{\mathrm{out}} \mathbf{v}_j$ // $\mathbf{Y}_{\mathrm{enh}} \in \mathbb{R}^{M \times d}$
end for
$\mathbf{Z}_{\mathrm{out}} \leftarrow \mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{Y}_{\mathrm{enh}})$ // $\mathbf{Z}_{\mathrm{out}} \in \mathbb{R}^{M \times d}$

Stage III: Final Output
$\mathbf{K}_g \leftarrow \mathbf{Z}_{\mathrm{out}} \mathbf{W}_K; \quad \mathbf{V}_g \leftarrow \mathbf{Z}_{\mathrm{out}} \mathbf{W}_V$ // $\mathbf{K}_g \in \mathbb{R}^{M \times d_k}$, $\mathbf{V}_g \in \mathbb{R}^{M \times d_v}$
$\mathbf{Y} \leftarrow \mathrm{softmax}\left(\mathbf{Q}\mathbf{K}_g^\top / \sqrt{d_k}, \mathrm{dim} = -1\right) \mathbf{V}_g$ // exact attention
return $\mathbf{Y}$

$\mathbf{Y} = \mathrm{softmax}(\mathbf{Q}\mathbf{K}_g^\top / \sqrt{d_k}, \mathrm{dim} = -1)\mathbf{V}_g$ is deterministic given $(\mathbf{Q}, \mathbf{K}_g, \mathbf{V}_g)$. Hence the only random variables are those created by the TensorSketch calls. □

How this appendix connects to the theory in the main text. Proposition D.3 is the algorithmic reason the error analysis can proceed in two steps: (i) bound the Stage II sketch discrepancy at the feature / embedding level (Theorem 3.1), and (ii) propagate that discrepancy through deterministic post-processing to obtain a forward-pass certificate on the final output (Theorem 3.2). Lemma D.2 identifies the deterministic knob τ_{g} that tightens the Stage II discrepancy bound.

E. A deterministic stability bound for softmax attention

This appendix records deterministic inequalities that propagate any perturbation on the key / value side to an end-to-end perturbation of the attention output. The key point is simple: *softmax attention is stable to small changes in logits and values*, and we make this stability explicit in the row-wise $(2, \infty)$ norm so it composes cleanly with our Stage I / Stage II analysis.

Lemma E.1 (Deterministic perturbation bound for softmax attention)**.** *Fix $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$. Let $(\mathbf{K}, \mathbf{V})$ and $(\mathbf{K}', \mathbf{V}')$ be two key/value pairs with $\mathbf{K}, \mathbf{K}' \in \mathbb{R}^{N \times d_k}$ and $\mathbf{V}, \mathbf{V}' \in \mathbb{R}^{N \times d_v}$. Define the rowwise $(2, \infty)$ matrix norm*

$$\|\mathbf{X}\|_{2,\infty} \triangleq \max_i \|\mathbf{X}_{i,:}\|_2,$$

and the perturbation magnitudes

$$\rho_K \triangleq \|\mathbf{K} - \mathbf{K}'\|_{2,\infty}, \qquad \rho_V \triangleq \|\mathbf{V} - \mathbf{V}'\|_{2,\infty}.$$

Define also the query-dependent sensitivity factor and a value-row bound

$$\Gamma_Q^{(N)} \triangleq \frac{1}{\sqrt{d_k}}\|\mathbf{Q}\|_{2,\infty}, \qquad V_{\max} \triangleq \|\mathbf{V}\|_{2,\infty}.$$

Then

$$\big\|\mathrm{Atten}(\mathbf{Q};\mathbf{K},\mathbf{V}) - \mathrm{Atten}(\mathbf{Q};\mathbf{K}',\mathbf{V}')\big\|_F \le \sqrt{N_q}\cdot\Big(\Gamma_Q^{(N)}\cdot\rho_K\cdot V_{\max} + \rho_V\Big). \tag{49}$$

Proof. We prove a uniform bound for a single query row and then lift it to the Frobenius norm.

Setup for one query. Fix an index $j \in [N_q]$ and write the jth query row as $\mathbf{q} \triangleq \mathbf{Q}_{j,:} \in \mathbb{R}^{d_k}$. Define the (scaled) logit vectors

$$\mathbf{s} \triangleq \frac{\mathbf{q}\mathbf{K}^\top}{\sqrt{d_k}} \in \mathbb{R}^N, \qquad \mathbf{s}' \triangleq \frac{\mathbf{q}\mathbf{K}'^\top}{\sqrt{d_k}} \in \mathbb{R}^N,$$

and the corresponding softmax weight vectors

$$\boldsymbol{\alpha} \triangleq \mathrm{softmax}(\mathbf{s}) \in \mathbb{R}^N, \qquad \boldsymbol{\alpha}' \triangleq \mathrm{softmax}(\mathbf{s}') \in \mathbb{R}^N.$$

The associated attention output rows are

$$\mathbf{y} \triangleq \boldsymbol{\alpha}^\top\mathbf{V} \in \mathbb{R}^{d_v}, \qquad \mathbf{y}' \triangleq \boldsymbol{\alpha}'^\top\mathbf{V}' \in \mathbb{R}^{d_v}.$$

We decompose

$$\begin{aligned}\mathbf{y} - \mathbf{y}' &= \boldsymbol{\alpha}^\top\mathbf{V} - \boldsymbol{\alpha}'^\top\mathbf{V}' \\ &= \underbrace{(\boldsymbol{\alpha}-\boldsymbol{\alpha}')^\top\mathbf{V}}_{\text{weight change}} + \underbrace{\boldsymbol{\alpha}'^\top(\mathbf{V}-\mathbf{V}')}_{\text{value change}}.\end{aligned} \tag{50}$$

Step 1: bound the value-change term. Because $\boldsymbol{\alpha}' = \mathrm{softmax}(\mathbf{s}')$ is a probability vector, $\alpha_i' \ge 0$ and $\sum_{i=1}^N \alpha_i' = 1$. Hence,

$$\begin{aligned}\big\|\boldsymbol{\alpha}'^\top(\mathbf{V}-\mathbf{V}')\big\|_2 &= \left\|\sum_{i=1}^N \alpha_i'\cdot\big(\mathbf{V}_{i,:} - \mathbf{V}'_{i,:}\big)\right\|_2 \\ &\le \sum_{i=1}^N \alpha_i'\cdot\big\|\mathbf{V}_{i,:} - \mathbf{V}'_{i,:}\big\|_2 \qquad \text{(triangle inequality)} \\ &\le \max_{i\in[N]}\big\|\mathbf{V}_{i,:} - \mathbf{V}'_{i,:}\big\|_2 \\ &= \rho_V.\end{aligned}$$

Step 2: bound the weight-change term via an ℓ_1 control. We first reduce the problem to bounding $\|\boldsymbol{\alpha}-\boldsymbol{\alpha}'\|_1$. Write

$$(\boldsymbol{\alpha}-\boldsymbol{\alpha}')^\top\mathbf{V} = \sum_{i=1}^N(\alpha_i - \alpha_i')\mathbf{V}_{i,:}.$$

Then, by the triangle inequality,

$$\begin{aligned}\big\|(\boldsymbol{\alpha}-\boldsymbol{\alpha}')^\top\mathbf{V}\big\|_2 &\le \sum_{i=1}^N|\alpha_i-\alpha_i'|\cdot\|\mathbf{V}_{i,:}\|_2 \\ &\le \Big(\max_{i\in[N]}\|\mathbf{V}_{i,:}\|_2\Big)\cdot\sum_{i=1}^N|\alpha_i-\alpha_i'| \\ &= V_{\max}\cdot\|\boldsymbol{\alpha}-\boldsymbol{\alpha}'\|_1.\end{aligned} \tag{51}$$

Step 3: Lipschitz control of the softmax weights by the logits. By Lemma B.2 (softmax is $\ell_\infty \to \ell_1$ Lipschitz),

$$\|\boldsymbol{\alpha} - \boldsymbol{\alpha}'\|_1 \le \|\mathbf{s} - \mathbf{s}'\|_\infty. \tag{52}$$

It remains to bound $\|\mathbf{s} - \mathbf{s}'\|_\infty$ using the key perturbation.

For each $i \in [N]$, we have

$$\begin{aligned}
|\mathbf{s}_i - \mathbf{s}'_i| &= \left| \frac{\langle \mathbf{q}, \mathbf{K}_{i,:} \rangle}{\sqrt{d_k}} - \frac{\langle \mathbf{q}, \mathbf{K}'_{i,:} \rangle}{\sqrt{d_k}} \right| \\
&= \frac{1}{\sqrt{d_k}} \cdot \left| \langle \mathbf{q}, \mathbf{K}_{i,:} - \mathbf{K}'_{i,:} \rangle \right| \\
&\le \frac{1}{\sqrt{d_k}} \cdot \|\mathbf{q}\|_2 \cdot \|\mathbf{K}_{i,:} - \mathbf{K}'_{i,:}\|_2 \qquad \text{(Cauchy–Schwarz)} \\
&\le \frac{1}{\sqrt{d_k}} \cdot \|\mathbf{Q}\|_{2,\infty} \cdot \|\mathbf{K} - \mathbf{K}'\|_{2,\infty} \\
&= \Gamma_Q^{(N)} \cdot \rho_K.
\end{aligned}$$

Taking the maximum over i yields

$$\|\mathbf{s} - \mathbf{s}'\|_\infty \le \Gamma_Q^{(N)} \cdot \rho_K. \tag{53}$$

Combining (51), (52), and (53) gives

$$\left\| (\boldsymbol{\alpha} - \boldsymbol{\alpha}')^\top \mathbf{V} \right\|_2 \le V_{\max} \cdot \Gamma_Q^{(N)} \cdot \rho_K. \tag{54}$$

Step 4: conclude a per-row bound. Applying the triangle inequality to (50) and using Step 1 and (54),

$$\|\mathbf{y} - \mathbf{y}'\|_2 \le \Gamma_Q^{(N)} \cdot \rho_K \cdot V_{\max} + \rho_V. \tag{55}$$

Importantly, the right-hand side does not depend on j, so the same bound holds for every query row.

Step 5: lift to the Frobenius norm. Let $\mathbf{Y} \triangleq \mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V})$ and $\mathbf{Y}' \triangleq \mathrm{Atten}(\mathbf{Q}; \mathbf{K}', \mathbf{V}')$. Then (55) implies

$$\begin{aligned}
\|\mathbf{Y} - \mathbf{Y}'\|_F^2 &= \sum_{j=1}^{N_q} \|\mathbf{Y}_{j,:} - \mathbf{Y}'_{j,:}\|_2^2 \\
&\le \sum_{j=1}^{N_q} \left(\Gamma_Q^{(N)} \cdot \rho_K \cdot V_{\max} + \rho_V \right)^2 \\
&= N_q \cdot \left(\Gamma_Q^{(N)} \cdot \rho_K \cdot V_{\max} + \rho_V \right)^2.
\end{aligned}$$

Taking square roots yields (49). □

F. A forward-pass deterministic comparator for Stage I

This section builds an *explicit, forward-pass computable* deterministic comparator for Stage I. Stage I compresses the key / value side by routing each key to one of M prototypes. We do *not* assume anything about how the prototypes were learned. Instead, for a fixed forward pass, we: (i) replace soft routing by a deterministic hard routing induced by the current logits, and (ii) quantify the resulting distortion through radii that can be computed from the forward pass.

Assumption F.1. Assume (as is standard in attention) that keys and prototypes are ℓ_2-normalized rowwise, i.e.,

$$\|\mathbf{K}_{i,:}\|_2 = 1 \text{ for all } i, \qquad \|\mathbf{P}_{j,:}\|_2 = 1 \text{ for all } j. \tag{56}$$

△

Fix a macro length $M \ll N_k$ and prototypes $\mathbf{P} \in \mathbb{R}^{M \times d_k}$. Stage I forms routing logits $\mathbf{S}$ and soft routing weights $\mathbf{A}$.

Hard-routing comparator. Fix one forward pass. Define the (deterministic) hard assignment

$$c(i) \triangleq \arg\max_{j \in [M]} \mathbf{S}_{i,j}, \qquad i \in [N_k],$$

and the associated hard routing matrix

$$\mathbf{A}^{\mathrm{hard}}_{i,j} \triangleq \mathbf{1}\{j = c(i)\}, \qquad (i,j) \in [N_k] \times [M].$$

Define cluster sizes

$$n_j \triangleq \sum_{i=1}^{N_k} \mathbf{A}^{\mathrm{hard}}_{i,j}, \qquad j \in [M],$$

and for each j with $n_j > 0$ define the cluster means

$$\bar{\mathbf{K}}[j,:] \triangleq \frac{1}{n_j} \sum_{i:\, c(i)=j} \mathbf{K}_{i,:}, \qquad \bar{\mathbf{V}}[j,:] \triangleq \frac{1}{n_j} \sum_{i:\, c(i)=j} \mathbf{V}_{i,:}, \tag{57}$$

and set $\bar{\mathbf{K}}[j,:] \triangleq \mathbf{0}$ and $\bar{\mathbf{V}}[j,:] \triangleq \mathbf{0}$ when $n_j = 0$.

Reconstruction operator (new technical term). Define the *hard-routing reconstruction operator* $\mathcal{R}_c : \mathbb{R}^{N_k \times d} \to \mathbb{R}^{N_k \times d}$ by

$$\Big(\mathcal{R}_c(\mathbf{X})\Big)_{i,:} \triangleq \begin{cases} \frac{1}{n_{c(i)}} \sum\limits_{i':\, c(i')=c(i)} \mathbf{X}_{i',:}, & n_{c(i)} > 0, \\ \mathbf{0}, & n_{c(i)} = 0, \end{cases} \tag{58}$$

where $i \in [N_k]$. With this notation the quantized (reconstructed) keys and values are

$$\mathbf{K}^{\mathrm{q}} \triangleq \mathcal{R}_c(\mathbf{K}), \qquad \mathbf{V}^{\mathrm{q}} \triangleq \mathcal{R}_c(\mathbf{V}), \tag{59}$$

i.e.,

$$\mathbf{K}^{\mathrm{q}}_{i,:} \triangleq \bar{\mathbf{K}}[c(i),:], \qquad \mathbf{V}^{\mathrm{q}}_{i,:} \triangleq \bar{\mathbf{V}}[c(i),:].$$

Forward-pass radii. Define the (deterministic) Stage I radii

$$\rho_K(M) \triangleq \|\mathbf{K} - \mathbf{K}^{\mathrm{q}}\|_{2,\infty}, \qquad \rho_V(M) \triangleq \|\mathbf{V} - \mathbf{V}^{\mathrm{q}}\|_{2,\infty}. \tag{60}$$

Let

$$\mathbf{Y}_{\mathrm{q}} \triangleq \mathrm{Atten}(\mathbf{Q}; \mathbf{K}^{\mathrm{q}}, \mathbf{V}^{\mathrm{q}}), . \tag{61}$$

Recall $\mathbf{Y}_{\mathrm{soft}} = \mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V})$ is defined in (1).

Lemma F.2 (Stage I comparator bound)**.** *Define*

$$V_{\max} \triangleq \|\mathbf{V}\|_{2,\infty}, \qquad \Gamma_Q^{(N_k)} \triangleq \frac{\|\mathbf{Q}\|_{2,\infty}}{\sqrt{d_k}}.$$

Then the end-to-end deviation between the original attention and the hard-routing comparator satisfies

$$\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{q}}\|_F \;\le\; \sqrt{N_q} \cdot \Big(\Gamma_Q^{(N_k)} \cdot \rho_K(M) \cdot V_{\max} + \rho_V(M)\Big). \tag{62}$$

Proof. We provide the argument in explicit steps.

Step 1 (identify the perturbation). By definition, $\mathbf{Y}_{\text{soft}} = \text{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V})$ and $\mathbf{Y}_{\text{q}} = \text{Atten}(\mathbf{Q}; \mathbf{K}^{\text{q}}, \mathbf{V}^{\text{q}})$. Thus $\mathbf{Y}_{\text{soft}} - \mathbf{Y}_{\text{q}}$ is the output difference of the same attention map under a perturbation of keys / values:

$$(\mathbf{K}, \mathbf{V}) \mapsto (\mathbf{K}^{\text{q}}, \mathbf{V}^{\text{q}}).$$

Step 2 (invoke the generic KV-perturbation bound). Apply Lemma E.1 with $N = N_k$ and

$$(\mathbf{K}', \mathbf{V}') \triangleq (\mathbf{K}^{\text{q}}, \mathbf{V}^{\text{q}}).$$

That lemma upper bounds $\| \text{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V}) - \text{Atten}(\mathbf{Q}; \mathbf{K}', \mathbf{V}')\|_F$ by a quantity that depends on $\|\mathbf{K} - \mathbf{K}'\|_{2,\infty}$ and $\|\mathbf{V} - \mathbf{V}'\|_{2,\infty}$, scaled by $\sqrt{N_q}$ and $\Gamma_Q^{(N_k)}$.

Step 3 (substitute the Stage I radii). By (60) and the choice of $(\mathbf{K}', \mathbf{V}')$,

$$\|\mathbf{K} - \mathbf{K}'\|_{2,\infty} = \|\mathbf{K} - \mathbf{K}^{\text{q}}\|_{2,\infty} = \rho_K(M), \qquad \|\mathbf{V} - \mathbf{V}'\|_{2,\infty} = \|\mathbf{V} - \mathbf{V}^{\text{q}}\|_{2,\infty} = \rho_V(M).$$

Substituting these into Lemma E.1 yields (62). □

Vanishing error as M increases (rigorous sufficient condition). Lemma F.2 shows it suffices to make $\rho_K(M) \to 0$ and $\rho_V(M) \to 0$ as $M \to \infty$. We next give a standard covering-number condition that guarantees such decay.

Definition F.3 (ϵ-net (covering number) in ℓ_2)**.** For a set $\mathcal{S} \subset \mathbb{R}^{d_k}$ and $\epsilon > 0$, an *ϵ-net* is a finite set $\mathcal{N} \subset \mathbb{R}^{d_k}$ such that for every $\mathbf{x} \in \mathcal{S}$ there exists $\mathbf{p} \in \mathcal{N}$ with $\|\mathbf{x} - \mathbf{p}\|_2 \le \epsilon$. Let $\mathcal{N}(\mathcal{S}, \epsilon)$ denote the minimal cardinality of an ϵ-net of $\mathcal{S}$. △

Theorem F.4 (A sufficient condition for $\rho_K(M) \to 0$ and a quantitative decay bound)**.** *Fix a forward pass and let $\mathcal{S}_K \triangleq \{\mathbf{K}_{i,:} : i \in [N_k]\} \subset \mathbb{R}^{d_k}$. Assume that the prototypes $\{\mathbf{P}_{j,:}\}_{j=1}^M$ form an ϵ-net of $\mathcal{S}_K$ in ℓ_2:*

$$\forall i \in [N_k]\ \exists j \in [M] \text{ s.t. } \|\mathbf{K}_{i,:} - \mathbf{P}_{j,:}\|_2 \le \epsilon. \tag{63}$$

In words, every key vector lies within distance ϵ of some *prototype. (We give a standard construction and a resulting rate after the proof.) Let $c(i) \triangleq \arg\max_{j \in [M]} \langle \mathbf{K}_{i,:}, \mathbf{P}_{j,:} \rangle$ be the induced hard routing. Then the Stage I reconstruction error satisfies*

$$\rho_K(M) \;=\; \|\mathbf{K} - \mathcal{R}_c(\mathbf{K})\|_{2,\infty} \;\le\; 2\epsilon. \tag{64}$$

In particular, if M increases so that there exists $\epsilon_M \to 0$ with (63) holding for $\epsilon = \epsilon_M$, then $\rho_K(M) \to 0$.

Proof. We bound each row error $\|\mathbf{K}_{i,:} - \mathbf{K}^{\text{q}}_{i,:}\|_2$ and then take the maximum over i.

Step 1 (hard routing picks a prototype that is still close). Fix any $i \in [N_k]$ and write $\mathbf{k} \triangleq \mathbf{K}_{i,:}$. By (63), pick $j^\star \in [M]$ such that $\|\mathbf{k} - \mathbf{P}_{j^\star,:}\|_2 \le \epsilon$. Since $c(i)$ maximizes the inner product with $\mathbf{k}$,

$$\langle \mathbf{k}, \mathbf{P}_{c(i),:} \rangle \;\ge\; \langle \mathbf{k}, \mathbf{P}_{j^\star,:} \rangle. \tag{65}$$

Step 2 (inner product maximality implies a distance bound). Using $\|\mathbf{a} - \mathbf{b}\|_2^2 = \|\mathbf{a}\|_2^2 + \|\mathbf{b}\|_2^2 - 2\langle \mathbf{a}, \mathbf{b} \rangle$,

$$\begin{aligned}
\|\mathbf{k} - \mathbf{P}_{c(i),:}\|_2^2 &= \|\mathbf{k}\|_2^2 + \|\mathbf{P}_{c(i),:}\|_2^2 - 2\langle \mathbf{k}, \mathbf{P}_{c(i),:} \rangle \\
&\le \|\mathbf{k}\|_2^2 + \|\mathbf{P}_{c(i),:}\|_2^2 - 2\langle \mathbf{k}, \mathbf{P}_{j^\star,:} \rangle \qquad \text{(by (65))} \\
&= \|\mathbf{k} - \mathbf{P}_{j^\star,:}\|_2^2 + \big(\|\mathbf{P}_{c(i),:}\|_2^2 - \|\mathbf{P}_{j^\star,:}\|_2^2\big).
\end{aligned}$$

Under (56), the parenthetical term is zero, so

$$\|\mathbf{k} - \mathbf{P}_{c(i),:}\|_2^2 \le \|\mathbf{k} - \mathbf{P}_{j^\star,:}\|_2^2 \le \epsilon^2,$$

hence

$$\|\mathbf{K}_{i,:} - \mathbf{P}_{c(i),:}\|_2 \le \epsilon. \tag{66}$$

Since i was arbitrary, (66) holds for all $i \in [N_k]$.

Step 3 (keys routed to the same prototype are within 2ϵ of each other). Fix any $j \in [M]$ and any i, i' with $c(i) = c(i') = j$. By triangle inequality and (66),

$$\|\mathbf{K}_{i,:} - \mathbf{K}_{i',:}\|_2 \le \|\mathbf{K}_{i,:} - \mathbf{P}_{j,:}\|_2 + \|\mathbf{K}_{i',:} - \mathbf{P}_{j,:}\|_2 \le \epsilon + \epsilon = 2\epsilon. \tag{67}$$

Step 4 (distance to the cluster mean is at most the cluster diameter). Fix i and set $j \triangleq c(i)$. Then $n_j \ge 1$ and $\mathbf{K}^{\mathrm{q}}_{i,:} = \bar{\mathbf{K}}[j,:]$. Using (57) and the triangle inequality,

$$\begin{aligned}
\|\mathbf{K}_{i,:} - \bar{\mathbf{K}}[j,:]\|_2 &= \left\| \frac{1}{n_j} \sum_{i' :\, c(i')=j} \big(\mathbf{K}_{i,:} - \mathbf{K}_{i',:}\big) \right\|_2 \\
&\le \frac{1}{n_j} \sum_{i' :\, c(i')=j} \|\mathbf{K}_{i,:} - \mathbf{K}_{i',:}\|_2 \\
&\le \frac{1}{n_j} \sum_{i' :\, c(i')=j} 2\epsilon \\
&= 2\epsilon,
\end{aligned}$$

where the last inequality uses (67). Thus, for every $i \in [N_k]$,

$$\|\mathbf{K}_{i,:} - \mathbf{K}^{\mathrm{q}}_{i,:}\|_2 = \|\mathbf{K}_{i,:} - \bar{\mathbf{K}}[c(i),:]\|_2 \le 2\epsilon.$$

Step 5 (take the maximum over rows). Taking the maximum over i gives

$$\rho_K(M) = \|\mathbf{K} - \mathcal{R}_c(\mathbf{K})\|_{2,\infty} = \max_{i \in [N_k]} \|\mathbf{K}_{i,:} - \mathbf{K}^{\mathrm{q}}_{i,:}\|_2 \le 2\epsilon,$$

which is (64). □

A concrete decay rate via covering numbers (and how to obtain prototypes). Recall the *unit sphere* in $\mathbb{R}^{d_k}$ is

$$\mathbb{S}^{d_k - 1} \triangleq \{\mathbf{x} \in \mathbb{R}^{d_k} : \|\mathbf{x}\|_2 = 1\}.$$

Under (56), every key row lies on the sphere, i.e., $\mathbf{K}_{i,:} \in \mathbb{S}^{d_k-1}$ for all i, so $\mathcal{S}_K \subset \mathbb{S}^{d_k-1}$.

Covering number reminder. For $\epsilon > 0$, the covering number $\mathcal{N}(\mathcal{S}, \epsilon)$ is the smallest number of ℓ_2-balls of radius ϵ whose union contains $\mathcal{S}$ (Definition above). In particular, $\mathcal{N}\big(\mathbb{S}^{d_k-1}, \epsilon\big)$ is the minimum number of radius-ϵ Euclidean balls needed to cover the entire sphere.

Existence and rate. A standard volumetric argument (via packing–covering duality on the sphere) implies

$$\mathcal{N}\big(\mathbb{S}^{d_k-1}, \epsilon\big) \;\le\; \left(\frac{3}{\epsilon}\right)^{d_k}, \qquad \epsilon \in (0, 1),$$

see, e.g., Vershynin (2018, Chapter 4) or Matoušek (2002, Chapter 13). Therefore, if

$$M \;\ge\; \left(\tfrac{3}{\epsilon}\right)^{d_k},$$

then there exists a set of M unit-norm prototypes that forms an ϵ-net of $\mathbb{S}^{d_k-1}$, and hence can satisfy (63) for any $\mathcal{S}_K \subset \mathbb{S}^{d_k-1}$. Equivalently, setting $\epsilon_M \triangleq 3 \cdot M^{-1/d_k}$ gives

$$\rho_K(M) \le 2\epsilon_M = 6 \cdot M^{-1/d_k}$$

by Theorem F.4.

How to obtain such prototypes in practice (for the finite set $\mathcal{S}_K$). The covering bound above is existential over the sphere, but our forward-pass set $\mathcal{S}_K = \{\mathbf{K}_{i,:}\}_{i=1}^{N_k}$ is finite. For a finite set, an ϵ-net is equivalent to choosing M representatives that minimize (or at least control) the worst-case distance $\max_i \min_j \|\mathbf{K}_{i,:} - \mathbf{P}_{j,:}\|_2$. Several standard constructions achieve this efficiently:

- k**-center / farthest-first traversal (greedy** ϵ**-net).** Initialize with any key as the first prototype, then repeatedly add the key farthest (in ℓ_2) from the current prototype set until M prototypes are chosen. This is the classic greedy algorithm for the metric k-center problem and yields a constant-factor approximation to the optimal worst-case radius (Gonzalez, 1985). In particular, the resulting prototypes directly control $\max_i \min_j \|\mathbf{K}_{i,:} - \mathbf{P}_{j,:}\|_2$, which is exactly the ϵ appearing in (63).
- k**-means on normalized keys (Lloyd updates).** Run k-means with $k = M$ on $\{\mathbf{K}_{i,:}\}$ using Lloyd's algorithm (Lloyd, 1982), optionally with k-means++ initialization (Arthur & Vassilvitskii, 2007). Then normalize the resulting centroids to unit norm to satisfy (56). While k-means optimizes average distortion (not worst-case radius), it often produces high-coverage prototypes in practice, especially when the key distribution is concentrated.
- **Learned prototypes (the default in PLASH).** Treat $\mathbf{P}$ as trainable parameters and update them by backpropagation. Empirically, end-to-end training tends to move prototypes toward regions of high key density (as in learned vector quantization / codebook methods (van den Oord et al., 2017)), while the forward-pass quantity $\rho_K(M)$ in (60) provides a direct diagnostic of current coverage of $\mathcal{S}_K$.

All three methods produce unit-norm prototypes and can be used to satisfy (63) (exactly for farthest-first, approximately for k-means and learned prototypes).

Implication for the Stage I comparator error. Substituting $\rho_K(M) \le 6 \cdot M^{-1/d_k}$ into Lemma F.2 yields

$$\|\mathbf{Y}_{\text{soft}} - \mathbf{Y}_{\text{q}}\|_F \le \sqrt{N_q} \cdot \Big(\Gamma_Q^{(N_k)} \cdot \big(6 \cdot M^{-1/d_k}\big) \cdot V_{\max} \; + \; \rho_V(M)\Big),$$

where $\rho_V(M)$ admits an analogous bound under the same type of covering condition in $\mathbb{R}^{d_v}$.

G. Proof of Theorem 3.1: CountSketch as the only source of randomness at Stage II

This section proves Theorem 3.1. The proof is organized to make the logic transparent: (i) we define the deterministic reference features, (ii) we bound their norms deterministically, (iii) we bound the TensorSketch norms with high probability, and (iv) we combine the two bounds via the triangle inequality to obtain a uniform discrepancy bound without explicitly computing any differences.

Stage II inputs and determinism. Stage II takes the normalized features $\widetilde{\mathbf{G}} \in \mathbb{R}^{M \times d'}$ defined in (7). For each $k \in \mathcal{K}$ we interpret $\widetilde{\mathbf{G}}_{j,:}$ as an element of $\mathbb{R}^{D_k}$ via a fixed deterministic convention (e.g., zero-padding when $D_k \ge d'$); this convention introduces no randomness.

Notations. Consider arbitrary $D \ge 1$ and write $[D] \triangleq \{0, 1, \ldots, D-1\}$. For $\mathbf{a}, \mathbf{b} \in \mathbb{R}^D$, the *circular convolution* $\mathbf{a} * \mathbf{b} \in \mathbb{R}^D$ is defined componentwise by

$$\Big(\mathbf{a} * \mathbf{b}\Big)[t] \;\triangleq\; \sum_{s \in [D]} \mathbf{a}[s] \cdot \mathbf{b}\Big[(t-s) \bmod D\Big], \qquad t \in [D].$$

For an integer $k \ge 1$ and $\mathbf{x} \in \mathbb{R}^D$, the k-fold circular convolution power is

$$\mathbf{x}^{(*k)} \;\triangleq\; \underbrace{\mathbf{x} * \mathbf{x} * \cdots * \mathbf{x}}_{k \text{ times}} \;\in\; \mathbb{R}^D.$$

Implemented Stage II features and embeddings (random). The implemented Stage II enrichment uses TensorSketch. For each macro index $j \in [M]$ and degree $k \in \mathcal{K}$, the sketched feature is

$$\mathbf{z}_{j,k}^{\text{ts}} \triangleq \text{TS}_k(\widetilde{\mathbf{G}}_{j,:}; D_k) \in \mathbb{R}^{D_k}. \tag{68}$$

The degree-mixture feature and Stage II embedding are

$$\mathbf{v}_j^{\text{ts}} \triangleq \bigoplus_{k \in \mathcal{K}} \beta_k \, \mathbf{z}_{j,k}^{\text{ts}} \in \mathbb{R}^{D_{\text{tot}}}, \quad \mathbf{Y}_{\text{enh}}[j,:] = \mathbf{W}_{\text{out}} \mathbf{v}_j^{\text{ts}} \in \mathbb{R}^d, \tag{69}$$

where D_{tot}, and $\mathbf{Y}_{\text{enh}}$ and $\mathbf{W}_{\text{out}}$ are defined in (9) and (10), respectively. In this construction, all randomness enters exclusively through the TensorSketch map TS_k in (68).

Deterministic Stage II reference (no randomness). We introduce a deterministic reference that keeps the interface $(\mathcal{K}, \{\beta_k\}, \{D_k\}_{k\in\mathcal{K}}, \mathbf{W}_{\text{out}})$ unchanged and replaces TensorSketch by exact k-fold circular convolution on $[D_k]$:

$$\mathbf{z}^{\text{det}}_{j,k} \triangleq \underbrace{\widetilde{\mathbf{G}}_{j,:} * \widetilde{\mathbf{G}}_{j,:} * \cdots * \widetilde{\mathbf{G}}_{j,:}}_{k \text{ times}} \in \mathbb{R}^{D_k}. \tag{70}$$

Equivalently, $\mathbf{z}^{\text{det}}_{j,k} = \text{IFFT}\big(\text{FFT}(\widetilde{\mathbf{G}}_{j,:})^{\odot k}\big)$ on length D_k. The associated degree-mixture feature is

$$\mathbf{v}^{\text{det}}_j \triangleq \bigoplus_{k\in\mathcal{K}} (\beta_k\, \mathbf{z}^{\text{det}}_{j,k}) \in \mathbb{R}^{D_{\text{tot}}}, \tag{71}$$

and the deterministic Stage II embedding is

$$\mathbf{Y}^{\text{det}}_{\text{enh}}[j,:] \triangleq \mathbf{W}_{\text{out}} \mathbf{v}^{\text{det}}_j \in \mathbb{R}^d. \tag{72}$$

All operations in (70)–(72) are deterministic.

Where randomness lives in the full block. Downstream of Stage II, the computation is deterministic and shared by the implemented and reference embeddings. At Stage II, the mixer is a deterministic map, as shown in in (11); Stage III applies the projections to $\mathbf{K}_g$ and $\mathbf{V}_g$ and then the exact attention readout in (12). Applying the same mixer / Stage III map to $\mathbf{Y}^{\text{det}}_{\text{enh}}$ yields the reference output

$$\mathbf{Y}^{\text{det}} \triangleq \mathcal{A}_{\mathbf{Q}}\big(\mathbf{Y}^{\text{det}}_{\text{enh}}\big),$$

where $\mathcal{A}_{\mathbf{Q}}$ denotes the deterministic composition of the mixer, projections, and attention readout. Consequently, *all* algorithmic randomness is localized to Stage II through (68). Any Stage II approximation bound can therefore be propagated end-to-end using deterministic stability of downstream maps.

When using $\mathbf{z}^{\text{det}}$ can hurt performance. Replacing the randomized Stage II features by the deterministic reference $\mathbf{z}^{\text{det}}$ removes sketch variance, but it can degrade test accuracy under standard training budgets for three complementary reasons:

- **Removing a beneficial noise source.** Stage II sketching injects structured, input-dependent noise into the representation. Such noise often acts as implicit regularization—analogous to dropout / noise-injection—by discouraging brittle feature co-adaptation and reducing overfitting. Consequently, making Stage II fully deterministic can increase the generalization gap when data or training time is limited (Srivastava et al., 2014).
- **Kernel-faithfulness vs. a fixed transform.** The deterministic map $\mathbf{z}^{\text{det}}$ is a fixed convolutional feature transform in $\mathbb{R}^{D_k}$ and therefore imposes a specific inductive bias. TensorSketch, in contrast, is explicitly constructed so that sketch-space inner products approximate the corresponding degree-k polynomial kernel inner products (in expectation, with concentration controlled by D_k), providing a principled surrogate for higher-order interactions (Pham & Pagh, 2013; Schölkopf & Smola, 2002). When higher-order interaction structure matters, this kernel-faithfulness can translate into better generalization than a fixed deterministic transform at the same D_k.
- **Fixed wall-clock budgets.** Our comparisons are ultimately wall-clock constrained. Even if $\mathbf{z}^{\text{det}}$ is FFT-computable, its end-to-end runtime can be less favorable on accelerators due to batching, padding, and kernel-fusion effects. Under a fixed training-time budget, slower per-step throughput means fewer optimizer updates (or smaller M / D_k), which can reduce final accuracy.

G.1. Stage II error certificate: TensorSketch vs. deterministic convolution

We bound the Stage II discrepancy between the implemented enriched embeddings $\mathbf{Y}_{\text{enh}}$ and the deterministic reference $\mathbf{Y}^{\text{det}}_{\text{enh}}$. The key idea is simple but powerful: we upper bound $\|\mathbf{z}^{\text{ts}}_{j,k} - \mathbf{z}^{\text{det}}_{j,k}\|_2$ using *separate* bounds on $\|\mathbf{z}^{\text{ts}}_{j,k}\|_2$ (high probability) and $\|\mathbf{z}^{\text{det}}_{j,k}\|_2$ (deterministic), then combine them via the triangle inequality.

Lemma G.1 (Norm growth under repeated circular convolution)**.** *Fix $D \geq 1$ and let $*$ be circular convolution on $\mathbb{R}^D$. For any $\mathbf{x} \in \mathbb{R}^D$ and any integer $k \geq 1$,*

$$\|\mathbf{x}^{(*k)}\|_2 \;\leq\; D^{\frac{k-1}{2}} \cdot \|\mathbf{x}\|_2^k. \tag{73}$$

Proof. We prove (73) in three explicit steps.

Step 1: A Young-type inequality for circular convolution. We prove that for any $\mathbf{a}, \mathbf{b} \in \mathbb{R}^D$,

$$\|\mathbf{a} * \mathbf{b}\|_2 \;\le\; \|\mathbf{a}\|_1 \cdot \|\mathbf{b}\|_2. \tag{74}$$

Step 1.1: Start from the coordinate definition. For each $t \in [D]$, circular convolution is

$$(\mathbf{a} * \mathbf{b})[t] = \sum_{s \in [D]} \mathbf{a}[s]\, \mathbf{b}\big[(t-s) \bmod D\big].$$

Step 1.2: Take absolute values and apply triangle inequality.

$$\begin{aligned}\big|(\mathbf{a} * \mathbf{b})[t]\big| &= \left| \sum_{s \in [D]} \mathbf{a}[s] \cdot \mathbf{b}\big[(t-s) \bmod D\big] \right| \\ &\le \sum_{s \in [D]} |\mathbf{a}[s]| \cdot \big|\mathbf{b}\big[(t-s) \bmod D\big]\big|.\end{aligned}$$

Step 1.3: Apply Cauchy–Schwarz to the sum over s. Let $x_s \triangleq |\mathbf{a}[s]|$ and $y_s \triangleq \big|\mathbf{b}\big[(t-s) \bmod D\big]\big|$. Then $\sum_s x_s y_s \le \|x\|_2 \|y\|_2$, so

$$\big|(\mathbf{a} * \mathbf{b})[t]\big| \le \left(\sum_{s \in [D]} |\mathbf{a}[s]|^2 \right)^{1/2} \left(\sum_{s \in [D]} \big|\mathbf{b}\big[(t-s) \bmod D\big]\big|^2 \right)^{1/2}.$$

Step 1.4: Upper bound $\|\mathbf{a}\|_2$ by $\|\mathbf{a}\|_1$. Since $|\mathbf{a}[s]|^2 \le |\mathbf{a}[s]| \cdot \|\mathbf{a}\|_1$ and summing over s gives $\|\mathbf{a}\|_2^2 \le \|\mathbf{a}\|_1^2$, we have $\|\mathbf{a}\|_2 \le \|\mathbf{a}\|_1$. Therefore,

$$\big|(\mathbf{a} * \mathbf{b})[t]\big| \le \|\mathbf{a}\|_1 \left(\sum_{s \in [D]} \big|\mathbf{b}\big[(t-s) \bmod D\big]\big|^2 \right)^{1/2}.$$

Step 1.5: Use the permutation property of circular shifts. The map $s \mapsto (t-s) \bmod D$ is a permutation of $[D]$, hence

$$\sum_{s \in [D]} \big|\mathbf{b}\big[(t-s) \bmod D\big]\big|^2 = \sum_{u \in [D]} |\mathbf{b}[u]|^2 = \|\mathbf{b}\|_2^2.$$

Thus, for every t,

$$\big|(\mathbf{a} * \mathbf{b})[t]\big| \le \|\mathbf{a}\|_1 \|\mathbf{b}\|_2.$$

Step 1.6: Convert the bound to an ℓ_2 inequality (avoid an extra factor of D). A direct "square-and-sum" using the uniform pointwise bound from Step 1.5 would introduce an unnecessary factor of D. Instead, we bound the ℓ_2 norm directly by rewriting convolution as a weighted sum of circular shifts and applying the triangle inequality.

Define the circular shift $\mathrm{shift}_s(\mathbf{b}) \in \mathbb{R}^D$ by

$$\big(\mathrm{shift}_s(\mathbf{b})\big)[t] \triangleq \mathbf{b}\big[(t-s) \bmod D\big].$$

Then, by the definition of circular convolution, for every $t \in [D]$,

$$(\mathbf{a} * \mathbf{b})[t] = \sum_{s \in [D]} \mathbf{a}[s] \cdot \big(\mathrm{shift}_s(\mathbf{b})\big)[t],$$

and hence (as vectors)

$$\mathbf{a} * \mathbf{b} = \sum_{s \in [D]} \mathbf{a}[s]\, \mathrm{shift}_s(\mathbf{b}).$$

Taking ℓ_2 norms and using $\|x+y\|_2 \le \|x\|_2 + \|y\|_2$ repeatedly,

$$\|\mathbf{a} * \mathbf{b}\|_2 \le \sum_{s\in[D]} \big\|\mathbf{a}[s]\ \mathrm{shift}_s(\mathbf{b})\big\|_2.$$

Now use $\|\alpha x\|_2 = |\alpha|\,\|x\|_2$ and the fact that a circular shift is a permutation of coordinates, so it preserves the ℓ_2 norm:

$$\big\|\mathbf{a}[s]\ \mathrm{shift}_s(\mathbf{b})\big\|_2 = |\mathbf{a}[s]|\ \|\mathrm{shift}_s(\mathbf{b})\|_2 = |\mathbf{a}[s]|\ \|\mathbf{b}\|_2.$$

Therefore,

$$\|\mathbf{a} * \mathbf{b}\|_2 \le \sum_{s\in[D]} |\mathbf{a}[s]|\ \|\mathbf{b}\|_2 = \|\mathbf{a}\|_1\ \|\mathbf{b}\|_2,$$

which proves (74).

Step 2: Relate ℓ_1 and ℓ_2 on $\mathbb{R}^D$. By Cauchy–Schwarz,

$$\|\mathbf{a}\|_1 \;=\; \sum_{t\in[D]} |\mathbf{a}[t]| \;\le\; \sqrt{D}\left(\sum_{t\in[D]} |\mathbf{a}[t]|^2\right)^{1/2} \;=\; \sqrt{D}\,\|\mathbf{a}\|_2. \tag{75}$$

Step 3: Iterate the one-step bound. Define $\mathbf{x}^{(*1)} \triangleq \mathbf{x}$ and $\mathbf{x}^{(*(m+1))} \triangleq \mathbf{x}^{(*m)} * \mathbf{x}$. Combine (74) and (75) with $\mathbf{a} = \mathbf{x}^{(*m)}$ and $\mathbf{b} = \mathbf{x}$:

$$\|\mathbf{x}^{(*(m+1))}\|_2 = \|\mathbf{x}^{(*m)} * \mathbf{x}\|_2 \le \|\mathbf{x}^{(*m)}\|_1 \|\mathbf{x}\|_2 \le \sqrt{D}\,\|\mathbf{x}^{(*m)}\|_2\,\|\mathbf{x}\|_2.$$

Apply this inequality for $m = 1, 2, \ldots, k-1$:

$$\|\mathbf{x}^{(*k)}\|_2 \le (\sqrt{D})^{k-1}\|\mathbf{x}\|_2^k = D^{\frac{k-1}{2}}\|\mathbf{x}\|_2^k,$$

which is (73). □

Corollary G.2 (Uniform deterministic bound for $\|\mathbf{z}_{j,k}^{\mathrm{det}}\|_2$)**.** *Assume the Stage II norm control in (7), i.e., $\|\widetilde{\mathbf{G}}_{j,:}\|_2 \le 1/\tau_{\mathrm{g}}$ for all $j \in [M]$. Then for all $j \in [M]$ and $k \in \mathcal{K}$,*

$$\|\mathbf{z}_{j,k}^{\mathrm{det}}\|_2 \;\le\; D_k^{\frac{k-1}{2}} \cdot \tau_{\mathrm{g}}^{-k}. \tag{76}$$

Proof. Fix $j \in [M]$ and $k \in \mathcal{K}$. By (70), $\mathbf{z}_{j,k}^{\mathrm{det}} = \widetilde{\mathbf{G}}_{j,:}^{(*k)}$ in $\mathbb{R}^{D_k}$. Apply Lemma G.1 with $D = D_k$ and $\mathbf{x} = \widetilde{\mathbf{G}}_{j,:}$:

$$\|\mathbf{z}_{j,k}^{\mathrm{det}}\|_2 = \|\widetilde{\mathbf{G}}_{j,:}^{(*k)}\|_2 \le D_k^{\frac{k-1}{2}}\|\widetilde{\mathbf{G}}_{j,:}\|_2^k.$$

Finally use $\|\widetilde{\mathbf{G}}_{j,:}\|_2 \le 1/\tau_{\mathrm{g}}$ from (7) to obtain (76). □

Lemma G.3 (Uniform high-probability norm bound for TensorSketch features)**.** *Fix $\eta \in (0,1)$ and $\delta \in (0,1)$. Assume the Stage II norm control in (7) and Assumption C.2. Choose per-degree budgets $\{\delta_k\}_{k\in\mathcal{K}}$ such that $\sum_{k\in\mathcal{K}} \delta_k \le \delta$. If*

$$D_k \;\ge\; \frac{2M}{\eta^2 \cdot \delta_k}, \qquad \forall k \in \mathcal{K}, \tag{77}$$

then with probability at least $1-\delta$, simultaneously for all $j \in [M]$ and $k \in \mathcal{K}$,

$$\|\mathbf{z}_{j,k}^{\mathrm{ts}}\|_2 \;\le\; \sqrt{1+\eta}\cdot\|\widetilde{\mathbf{G}}_{j,:}\|_2^k \;\le\; \sqrt{1+\eta}\cdot\tau_{\mathrm{g}}^{-k}. \tag{78}$$

Proof. Fix any pair (j,k) and define $\mathbf{x} \triangleq \widetilde{\mathbf{G}}_{j,:}$ and $D \triangleq D_k$. Let

$$Z \;\triangleq\; \|\mathrm{TS}_k(\mathbf{x}; D)\|_2^2.$$

Under Assumption C.2, Proposition C.4 gives the moment bounds

$$\mathbb{E}[Z] = \|\mathbf{x}\|_2^{2k}, \qquad \mathrm{Var}(Z) \le \frac{2}{D}\|\mathbf{x}\|_2^{4k}.$$

Apply Chebyshev's inequality (Tchebychef, 1867) with threshold $\eta\|\mathbf{x}\|_2^{2k}$:

$$\begin{aligned}\Pr\big(Z > (1+\eta)\|\mathbf{x}\|_2^{2k}\big) &= \Pr\big(Z - \mathbb{E}[Z] > \eta\|\mathbf{x}\|_2^{2k}\big)\\ &\le \frac{\mathrm{Var}(Z)}{\eta^2\|\mathbf{x}\|_2^{4k}}\\ &\le \frac{2}{D\eta^2}.\end{aligned}$$

Now allocate failure budget across the M macro indices by setting $\delta'_{j,k} \triangleq \delta_k/M$. Under (77), we have $D_k \ge 2/(\eta^2\delta'_{j,k})$, hence

$$\Pr\big[Z > (1+\eta)\|\mathbf{x}\|_2^{2k}\big] \le \delta'_{j,k}.$$

Equivalently, with probability at least $1-\delta'_{j,k}$,

$$\|\mathrm{TS}_k(\mathbf{x}; D_k)\|_2 \le \sqrt{1+\eta}\cdot\|\mathbf{x}\|_2^k.$$

A union bound over $j \in [M]$ yields failure probability at most δ_k for each fixed k. A second union bound over $k \in \mathcal{K}$ yields total failure probability at most $\sum_{k\in\mathcal{K}}\delta_k \le \delta$, proving the first inequality in (78). Finally, (7) implies $\|\mathbf{x}\|_2 \le 1/\tau_{\mathrm{g}}$, giving the second inequality. □

G.2. A complete Stage II feature discrepancy bound (no difference computation)

Theorem G.4 (Uniform high-probability bound for $\|\mathbf{z}^{\mathrm{ts}}_{j,k} - \mathbf{z}^{\mathrm{det}}_{j,k}\|_2$)**.** *Fix $\eta \in (0,1)$ and $\delta \in (0,1)$. Recall the normalized macro features $\widetilde{\mathbf{G}}$ in (7), and assume Assumption C.2. For each $k \in \mathcal{K}$, let $\mathbf{z}^{\mathrm{ts}}_{j,k} \in \mathbb{R}^{D_k}$ denote the degree-k TensorSketch feature (Definition C.1) and let $\mathbf{z}^{\mathrm{det}}_{j,k} \in \mathbb{R}^{D_k}$ denote the degree-k deterministic convolutional feature (70). Choose per-degree failure budgets $\{\delta_k\}_{k\in\mathcal{K}}$ such that $\sum_{k\in\mathcal{K}}\delta_k \le \delta$, and choose sketch sizes $\{D_k\}_{k\in\mathcal{K}}$ satisfying (77). Then, with probability at least $1-\delta$, simultaneously for all $j \in [M]$ and $k \in \mathcal{K}$,*

$$\big\|\mathbf{z}^{\mathrm{ts}}_{j,k} - \mathbf{z}^{\mathrm{det}}_{j,k}\big\|_2 \;\le\; \left(\sqrt{1+\eta} + D_k^{\frac{k-1}{2}}\right)\cdot\tau_{\mathrm{g}}^{-k}. \tag{79}$$

In particular, for any target tolerance $\epsilon_{\mathrm{feat}} > 0$, the deterministic choice

$$\tau_{\mathrm{g}} \;\ge\; \max_{k\in\mathcal{K}}\left(\left(\frac{\sqrt{1+\eta} + (D_k)^{(k-1)/2}}{\epsilon_{\mathrm{feat}}}\right)^{1/k}\right) \tag{80}$$

implies $\big\|\mathbf{z}^{\mathrm{ts}}_{j,k} - \mathbf{z}^{\mathrm{det}}_{j,k}\big\|_2 \le \epsilon_{\mathrm{feat}}$ uniformly over all (j,k) on the same probability-$1-\delta$ event.

Proof. We prove the bound by conditioning on the uniform TensorSketch norm event and then using the triangle inequality.

Step 1: define the high-probability TensorSketch norm event. Let

$$\mathcal{E}_{\mathrm{ts}} \triangleq \left\{\forall j \in [M],\ \forall k \in \mathcal{K}:\ \big\|\mathbf{z}^{\mathrm{ts}}_{j,k}\big\|_2 \le \sqrt{1+\eta}\cdot\tau_{\mathrm{g}}^{-k}\right\}.$$

By Lemma G.3 and (77), $\Pr[\mathcal{E}_{\mathrm{ts}}] \ge 1-\delta$.

Step 2: use the deterministic bound for the reference features. By Corollary G.2, deterministically for all $j \in [M]$ and $k \in \mathcal{K}$,

$$\big\|\mathbf{z}^{\mathrm{det}}_{j,k}\big\|_2 \le D_k^{\frac{k-1}{2}}\cdot\tau_{\mathrm{g}}^{-k}.$$

Step 3: combine via the triangle inequality. Fix any (j,k). On the event $\mathcal{E}_{\mathrm{ts}}$,

$$\begin{aligned}\left\|\mathbf{z}_{j,k}^{\mathrm{ts}}-\mathbf{z}_{j,k}^{\mathrm{det}}\right\|_2 &\le \left\|\mathbf{z}_{j,k}^{\mathrm{ts}}\right\|_2+\left\|\mathbf{z}_{j,k}^{\mathrm{det}}\right\|_2\\ &\le \sqrt{1+\eta}\cdot\tau_{\mathrm{g}}^{-k}+D_k^{\frac{k-1}{2}}\cdot\tau_{\mathrm{g}}^{-k}\\ &=\left(\sqrt{1+\eta}+D_k^{\frac{k-1}{2}}\right)\cdot\tau_{\mathrm{g}}^{-k},\end{aligned}$$

which is (79). Since this holds for every (j,k) on $\mathcal{E}_{\mathrm{ts}}$, it holds simultaneously with probability at least $1-\delta$.

Step 4: derive the sufficient choice of τ_{g}. Requiring the right-hand side of (79) to be at most ϵ_{feat} for all $k\in\mathcal{K}$ is equivalent to requiring, for each k,

$$\tau_{\mathrm{g}}\ge\left(\frac{\sqrt{1+\eta}+(D_k)^{(k-1)/2}}{\epsilon_{\mathrm{feat}}}\right)^{1/k}.$$

Taking the maximum over $k\in\mathcal{K}$ yields (80). □

Corollary G.5 (Stage II macro-embedding certificate)**.** *Under the conditions of Theorem G.4, with probability at least* $1-\delta$*,*

$$\left\|\mathbf{Y}_{\mathrm{enh}}-\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}\right\|_{2,\infty}\ \le\ \|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}}\cdot\left(\sum_{k\in\mathcal{K}}\beta_k^2\cdot\left(\left(\sqrt{1+\eta}+D_k^{\frac{k-1}{2}}\right)\cdot\tau_{\mathrm{g}}^{-k}\right)^2\right)^{1/2}. \tag{81}$$

Proof. We expand the objects and then bound them using basic norm properties.

Step 1: write the difference in terms of the degree-mixture vectors. Recall

$$\mathbf{Y}_{\mathrm{enh}}[j,:]=\mathbf{W}_{\mathrm{out}}\mathbf{v}_j^{\mathrm{ts}},\qquad \mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}[j,:]=\mathbf{W}_{\mathrm{out}}\mathbf{v}_j^{\mathrm{det}},$$

where

$$\mathbf{v}_j^{\mathrm{ts}}=\bigoplus_{k\in\mathcal{K}}(\beta_k\cdot\mathbf{z}_{j,k}^{\mathrm{ts}}),\qquad \mathbf{v}_j^{\mathrm{det}}=\bigoplus_{k\in\mathcal{K}}(\beta_k\cdot\mathbf{z}_{j,k}^{\mathrm{det}}).$$

Thus, for each j,

$$\mathbf{Y}_{\mathrm{enh}}[j,:]-\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}[j,:]=\mathbf{W}_{\mathrm{out}}\big(\mathbf{v}_j^{\mathrm{ts}}-\mathbf{v}_j^{\mathrm{det}}\big).$$

Step 2: apply the operator norm bound. By the definition of $\|\cdot\|_{\mathrm{op}}$,

$$\left\|\mathbf{Y}_{\mathrm{enh}}[j,:]-\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}[j,:]\right\|_2\le\|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}}\cdot\left\|\mathbf{v}_j^{\mathrm{ts}}-\mathbf{v}_j^{\mathrm{det}}\right\|_2.$$

Step 3: compute the norm of a block concatenation. Because $\mathbf{v}_j^{\mathrm{ts}}-\mathbf{v}_j^{\mathrm{det}}$ is a direct sum with disjoint coordinate blocks,

$$\left\|\mathbf{v}_j^{\mathrm{ts}}-\mathbf{v}_j^{\mathrm{det}}\right\|_2^2=\sum_{k\in\mathcal{K}}\beta_k^2\cdot\left\|\mathbf{z}_{j,k}^{\mathrm{ts}}-\mathbf{z}_{j,k}^{\mathrm{det}}\right\|_2^2.$$

Step 4: substitute the uniform feature discrepancy bound. On the probability-$1-\delta$ event of Theorem G.4, for all j and k,

$$\left\|\mathbf{z}_{j,k}^{\mathrm{ts}}-\mathbf{z}_{j,k}^{\mathrm{det}}\right\|_2\le\left(\sqrt{1+\eta}+D_k^{\frac{k-1}{2}}\right)\cdot\tau_{\mathrm{g}}^{-k}.$$

Substitute into Step 3, take square roots, and then take the maximum over $j\in[M]$ to obtain (98). □

H. Proof of Theorem 3.2: an end-to-end approximation error analysis and a forward-pass checkable sufficient condition

This section proves Theorem 3.2. Recall that Stage II is the *only* source of algorithmic randomness in PLASH. Stages I and III are deterministic given inputs and parameters. Therefore, once we control the Stage II approximation error, we can propagate it to the final output using deterministic stability of the post-processing pipeline.

Notation and norms. For a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ we write $\|\mathbf{A}\|_{2,\infty} \triangleq \max_{i\in[n]} \|\mathbf{A}_{i,:}\|_2$ and $\|\cdot\|_F$ for the Frobenius norm. We use $\|\cdot\|_{\mathrm{op}}$ for the spectral operator norm (Definition A.3).

H.1. A deterministic view of Mixer and Stage III

Deterministic post-processing map. Fix a forward pass and queries $\mathbf{Q}$. Define the deterministic map

$$\mathcal{T}_{\mathbf{Q}} : \mathbb{R}^{M\times d} \to \mathbb{R}^{N_q\times d_v}$$

as the composition of (i) the macro mixer as defined in Definiton A.7, (ii) linear projections to macro keys and values, and (iii) the exact softmax readout from queries to the compressed keys / values (cf. Section 2.2.3 and (11)–(12)):

$$\mathcal{T}_{\mathbf{Q}}(\mathbf{U}) \triangleq \mathrm{Atten}\Big(\mathbf{Q};\ \Big(\mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U})\Big)\cdot\mathbf{W}_K,\ \Big(\mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U})\Big)\cdot\mathbf{W}_V\Big).$$

Accordingly,

$$\mathbf{Y}_{\mathrm{PLASH}} \triangleq \mathcal{T}_{\mathbf{Q}}(\mathbf{Y}_{\mathrm{enh}}), \qquad \mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}} \triangleq \mathcal{T}_{\mathbf{Q}}(\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}),$$

where $\mathbf{Y}_{\mathrm{enh}}$ is the implemented Stage II embedding (69) and $\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}$ is the deterministic Stage II comparator (72).

Lemma H.1 (Deterministic stability of the Mixer and Stage III post-processing pipeline)**.** *Fix a forward pass and queries* $\mathbf{Q}$*. Let the segment*

$$\mathcal{S} \triangleq \Big\{\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}} + t\cdot\Big(\mathbf{Y}_{\mathrm{enh}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}\Big) :\ t\in[0,1]\Big\} \subseteq \mathbb{R}^{M\times d} \tag{82}$$

connect $\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}$ *and* $\mathbf{Y}_{\mathrm{enh}}$*. Recall that Theorem B.5 (with the practical upper bound from Theorem B.8) holds on* $\mathcal{S}$*, i.e., there exists a deterministic constant* $L_{\mathrm{mix}} > 0$ *such that for all* $\mathbf{U},\mathbf{U}' \in \mathcal{S}$*,*

$$\big\|\mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U}) - \mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U}')\big\|_{2,\infty} \le L_{\mathrm{mix}}\cdot\|\mathbf{U}-\mathbf{U}'\|_{2,\infty}. \tag{83}$$

Define

$$\begin{aligned} L_K &\triangleq \|\mathbf{W}_K\|_{\mathrm{op}},\\ L_V &\triangleq \|\mathbf{W}_V\|_{\mathrm{op}},\\ \Gamma_Q^{(M)} &\triangleq \frac{1}{\sqrt{d_k}}\cdot\|\mathbf{Q}\|_{2,\infty}, \end{aligned} \tag{84}$$

and let

$$\Gamma_V(\mathcal{S}) \triangleq \sup_{\mathbf{U}\in\mathcal{S}}\big\|\mathbf{V}_g(\mathbf{U})\big\|_{2,\infty}, \tag{85}$$

where $\mathbf{V}_g(\mathbf{U})$ *denotes the Stage III macro-value map*

$$\mathbf{V}_g(\mathbf{U}) \triangleq \Big(\mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U})\Big)\cdot\mathbf{W}_V.$$

Then for all $\mathbf{U},\mathbf{U}' \in \mathcal{S}$*,*

$$\big\|\mathcal{T}_{\mathbf{Q}}(\mathbf{U}) - \mathcal{T}_{\mathbf{Q}}(\mathbf{U}')\big\|_F \le \sqrt{N_q}\cdot L_{\mathrm{post}}(\mathcal{S})\cdot\|\mathbf{U}-\mathbf{U}'\|_{2,\infty}, \tag{86}$$

where

$$L_{\mathrm{post}}(\mathcal{S}) \triangleq L_{\mathrm{mix}}\cdot\Big(\Gamma_Q^{(M)}\cdot L_K\cdot\Gamma_V(\mathcal{S}) + L_V\Big). \tag{87}$$

In particular, taking $(\mathbf{U},\mathbf{U}') = \big(\mathbf{Y}_{\mathrm{enh}}, \mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}\big)$ *yields*

$$\big\|\mathbf{Y}_{\mathrm{PLASH}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}}\big\|_F \le \sqrt{N_q}\cdot L_{\mathrm{post}}(\mathcal{S})\cdot\big\|\mathbf{Y}_{\mathrm{enh}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}\big\|_{2,\infty}. \tag{88}$$

Proof. Fix $\mathbf{U}, \mathbf{U}' \in \mathcal{S}$ and abbreviate

$$\mathbf{Z} \triangleq \mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U}), \qquad \mathbf{Z}' \triangleq \mathrm{Mixer}_{L_{\mathrm{mix}}}(\mathbf{U}').$$

Step 1: stability of the mixer. By the local Lipschitz property (83),

$$\|\mathbf{Z} - \mathbf{Z}'\|_{2,\infty} \le L_{\mathrm{mix}} \cdot \|\mathbf{U} - \mathbf{U}'\|_{2,\infty}. \tag{89}$$

Step 2: stability of linear projections. Define the induced compressed keys and values

$$\mathbf{K}_g(\mathbf{U}) \triangleq \mathbf{Z} \cdot \mathbf{W}_K, \qquad \mathbf{V}_g(\mathbf{U}) \triangleq \mathbf{Z} \cdot \mathbf{W}_V, \tag{90}$$

and analogously $\mathbf{K}_g(\mathbf{U}')$, $\mathbf{V}_g(\mathbf{U}')$ using $\mathbf{Z}'$. By Lemma B.1 (right-multiplication stability in $\|\cdot\|_{2,\infty}$),

$$\big\|\mathbf{K}_g(\mathbf{U}) - \mathbf{K}_g(\mathbf{U}')\big\|_{2,\infty} \le L_K \cdot \|\mathbf{Z} - \mathbf{Z}'\|_{2,\infty}, \tag{91}$$

$$\big\|\mathbf{V}_g(\mathbf{U}) - \mathbf{V}_g(\mathbf{U}')\big\|_{2,\infty} \le L_V \cdot \|\mathbf{Z} - \mathbf{Z}'\|_{2,\infty}. \tag{92}$$

Step 3: stability of attention with respect to key / value perturbations. Apply Lemma E.1 with $N = M$ and the identification

$$(\mathbf{K}, \mathbf{V}) \triangleq \big(\mathbf{K}_g(\mathbf{U}), \mathbf{V}_g(\mathbf{U})\big), \qquad (\mathbf{K}', \mathbf{V}') \triangleq \big(\mathbf{K}_g(\mathbf{U}'), \mathbf{V}_g(\mathbf{U}')\big).$$

This gives

$$\big\|\mathcal{T}_{\mathbf{Q}}(\mathbf{U}) - \mathcal{T}_{\mathbf{Q}}(\mathbf{U}')\big\|_F \le \sqrt{N_q} \cdot \left(\Gamma_Q^{(M)} \cdot \big\|\mathbf{K}_g(\mathbf{U}) - \mathbf{K}_g(\mathbf{U}')\big\|_{2,\infty} \cdot \big\|\mathbf{V}_g(\mathbf{U})\big\|_{2,\infty} + \big\|\mathbf{V}_g(\mathbf{U}) - \mathbf{V}_g(\mathbf{U}')\big\|_{2,\infty}\right). \tag{93}$$

Step 4: uniform control of values on the segment. Because $\mathbf{U} \in \mathcal{S}$, the definition (85) implies

$$\big\|\mathbf{V}_g(\mathbf{U})\big\|_{2,\infty} \le \Gamma_V(\mathcal{S}). \tag{94}$$

Step 5: collect bounds and simplify. Substitute (94), (91), and (92) into (93) to obtain

$$\big\|\mathcal{T}_{\mathbf{Q}}(\mathbf{U}) - \mathcal{T}_{\mathbf{Q}}(\mathbf{U}')\big\|_F \le \sqrt{N_q} \cdot \Big(\Gamma_Q^{(M)} \cdot L_K \cdot \Gamma_V(\mathcal{S}) + L_V\Big) \cdot \|\mathbf{Z} - \mathbf{Z}'\|_{2,\infty}.$$

Finally, apply (89) to conclude

$$\big\|\mathcal{T}_{\mathbf{Q}}(\mathbf{U}) - \mathcal{T}_{\mathbf{Q}}(\mathbf{U}')\big\|_F \le \sqrt{N_q} \cdot L_{\mathrm{mix}} \cdot \Big(\Gamma_Q^{(M)} \cdot L_K \cdot \Gamma_V(\mathcal{S}) + L_V\Big) \cdot \|\mathbf{U} - \mathbf{U}'\|_{2,\infty},$$

which is (86) with $L_{\mathrm{post}}(\mathcal{S})$ defined in (87). The specialization (88) follows by taking $(\mathbf{U}, \mathbf{U}') = \big(\mathbf{Y}_{\mathrm{enh}}, \mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}\big)$. □

H.2. From Stage II discrepancy to an end-to-end bound

Theorem H.2 ($\mathcal{K}$-PLASH: controlled approximation to standard attention)**.** *Fix $\eta \in (0, 1)$ and $\delta \in (0, 1)$. Recall the normalized macro features $\widetilde{\mathbf{G}}$ in (7), and assume Assumption C.2. Let $\mathcal{K} \subset \mathbb{Z}_{\ge 1}$ be finite, with mixture weights $\{\beta_k\}_{k\in\mathcal{K}}$ and sketch lengths $\{D_k\}_{k\in\mathcal{K}}$.*

Recall

$$\mathbf{Y}_{\mathrm{PLASH}} = \mathcal{T}_{\mathbf{Q}}\Big(\mathbf{Y}_{\mathrm{enh}}\Big), \qquad \mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}} = \mathcal{T}_{\mathbf{Q}}\Big(\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}\Big),$$

and assume the deterministic constants in Lemma H.1 are well-defined on the segment $\mathcal{S}$.

Choose per-degree failure budgets $\{\delta_k\}_{k\in\mathcal{K}}$ with $\sum_{k\in\mathcal{K}} \delta_k \le \delta$. If, for every $k \in \mathcal{K}$,

$$D_k \ge \frac{2M}{\eta^2 \cdot \delta_k}, \tag{95}$$

then with probability at least $1-\delta$*,*

$$\left\|\mathbf{Y}_{\mathrm{PLASH}}-\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}}\right\|_F \;\le\; \sqrt{N_q}\cdot L_{\mathrm{post}}(\mathcal{S})\cdot\|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}}\cdot\left(\sum_{k\in\mathcal{K}}\beta_k^2\Big(\big(\sqrt{1+\eta}+1\big)^2\cdot\tau_{\mathrm{g}}^{-2k}\Big)\right)^{1/2}. \tag{96}$$

Moreover, with $\mathbf{Y}_{\mathrm{soft}}$ *defined in* (1) *and* $\mathbf{Y}_{\mathrm{q}}$ *defined in* (61)*, we have the deterministic decomposition*

$$\left\|\mathbf{Y}_{\mathrm{soft}}-\mathbf{Y}_{\mathrm{PLASH}}\right\|_F \;\le\; \left\|\mathbf{Y}_{\mathrm{soft}}-\mathbf{Y}_{\mathrm{q}}\right\|_F \;+\; \left\|\mathbf{Y}_{\mathrm{q}}-\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}}\right\|_F \;+\; \left\|\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}}-\mathbf{Y}_{\mathrm{PLASH}}\right\|_F, \tag{97}$$

where the first term is deterministically bounded by (62) *and the third term is bounded with probability at least* $1-\delta$ *by* (96)*.*

Proof. We prove the two claims in order.

Step 1: Stage II embedding discrepancy (probability $1-\delta$). Apply Corollary G.5 degree-wise and take a union bound over $k\in\mathcal{K}$ with budgets $\{\delta_k\}_{k\in\mathcal{K}}$. Under (95), with probability at least $1-\delta$ we have simultaneously for all $k\in\mathcal{K}$ the Stage II feature discrepancy needed by the corollary. Therefore, on this event,

$$\left\|\mathbf{Y}_{\mathrm{enh}}-\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}\right\|_{2,\infty} \;\le\; \|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}}\cdot\left(\sum_{k\in\mathcal{K}}\beta_k^2\cdot\big(\sqrt{1+\eta}+1\big)^2\cdot\tau_{\mathrm{g}}^{-2k}\right)^{1/2}. \tag{98}$$

Step 2: propagate Stage II discrepancy through deterministic post-processing. On the same event, apply Lemma H.1 with $(\mathbf{U},\mathbf{U}')=(\mathbf{Y}_{\mathrm{enh}},\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}})$:

$$\begin{aligned}\left\|\mathbf{Y}_{\mathrm{PLASH}}-\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}}\right\|_F &= \left\|\mathcal{T}_{\mathbf{Q}}(\mathbf{Y}_{\mathrm{enh}})-\mathcal{T}_{\mathbf{Q}}(\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}})\right\|_F\\ &\le \sqrt{N_q}\cdot L_{\mathrm{post}}(\mathcal{S})\cdot\left\|\mathbf{Y}_{\mathrm{enh}}-\mathbf{Y}_{\mathrm{enh}}^{\mathrm{det}}\right\|_{2,\infty}.\end{aligned}$$

Substitute (98) to obtain (96).

Step 3: deterministic decomposition against standard attention. By the identity

$$\mathbf{Y}_{\mathrm{soft}}-\mathbf{Y}_{\mathrm{PLASH}}=(\mathbf{Y}_{\mathrm{soft}}-\mathbf{Y}_{\mathrm{q}})+(\mathbf{Y}_{\mathrm{q}}-\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}})+(\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}}-\mathbf{Y}_{\mathrm{PLASH}})$$

and the triangle inequality, we obtain (97). The stated bounds on the first and third terms are exactly the referenced results. □

H.3. Well-definedness of the post-processing constant

On the assumption that the deterministic constants in Lemma H.1 are well-defined on $\mathcal{S}$. Lemma H.1 requires (i) a valid local Lipschitz bound L_{mix} on the segment $\mathcal{S}$ and (ii) finiteness of $\Gamma_V(\mathcal{S})$. Both are guaranteed under the same safeguards used in standard Transformer implementations:

1. **Stabilized normalization.** Any LayerNorm / Norm in the mixer uses a stabilizer (e.g., $\varepsilon_{\mathrm{ln}}>0$), so denominators are bounded away from 0 on $\mathcal{S}$.
2. **Finite weights.** Linear maps have finite operator norms, hence L_K and L_V are finite.
3. **Compactness.** The set $\mathcal{S}$ is a closed line segment in $\mathbb{R}^{M\times d}$ and is therefore compact.

Under these conditions, the deterministic composition that defines $\mathrm{Mixer}_{L_{\mathrm{mix}}}$ is continuous on $\mathcal{S}$ and admits a finite local Lipschitz modulus on $\mathcal{S}$ (Theorem B.5 with Theorem B.8); likewise, $\mathbf{V}_g(\mathbf{U})$ defined in (90) is continuous in $\mathbf{U}$, so $\Gamma_V(\mathcal{S})<\infty$ by the extreme value theorem. Therefore $L_{\mathrm{post}}(\mathcal{S})$ in (87) is finite and well-defined.

Theorem H.3 (Forward-pass checkable $(\epsilon_{\mathrm{out}},\delta)$ guarantee for $\mathcal{K}$-PLASH)**.** *Fix* $\epsilon_{\mathrm{out}}>0$ *and* $\delta\in(0,1)$*. Let* $\mathbf{Y}_{\mathrm{soft}}=\mathrm{Atten}(\mathbf{Q};\mathbf{K},\mathbf{V})$ *and let* $\mathbf{Y}_{\mathrm{PLASH}}$ *be the* PLASH *output on the same* $(\mathbf{Q},\mathbf{K},\mathbf{V})$ *with degree set* $\mathcal{K}$*. Write*

$$k_{\min}\triangleq\min\mathcal{K}.$$

Fix $\eta\in(0,1)$ *and choose per-degree budgets* $\{\delta_k\}_{k\in\mathcal{K}}$ *with* $\sum_{k\in\mathcal{K}}\delta_k\le\delta$*. If, for every* $k\in\mathcal{K}$*,*

$$D_k \;\ge\; \frac{2M}{\eta^2\cdot\delta_k}, \tag{99}$$

then with probability at least $1-\delta$, the Stage II event (98) holds.

Define the following forward-pass computable quantities.

(I) Deterministic Stage I term. *Compute the hard-routing comparator $(\mathbf{K}^{\mathrm{q}}, \mathbf{V}^{\mathrm{q}})$ and radii $\rho_K(M), \rho_V(M)$ as in (60), and set*

$$\epsilon_{\mathrm{I}} \triangleq \sqrt{N_q} \cdot \Big(\Gamma_Q^{(N_k)} \cdot \rho_K(M) \cdot V_{\max} + \rho_V(M) \Big), \tag{100}$$

where $\Gamma_Q^{(N_k)}$ and $V_{\max}$ are as in Lemma F.2. Then $\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{q}}\|_F \le \epsilon_{\mathrm{I}}$ deterministically.

(det) Deterministic bias term. *Compute $\mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}} = \mathcal{T}_{\mathbf{Q}}(\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}})$ and define*

$$\epsilon_{\mathrm{det}} \triangleq \big\|\mathbf{Y}_{\mathrm{q}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}}\big\|_F. \tag{101}$$

(II) Random Stage II term propagated through deterministic post-processing. *Compute $L_{\mathrm{post}}(\mathcal{S})$ via (87) using any valid local Lipschitz upper bound for L_{mix} on the segment $\mathcal{S}$, and define*

$$\epsilon_{\mathrm{II}} \triangleq \epsilon_{\mathrm{out}} - \epsilon_{\mathrm{I}} - \epsilon_{\mathrm{det}}. \tag{102}$$

Sufficient condition and conclusion. *Assume $\epsilon_{\mathrm{II}} > 0$. If τ_{g} satisfies*

$$\tau_{\mathrm{g}} \ge \epsilon_{\mathrm{II}}^{-1/k_{\min}} \cdot \left(\sqrt{N_q} \cdot L_{\mathrm{post}}(\mathcal{S}) \cdot \|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}} \cdot \Big(\sum_{k \in \mathcal{K}} |\beta_k|^2 \Big)^{1/2} \cdot \Big(\sqrt{1+\eta} + 1 \Big) \right)^{1/k_{\min}}, \tag{103}$$

where $k_{\min} \triangleq \min\{k \mid k \in \mathcal{K}\}$, then, under (99), we have with probability at least $1-\delta$,

$$\big\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{PLASH}}\big\|_F \le \epsilon_{\mathrm{out}}. \tag{104}$$

Proof. We proceed by bounding the three terms in the deterministic decomposition (97).

Step 0: deterministic decomposition. By (97) and the triangle inequality,

$$\big\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{PLASH}}\big\|_F \le \underbrace{\big\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{q}}\big\|_F}_{\mathrm{(I)}} + \underbrace{\big\|\mathbf{Y}_{\mathrm{q}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}}\big\|_F}_{\mathrm{(det)}} + \underbrace{\big\|\mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}} - \mathbf{Y}_{\mathrm{PLASH}}\big\|_F}_{\mathrm{(II)}}. \tag{105}$$

Step 1: bound term (I). Lemma F.2 together with the definition (100) yields

$$\big\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{q}}\big\|_F \le \epsilon_{\mathrm{I}} \qquad \text{deterministically.}$$

Step 2: term (det) is exactly ϵ_{det}. By definition (101),

$$\big\|\mathbf{Y}_{\mathrm{q}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}}\big\|_F = \epsilon_{\mathrm{det}} \qquad \text{deterministically.}$$

Step 3: bound term (II) using Stage II control and post-processing stability. First apply Lemma H.1 with $(\mathbf{U}, \mathbf{U}') = (\mathbf{Y}_{\mathrm{enh}}, \mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}})$:

$$\big\|\mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}} - \mathbf{Y}_{\mathrm{PLASH}}\big\|_F \le \sqrt{N_q} \cdot L_{\mathrm{post}}(\mathcal{S}) \cdot \big\|\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}} - \mathbf{Y}_{\mathrm{enh}}\big\|_{2,\infty}. \tag{106}$$

Next, on the probability-$(1-\delta)$ event implied by (99), the Stage II bound (98) holds. Therefore, on that same event,

$$\begin{aligned} \big\|\mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}} - \mathbf{Y}_{\mathrm{enh}}\big\|_{2,\infty} &\le \|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}} \cdot \left(\sum_{k \in \mathcal{K}} |\beta_k|^2 \cdot \big(\sqrt{1+\eta} + 1 \big)^2 \cdot \tau_{\mathrm{g}}^{-2k} \right)^{1/2} \\ &\le \|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}} \cdot \left(\sum_{k \in \mathcal{K}} |\beta_k|^2 \right)^{1/2} \cdot \big(\sqrt{1+\eta} + 1 \big) \cdot \tau_{\mathrm{g}}^{-k_{\min}}. \end{aligned} \tag{107}$$

Substitute (107) into (106). On the same probability-$(1-\delta)$ event,

$$\left\|\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}} - \mathbf{Y}_{\mathrm{PLASH}}\right\|_F \;\le\; \sqrt{N_q}\cdot L_{\mathrm{post}}(\mathcal{S})\cdot\|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}}\cdot\left(\sum_{k\in\mathcal{K}}|\beta_k|^2\right)^{1/2}\cdot\left(\sqrt{1+\eta}+1\right)\cdot\tau_{\mathrm{g}}^{-k_{\min}}. \tag{108}$$

Step 4: enforce term (II) $\le \epsilon_{\mathrm{II}}$ by choosing τ_{g}. Condition (103) is exactly the rearrangement of (108) that yields

$$\left\|\mathbf{Y}_{\mathrm{PLASH}}^{\mathrm{det}} - \mathbf{Y}_{\mathrm{PLASH}}\right\|_F \;\le\; \epsilon_{\mathrm{II}} \qquad \text{on the probability-}1-\delta\text{ event.}$$

Step 5: conclude the end-to-end guarantee. Combine Steps 1–4 with (105). On the probability-$1-\delta$ event,

$$\left\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{PLASH}}\right\|_F \;\le\; \epsilon_{\mathrm{I}} + \epsilon_{\mathrm{det}} + \epsilon_{\mathrm{II}} \;=\; \epsilon_{\mathrm{out}},$$

which proves (104). □

Well-definedness of $L_{\mathrm{post}}(\mathcal{S})$ on the segment $\mathcal{S}$. Lemma H.1 defines $L_{\mathrm{post}}(\mathcal{S})$ as a local Lipschitz constant of the deterministic post-processing map $\mathcal{T}_{\mathbf{Q}}(\cdot)$ restricted to the line segment $\mathcal{S}$ (cf. (87)). We justify that the supremum in (87) is finite under standard architectural choices.

Step 1: $\mathcal{S}$ is compact. By construction, $\mathcal{S}$ is a closed and bounded line segment in a finite-dimensional Euclidean space (it is the image of the compact interval $[0,1]$ under an affine map). Hence $\mathcal{S}$ is compact.

Step 2: $\mathcal{T}_{\mathbf{Q}}$ is continuously differentiable on an open neighborhood of $\mathcal{S}$. The post-processing map $\mathcal{T}_{\mathbf{Q}}(\cdot)$ is a composition of standard primitives (linear maps, residual adds, activation functions, stabilized LayerNorm, and softmax).

1. Linear maps and residual adds are smooth.
2. Common pointwise activations (e.g., GELU / ReLU) are continuous and differentiable almost everywhere; for Lipschitz bounds on a compact set, it suffices that $\mathcal{T}_{\mathbf{Q}}$ is locally Lipschitz and that a Jacobian-based upper bound exists almost everywhere.
3. Softmax is smooth on $\mathbb{R}^m$ (row-wise when applied to a matrix).
4. LayerNorm is smooth provided it includes a stabilizer $\varepsilon_{\mathrm{ln}} > 0$, which avoids division by zero in the variance normalization.

Therefore, under the standard stabilized LayerNorm implementation (assumption (2) in the original list), $\mathcal{T}_{\mathbf{Q}}$ is continuously differentiable on an open set containing $\mathcal{S}$.

Step 3: the Jacobian operator norm is bounded on $\mathcal{S}$. Since $\mathcal{T}_{\mathbf{Q}}$ is continuously differentiable on a neighborhood of $\mathcal{S}$, its Jacobian $J_{\mathcal{T}_{\mathbf{Q}}}(x)$ exists and depends continuously on x for $x\in\mathcal{S}$. The map

$$x \;\longmapsto\; \left\|J_{\mathcal{T}_{\mathbf{Q}}}(x)\right\|_{\mathrm{op}}$$

is continuous (composition of a continuous Jacobian map with the continuous operator-norm function). By compactness of $\mathcal{S}$, this continuous function attains its maximum on $\mathcal{S}$, hence

$$\sup_{x\in\mathcal{S}}\left\|J_{\mathcal{T}_{\mathbf{Q}}}(x)\right\|_{\mathrm{op}} \;<\; \infty.$$

Step 4: a finite Jacobian bound implies a finite Lipschitz constant on $\mathcal{S}$. Let $x,y\in\mathcal{S}$ and consider the curve $\gamma(t) = x + t(y-x)$ for $t\in[0,1]$, which lies entirely in $\mathcal{S}$. By the fundamental theorem of calculus (vector-valued form),

$$\mathcal{T}_{\mathbf{Q}}(y) - \mathcal{T}_{\mathbf{Q}}(x) = \int_0^1 J_{\mathcal{T}_{\mathbf{Q}}}(\gamma(t))\,(y-x)\,dt.$$

Taking norms and using submultiplicativity,

$$\|\mathcal{T}_{\mathbf{Q}}(y) - \mathcal{T}_{\mathbf{Q}}(x)\| \le \int_0^1 \left\|J_{\mathcal{T}_{\mathbf{Q}}}(\gamma(t))\right\|_{\mathrm{op}}\,dt\cdot\|y-x\| \le \Big(\sup_{z\in\mathcal{S}}\|J_{\mathcal{T}_{\mathbf{Q}}}(z)\|_{\mathrm{op}}\Big)\|y-x\|.$$

Thus $\mathcal{T}_{\mathbf{Q}}$ is Lipschitz on $\mathcal{S}$ with a finite constant, and the supremum in (87) is well-defined and finite. In practice, one upper bounds $L_{\mathrm{post}}(\mathcal{S})$ using the same weight-norm / activation-norm composition bounds already used in Lemma H.1.

Checkability and computational overhead for general $\mathcal{K}$. All terms in Theorem H.3 are *forward-pass computable* and do not require forming the $N_q \times N_k$ logit matrix. The sketch sizing rule (e.g., (99) with per-degree parameters) is a one-time design choice (fixed before training / inference).

At inference time:

1. **Stage I term ϵ_{I}.** The bound uses only Stage I tensors (routing weights and compressed keys / values) and is computed from quantities already produced in Stage I (cf. the Stage I bound, e.g., (60)).
2. **Deterministic bias ϵ_{det}.** By definition (101), ϵ_{det} compares the PLASH output to a *deterministic reference* that replaces the randomized Stage II sketches by their deterministic counterparts at each degree $k \in \mathcal{K}$. Crucially, we do *not* compute $\mathbf{Y}^{\mathrm{det}}_{\mathrm{PLASH}}$ explicitly. Instead, (101) expresses ϵ_{det} through intermediate deterministic quantities on the compressed path (Stage I outputs and deterministic Stage II features / embeddings). Thus evaluating ϵ_{det} requires only running the same length-M Stage II pipeline once with the deterministic per-degree maps $\{\mathbf{z}^{\mathrm{det}}_{j,k}\}_{k\in\mathcal{K}}$ (and then applying the same readout / mixer / projections, as required by (101)). This introduces no $N_q \times N_k$ computation.
3. **Stage II stochastic term ϵ_{II} and the sufficient condition.** For general $\mathcal{K}$, the stochastic term is controlled by the discrepancy $\|\mathbf{Y}_{\mathrm{enh}} - \mathbf{Y}^{\mathrm{det}}_{\mathrm{enh}}\|_{2,\infty}$ where both tensors lie in $\mathbb{R}^{M\times d}$. The sufficient condition (103) and guarantee (104) use only such length-M quantities and deterministic post-processing factors (cf. Appendix H), and therefore never introduce an $N_q \times N_k$ computation.

Overall, certification adds only *compressed-side* overhead: it scales with the bottleneck sizes $(M, \{D_k\}_{k\in\mathcal{K}})$ (e.g., one additional deterministic Stage II pass on length M) and preserves PLASH's linear-in-length structure in (N_q, N_k).

H.4. Generality and empirical prevalence of the sufficient condition.

The sufficient condition in Theorem H.3 is *modular*: the proof uses only the deterministic decomposition (97), the Stage I perturbation bound (Lemma F.2), the deterministic post-processing stability on the segment $\mathcal{S}$ (Lemma H.1), and the high-probability Stage II bound (98). Therefore, the same certification pattern applies whenever Stage II admits a forward-pass norm control analogous to (98), while the downstream map $\mathcal{T}_{\mathbf{Q}}(\cdot)$ is deterministic. This is exactly the design intent of PLASH: *localize randomness to Stage II, certify Stage II, then propagate deterministically.*

Beyond structural generality, it is natural to ask how often the certificate is *active* on real inputs. Because the certificate is forward-pass checkable, it defines an empirical notion of *certified accuracy* (or certification rate) in the same spirit as standard robustness certification evaluations (e.g., certified accuracy in randomized smoothing and other certification benchmarks; see, e.g., (Cohen et al., 2019)). Concretely, given a collection of T evaluation inputs $\{(\mathbf{Q}^{(t)}, \mathbf{K}^{(t)}, \mathbf{V}^{(t)})\}_{t=1}^{T}$, define

$$\Delta^{(t)} \triangleq \epsilon_{\mathrm{out}} - \epsilon_{\mathrm{I}}^{(t)} - \epsilon_{\mathrm{det}}^{(t)},$$
$$C^{(t)} \triangleq \sqrt{N_q}\cdot L^{(t)}_{\mathrm{post}}(\mathcal{S})\cdot \|\mathbf{W}_{\mathrm{out}}\|_{\mathrm{op}}\cdot\left(\sum_{k\in\mathcal{K}}|\beta_k|^2\right)^{1/2}\cdot\left(\sqrt{1+\eta}+1\right),$$

where $\epsilon_{\mathrm{I}}^{(t)}$ and $\epsilon_{\mathrm{det}}^{(t)}$ are computed by (100) and (101), and $L^{(t)}_{\mathrm{post}}(\mathcal{S})$ is computed via (87). Then the empirical certification rate is

$$\mathrm{CertRate}(\epsilon_{\mathrm{out}}) \triangleq \frac{1}{T}\sum_{t=1}^{T}\mathbf{1}\left\{\{\Delta^{(t)}>0\}\ \wedge\ \left\{\tau_{\mathrm{g}}\ \geq\ \left(\frac{C^{(t)}}{\Delta^{(t)}}\right)^{1/k_{\min}}\right\}\right\},$$

with $k_{\min}\triangleq \min\mathcal{K}$ (and $k_{\min}=1$ for the $\mathcal{K}=\{1\}$ specialization). This statistic reports *how often real inputs yield a provable* ϵ_{out} guarantee under the chosen hyperparameters, without any additional distributional assumptions.

Why the certified set is not an artificial "tiny" subset. We make the "non-tiny" claim precise by showing that, whenever the certificate holds with *strict slack* at one input, it also holds on a whole neighborhood of nearby inputs.

Step 1: continuity of the forward-pass quantities. Fix the model parameters. Consider the map

$$(\mathbf{Q},\mathbf{K},\mathbf{V})\ \longmapsto\ \Big(\epsilon_{\mathrm{I}}, \epsilon_{\mathrm{det}}, L_{\mathrm{post}}(\mathcal{S})\Big).$$

Each component is obtained by composing continuous primitives: matrix products, softmax, stabilized normalization, and linear maps. Under the standing assumption of Lemma H.1 that the constants defining $L_{\mathrm{post}}(\mathcal{S})$ are finite on the segment $\mathcal{S}$, the construction in (87) yields a finite quantity that varies continuously with the endpoints of $\mathcal{S}$, and thus continuously with the forward-pass inputs. Hence the map above is continuous on any region where the post-processing bound is valid.

Step 2: strict inequalities persist under small perturbations. Suppose the certificate holds at an input $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ with strict slack:

$$\epsilon_{\mathrm{out}} - \epsilon_{\mathrm{I}} - \epsilon_{\mathrm{det}} \;>\; 0 \quad \text{and} \quad \tau_{\mathrm{g}} \;>\; \Big(\frac{C}{\epsilon_{\mathrm{out}} - \epsilon_{\mathrm{I}} - \epsilon_{\mathrm{det}}}\Big)^{1/k_{\min}},$$

where C denotes the corresponding coefficient built from $L_{\mathrm{post}}(\mathcal{S})$ and other forward-pass quantities (as in the definition of $C^{(t)}$ above). By continuity, there exists a neighborhood of $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ in which both strict inequalities remain true. Therefore, the set of certified inputs contains an open neighborhood around any strictly certified input.

Conclusion. Whenever the certified set is nonempty with slack, it contains open subsets of the input space; it is not restricted to isolated, measure-zero configurations. This is precisely why it is meaningful to report $\mathrm{CertRate}(\epsilon_{\mathrm{out}})$: certification is expected to persist across nearby inputs, and the rate quantifies how frequently the model operates in that stable regime.

To summarize, the certificate is not a purely existential statement: it is *auditable at scale*. One can compute $\mathrm{CertRate}(\epsilon_{\mathrm{out}})$ over standard evaluation sets and sweep $(M, \{D_k\}_{k\in\mathcal{K}}, \tau_{\mathrm{g}})$ to characterize when certification is typical. Empirically, the sufficient condition is expected to be active whenever:

1. $\rho_K(M)$ and $\rho_V(M)$ are moderate (routing / compression is not overly lossy);
2. ϵ_{det} is controlled by the learned deterministic reference pipeline; and
3. $L_{\mathrm{post}}(\mathcal{S})$ is moderate (post-processing is locally stable on the realized segment),

all of which can be verified directly from forward-pass logs.

Certificate for $\mathcal{K} = \{1\}$. When $\mathcal{K} = \{1\}$, Stage II introduces *no* higher-order interactions: the enrichment reduces to a randomized *linear* feature map on $\widetilde{\mathbf{G}}$ followed by the same deterministic readout and post-processing. In this regime, PLASH can be interpreted as a controlled approximation to standard attention in which the only approximation sources are: (i) compression in Stage I (captured by ϵ_{I} in (100)), and (ii) randomized features in Stage II (controlled in probability via (99) and propagated through deterministic post-processing). Consequently, Theorem H.3 yields a forward-pass checkable $(\epsilon_{\mathrm{out}}, \delta)$-certificate for $(\mathcal{K} = \{1\})$-PLASH and makes precise the statement that $(\mathcal{K} = \{1\})$-PLASH is a *controlled* approximation to $\mathbf{Y}_{\mathrm{soft}} = \mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V})$.

H.5. Empirical Prevalence of the Certificate in Theorem H.3 with $k = 1$ (Heatmaps over τ_{g} and ϵ_{out}).

Figure 4 shows that the forward-pass certificate from Theorem H.3 is *frequently satisfied on random inputs* and exhibits the expected monotone dependence on the two theorem parameters that directly control certifiability: the Stage II norm-control temperature τ_{g} and the target tolerance ϵ_{out}. To make this statement falsifiable, every pixel in the heatmaps is obtained from an explicit Monte-Carlo estimate of $\mathrm{CertRate}(\epsilon_{\mathrm{out}})$ described below.

Common-range, multi-M visualization. To isolate the effect of the number of groups M (i.e., the number of prototypes / compressed items), we report four panels with $M \in \{16, 32, 48, 64\}$ while keeping all other architectural and sampling hyperparameters fixed. All panels share the *same* swept ranges (a common overlap across M): a 15×15 log-spaced grid with $\tau_{\mathrm{g}} \in [6.31 \times 10^2,\ 1.585 \times 10^3]$ and $\epsilon_{\mathrm{out}} \in [5.20 \times 10^2,\ 8.53 \times 10^2]$. This shared axis range ensures that any shift in the transition band across panels is attributable to M, rather than plot scaling.

Sampling setup. For each grid point $(\tau_{\mathrm{g}}, \epsilon_{\mathrm{out}})$ and each M, we draw $T = 200$ i.i.d. attention inputs $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ with $\mathbf{Q} \in \mathbb{R}^{N_q \times h \times d}$ and $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{N_k \times h \times d}$ (here $N_q = 32$, $N_k = 64$, $h = 4$, $d = 32$). Heads are sampled independently. To reflect variability in input magnitude across trials, we sample a per-trial scale $\sigma \sim \mathrm{Unif}[\sigma_{\min}, \sigma_{\max}]$ (with $\sigma_{\min} = 0.004$, $\sigma_{\max} = 0.08$) and then draw entries of $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ i.i.d. from $\mathcal{N}(0, \sigma^2)$.

PLASH instantiation ($\mathcal{K} = \{1\}$). For each draw we run the $\mathcal{K} = \{1\}$ PLASH block in the standard $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ interface with sketch dimension $D = 256$: Stage I computes routing scores $\mathbf{S} = \mathbf{K}\mathbf{P}^\top$ to M prototypes $\mathbf{P} \in \mathbb{R}^{M \times d}$ and forms compressed summaries $\widetilde{\mathbf{K}} = \mathbf{A}^\top \mathbf{K}$ and $\widetilde{\mathbf{V}} = \mathbf{A}^\top \mathbf{V}$ with $\mathbf{A} = \mathrm{softmax}(\mathbf{S}/\tau)$ (row-wise; $\tau = 1$). Stage II applies the

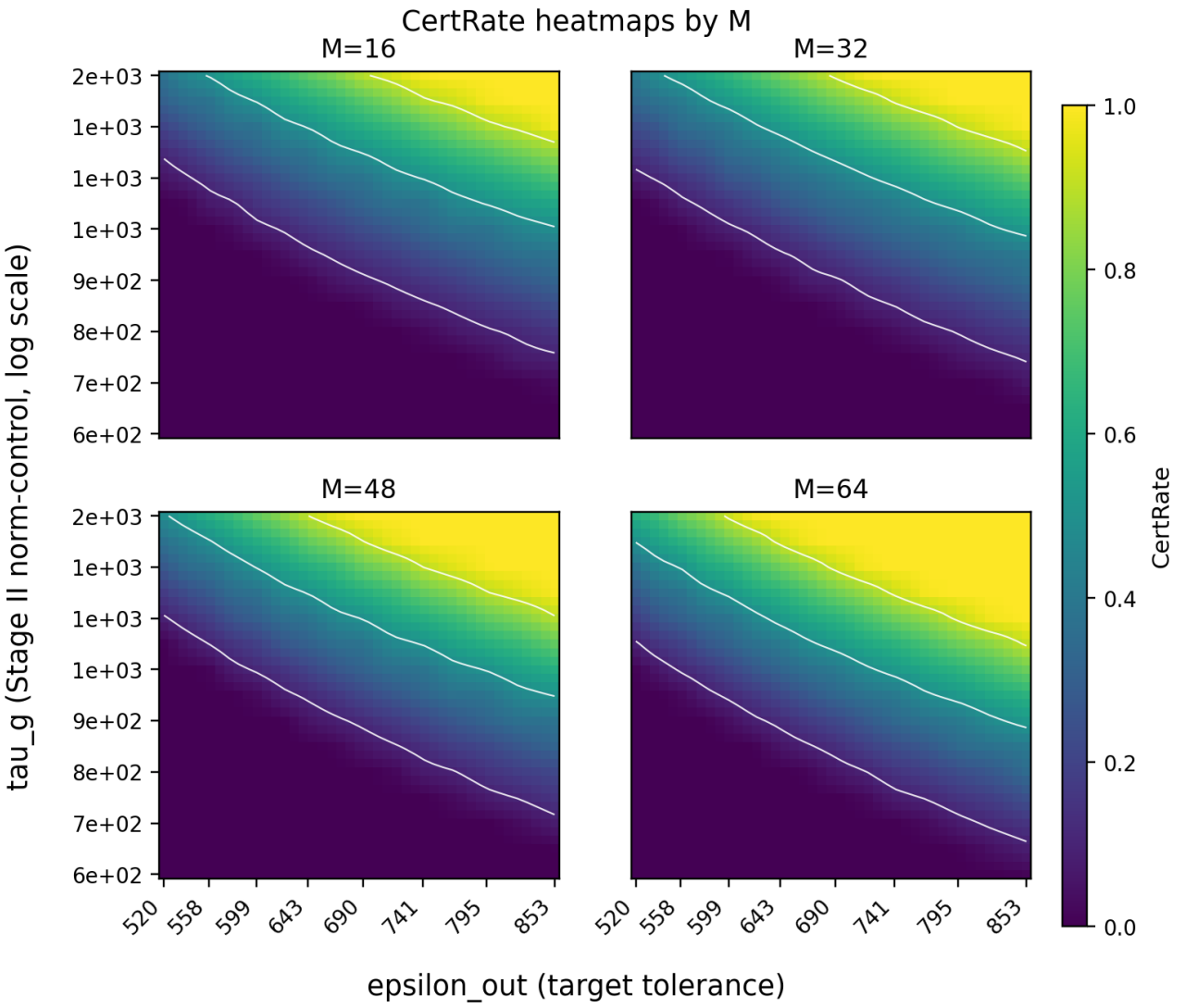


Figure 4. **Empirical certification rate for $\mathcal{K} = \{1\}$ PLASH, shown at a common $(\tau_{\mathrm{g}}, \epsilon_{\mathrm{out}})$ range for different M.** Each panel corresponds to a different number of groups $M \in \{16, 32, 48, 64\}$, while keeping $(N_q, N_k, h, d, D) = (32, 64, 4, 32, 256)$ fixed. Axes are shared across panels: a 15×15 log-spaced grid with $\tau_{\mathrm{g}} \in [6.31 \times 10^2,\ 1.585 \times 10^3]$ and $\epsilon_{\mathrm{out}} \in [5.20 \times 10^2,\ 8.53 \times 10^2]$. Each cell reports $\mathrm{CertRate}(\epsilon_{\mathrm{out}})$ computed from $T = 200$ i.i.d. draws of $(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ with per-trial scale $\sigma \sim \mathrm{Unif}[0.004, 0.08]$. Color encodes the fraction of trials for which the forward-pass condition in Theorem H.3 certifies $\|\mathbf{Y}_{\mathrm{soft}} - \mathbf{Y}_{\mathrm{PLASH}}\|_F \le \epsilon_{\mathrm{out}}$.

pre-map ψ to the concatenated compressed features and enforces stabilized norm control

$$\widetilde{\mathbf{G}}_{j,:} \;=\; \mathbf{G}_{j,:} \big/ \big(\max\{\|\mathbf{G}_{j,:}\|_2, \varepsilon_g\} \cdot \tau_{\mathrm{g}} \big),$$

followed by the only randomized module, CountSketch (TensorSketch with $\mathcal{K} = \{1\}$) of dimension D and a linear readout. The mixer applies a deterministic mixer over the length-M compressed sequence. Stage III then performs *exact* scaled dot-product attention from $\mathbf{Q}$ to the resulting compressed keys / values, producing $\mathbf{Y}_{\mathrm{PLASH}}$. As a baseline we compute the standard attention output $\mathbf{Y}_{\mathrm{soft}} = \mathrm{Atten}(\mathbf{Q}; \mathbf{K}, \mathbf{V})$.

What is counted as "certified". On each trial we compute the forward-pass quantities in Theorem H.3: the Stage I term ϵ_{I} (via the hard-routing reconstruction and radii $\rho_K(M) = \|\mathbf{K} - \mathbf{K}^{\mathrm{q}}\|_{2,\infty}$, $\rho_V(M) = \|\mathbf{V} - \mathbf{V}^{\mathrm{q}}\|_{2,\infty}$), the deterministic bias ϵ_{det} (using the deterministic k=1 comparator obtained by padding / truncation to length D), and a checkable upper bound on $L_{\mathrm{post}}(\mathcal{S})$ as in (87). Given ϵ_{out}, the certificate holds if the slack $\Delta = \epsilon_{\mathrm{out}} - \epsilon_{\mathrm{I}} - \epsilon_{\mathrm{det}}$ is positive and the sufficient condition (103) is satisfied. We then report $\mathrm{CertRate}(\epsilon_{\mathrm{out}}) = T^{-1} \cdot \sum_{t=1}^{T} \mathbf{1}\{\text{certificate holds on trial } t\}$.

How to read Figure 4. Each panel shows a smooth transition from low to high certification as either τ_{g} or ϵ_{out} increases:

increasing ϵ_{out} enlarges the slack Δ, while increasing τ_{g} strengthens the Stage II discrepancy control without changing the Stage III attention operator. Crucially, within the shared window the heatmaps contain substantial mass with $0 < \text{CertRate}(\epsilon_{\text{out}}) < 1$ (quantization step $1/T = 0.005$), providing a direct non-vacuity check: the condition is selective, yet it triggers frequently.

Effect of M: certification becomes easier. Holding $(\tau_{\text{g}}, \epsilon_{\text{out}})$ fixed, the certified region expands with M. Quantitatively, over the common $(\tau_{\text{g}}, \epsilon_{\text{out}})$ grid the mean certification rate increases from 0.246 ($M = 16$) to 0.368 ($M = 64$), and the fraction of grid points with $\text{CertRate} \geq 0.9$ increases from 0.053 to 0.160. The $\text{CertRate} = 0.9$ transition also shifts toward smaller tolerances / temperatures: the median ϵ_{out} needed to reach $\text{CertRate} \geq 0.9$ decreases from ≈ 767 at $M = 16$ to ≈ 715 at $M = 64$, and the median τ_{g} needed decreases from $\approx 1.48 \times 10^3$ to $\approx 1.39 \times 10^3$ (within this window). These shifts match the theorem structure: increasing M reduces the Stage I approximation radii (and thus ϵ_{I}), increasing the slack available for certification.

I. Complexity Analysis of the PLASH Module

This section proves Theorem 3.3 by a stage-wise accounting of the dominant linear-algebra and FFT operations in Algorithm 1. We keep all labels unchanged and state each bound in a form that makes the dependence on (N_q, N_k, M) explicit.

Theorem I.1 (Baseline complexity of Stages I–III (no accuracy constraints))**.** *Consider Algorithm 1 with $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$, $\mathbf{K} \in \mathbb{R}^{N_k \times d_k}$, $\mathbf{V} \in \mathbb{R}^{N_k \times d_v}$, and macro length $M \ll N_k$. Let $\mathcal{K} \subset \mathbb{Z}_{\geq 1}$ be the degree set and let $\{D_k\}_{k \in \mathcal{K}}$ be the TensorSketch dimensions, with $D_{\text{tot}} \triangleq \sum_{k \in \mathcal{K}} D_k$. Assume Stage II computes TensorSketch features via CountSketch followed by FFT-based circular convolutions, and the mixer uses a Transformer-encoder mixer of depth L_{mix}.*

Then the runtime decomposes as

$$T_{\text{PLASH}} = T_{\text{I}} + T_{\text{II}} + T_{\text{III}},$$

with stage-wise bounds

$$T_{\text{I}} = O\Big(N_k \cdot M \cdot d_k\Big) + O\Big(N_k \cdot M\Big) + O\Big(N_k \cdot M \cdot d_k\Big) + O\Big(N_k \cdot M \cdot d_v\Big),$$

$$T_{\text{II}} = O\Big(M \cdot \text{cost}(\psi)\Big) + O\Big(M \cdot d \cdot D_{\text{tot}}\Big) + O\left(M \cdot \sum_{k \in \mathcal{K}} k \cdot \Big(d' + D_k \log D_k\Big)\right) + O\Big(L_{\text{mix}} \cdot \big(M^2 \cdot d + M \cdot d^2\big)\Big),$$

$$T_{\text{III}} = O\Big(N_q \cdot M \cdot d_k\Big) + O\Big(N_q \cdot M\Big) + O\Big(N_q \cdot M \cdot d_v\Big).$$

In particular, if $(d, d', d_k, d_v, \mathcal{K}, \{D_k\}_{k \in \mathcal{K}}, L_{\text{mix}})$ are treated as fixed independently of (N_q, N_k), then

$$T_{\text{PLASH}} \;=\; O\Big(\big(N_q + N_k\big) \cdot M\Big).$$

If the routing weights $\mathbf{A} \in \mathbb{R}^{N_k \times M}$ and the readout attention weights $\text{softmax}\Big(\mathbf{Q}\mathbf{K}_g^\top / \sqrt{d_k}\Big) \in \mathbb{R}^{N_q \times M}$ are materialized, then the peak memory for these two matrices is $O(N_k \cdot M + N_q \cdot M)$. Both Stage I and Stage III admit streaming implementations that avoid storing these matrices.

Proof. We upper bound the arithmetic cost of each stage by explicitly identifying the dominant operations and applying standard cost rules: (i) a dense matrix multiplication $(a \times b) \cdot (b \times c)$ costs $O(abc)$ arithmetic operations; (ii) applying row-wise softmax to an $a \times c$ matrix costs $O(ac)$; (iii) an FFT or IFFT of length D costs $O(D \log D)$ (e.g., Cooley–Tukey; (Cooley & Tukey, 1965)). We suppress constant factors and lower-order terms.

Stage I (routing and macro summaries). Stage I forms routing logits $\mathbf{S}$ and weights $\mathbf{A}$, then aggregates $\mathbf{K}$ and $\mathbf{V}$ using the same $\mathbf{A}$.

(i) *Compute routing logits.* By (2), $\mathbf{S} = \mathbf{K}\mathbf{P}^\top$ where $\mathbf{K} \in \mathbb{R}^{N_k \times d_k}$ and $\mathbf{P}^\top \in \mathbb{R}^{d_k \times M}$. This matrix product costs $O(N_k \cdot M \cdot d_k)$.

(ii) *Compute routing weights.* By (3), $\mathbf{A} = \mathrm{softmax}(\mathbf{S}/\tau, \mathrm{dim} = -1)$ applies a row-wise softmax to an $N_k \times M$ matrix, which costs $O(N_k \cdot M)$.

(iii) *Aggregate keys and values.* By (4), $\widetilde{\mathbf{K}} = \mathbf{A}^\top \mathbf{K}$ where $\mathbf{A}^\top \in \mathbb{R}^{M \times N_k}$ and $\mathbf{K} \in \mathbb{R}^{N_k \times d_k}$, so the cost is $O(N_k \cdot M \cdot d_k)$. Similarly, by (5), $\widetilde{\mathbf{V}} = \mathbf{A}^\top \mathbf{V}$ costs $O(N_k \cdot M \cdot d_v)$.

Summing (i)–(iii) yields the stated bound for T_{I}.

Stage II (macro core: pre-map, TensorSketch, linear readout, and mixer). Stage II has four components. We bound each separately and then add them.

(i) Pre-map. The map ψ is applied row-wise to M inputs. By definition of $\mathrm{cost}(\psi)$, this contributes $O\big(M \cdot \mathrm{cost}(\psi)\big)$.

(ii) TensorSketch features. Fix $k \in \mathcal{K}$ and one row $\widetilde{\mathbf{G}}_{j,:} \in \mathbb{R}^{d'}$. By assumption, $\mathrm{TS}_k(\cdot; D_k)$ is implemented as: (a) k CountSketch maps of the length-d' input into $\mathbb{R}^{D_k}$, followed by (b) a k-fold circular convolution computed with FFTs.

(a) *CountSketch passes.* Constructing one CountSketch vector from a length-d' input is a single pass over the coordinates and costs $O(d')$. Therefore, k independent CountSketch maps cost $O(k \cdot d')$.

(b) *FFT-based convolution.* Let $\mathbf{c}_{k,1}, \ldots, \mathbf{c}_{k,k} \in \mathbb{R}^{D_k}$ denote the k CountSketch outputs. The standard FFT implementation computes the k-fold circular convolution as

$$\mathrm{IFFT}(\mathrm{FFT}(\mathbf{c}_{k,1}) \odot \cdots \odot \mathrm{FFT}(\mathbf{c}_{k,k}))\,.$$

This requires k FFTs and one IFFT of length D_k, plus $k-1$ pointwise products. Using the FFT cost rule, the transform cost is $O(k \cdot D_k \log D_k)$. The pointwise products cost $O(k \cdot D_k)$, which is dominated by $O(k \cdot D_k \log D_k)$ for $D_k \geq 2$. Hence the convolution step costs $O(k \cdot D_k \log D_k)$.

Combining (a) and (b), the per-row, per-degree cost is

$$O\Big(k \cdot d' + k \cdot D_k \log D_k\Big) = O\Big(k \cdot (d' + D_k \log D_k)\Big).$$

Summing over all $j \in [M]$ and all $k \in \mathcal{K}$ yields

$$O\left(M \cdot \sum_{k \in \mathcal{K}} k \cdot \Big(d' + D_k \log D_k\Big)\right),$$

as stated.

(iii) Linear readout. For each $j \in [M]$, the feature vector $\mathbf{v}_j \in \mathbb{R}^{D_{\mathrm{tot}}}$ is mapped to $\mathbf{Y}_{\mathrm{grp}}[j,:] = \mathbf{W}_{\mathrm{out}} \mathbf{v}_j$ with $\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{d \times D_{\mathrm{tot}}}$. This costs $O(d \cdot D_{\mathrm{tot}})$ per j, hence $O(M \cdot d \cdot D_{\mathrm{tot}})$ total.

(iv) Macro mixer. We bound one encoder layer in Definition A.7 on an input in $\mathbb{R}^{M \times d}$ and then multiply by L_{mix}.

(a) Multi-head self-attention (MHSA). Fix a layer ℓ. For each head $h \in [H]$, the three projections $\mathbf{UW}_Q^{(\ell,h)}$, $\mathbf{UW}_K^{(\ell,h)}$, $\mathbf{UW}_V^{(\ell,h)}$ have shape $(M \times d) \cdot (d \times d_h)$ and cost $O(M \cdot d \cdot d_h)$ each. Across three projections and H heads, this gives $O(H \cdot M \cdot d \cdot d_h) = O(M \cdot d^2)$ using $H d_h = d$.

The logit matrix per head is $(M \times d_h) \cdot (d_h \times M)$, which costs $O(M^2 \cdot d_h)$, and across H heads this is $O(M^2 \cdot d)$. The row-wise softmax on each $M \times M$ logit matrix costs $O(M^2)$ per head and hence $O(H \cdot M^2)$ total, which is subsumed by $O(M^2 \cdot d)$ in the standard regime $d \geq H$ (equivalently $d_h \geq 1$). The attention-weighted value aggregation per head costs $O(M^2 \cdot d_h)$ and hence $O(M^2 \cdot d)$ total. Finally, the output projection is $(M \times d) \cdot (d \times d)$ and costs $O(M \cdot d^2)$. Therefore, one MHSA block satisfies

$$T_{\mathrm{MHSA},\ell} = O(M^2 \cdot d) + O(M \cdot d^2).$$

(b) Position-wise feed-forward network (FFN). The FFN applies two matrix multiplications: $(M \times d) \cdot (d \times d_{\mathrm{ff}})$ and $(M \times d_{\mathrm{ff}}) \cdot (d_{\mathrm{ff}} \times d)$, contributing $O(M \cdot d \cdot d_{\mathrm{ff}})$ total. Under the standard Transformer choice $d_{\mathrm{ff}} = \Theta(d)$ (e.g., Section 3.3 in (Vaswani et al., 2017)), this becomes $O(M \cdot d^2)$.

(c) LayerNorm and residual. LayerNorm and residual additions are row-wise and cost $O(M \cdot d)$ each, which is lower-order than $O(M^2 \cdot d)$ and $O(M \cdot d^2)$.

(d) Per-layer and total mixer cost. Combining (a)–(c), one encoder layer costs $O(M^2 \cdot d) + O(M \cdot d^2)$, and the depth-$L_{\mathrm{mix}}$ mixer costs

$$T_{\mathrm{mixer}} = O\Big(L_{\mathrm{mix}} \cdot (M^2 \cdot d + M \cdot d^2)\Big).$$

This yields the stated mixer term in T_{II}.

Conclusion for Stage II. Adding (i)–(iv) gives the stated bound for T_{II}.

Stage III (exact attention readout). Stage III computes exact scaled dot-product attention between N_q queries and M compressed keys / values.

(i) *Logits.* Compute $\mathbf{L} = \mathbf{Q}\mathbf{K}_g^\top \in \mathbb{R}^{N_q \times M}$ with $\mathbf{Q} \in \mathbb{R}^{N_q \times d_k}$ and $\mathbf{K}_g \in \mathbb{R}^{M \times d_k}$. This costs $O(N_q \cdot M \cdot d_k)$.

(ii) *Softmax.* Row-wise softmax on an $N_q \times M$ matrix costs $O(N_q \cdot M)$.

(iii) *Weighted sum.* Multiply the attention weights $(N_q \times M)$ by $\mathbf{V}_g \in \mathbb{R}^{M \times d_v}$, which costs $O(N_q \cdot M \cdot d_v)$.

Summing (i)–(iii) yields the stated bound for T_{III}.

Total runtime and simplified scaling. Summing the bounds for T_{I}, T_{II}, and T_{III} gives the stated decomposition. If $(d, d', d_k, d_v, \mathcal{K}, \{D_k\}_{k\in\mathcal{K}}, L_{\mathrm{mix}})$ are fixed independently of (N_q, N_k), then all Stage II terms depend only on M (and fixed constants), while Stage I and Stage III contribute the only dependence on N_k and N_q. Therefore,

$$T_{\mathrm{PLASH}} = O(N_k \cdot M) + O(N_q \cdot M) = O\big((N_q + N_k) \cdot M\big).$$

Memory and streaming. If one explicitly stores $\mathbf{A} \in \mathbb{R}^{N_k \times M}$ and the readout attention weights $\mathrm{softmax}(\mathbf{Q}\mathbf{K}_g^\top/\sqrt{d_k}) \in \mathbb{R}^{N_q \times M}$, then these two matrices contribute $O(N_k \cdot M + N_q \cdot M)$ peak storage.

Both can be avoided by standard streaming:

- *Stage I:* compute $\mathbf{S}$ and $\mathbf{A}$ row-by-row and immediately accumulate $\widetilde{\mathbf{K}} = \sum_{i=1}^{N_k} \mathbf{A}_{i,:}^\top \mathbf{K}_{i,:}$ and $\widetilde{\mathbf{V}} = \sum_{i=1}^{N_k} \mathbf{A}_{i,:}^\top \mathbf{V}_{i,:}$, storing only the running sums in $\mathbb{R}^{M \times d_k}$ and $\mathbb{R}^{M \times d_v}$.
- *Stage III:* compute attention one query row at a time: for each query q, form the length-M logit vector, apply softmax, and multiply by $\mathbf{V}_g$. This avoids storing the full $N_q \times M$ weight matrix.

The remaining working storage is dominated by macro-side activations (size $O(M \cdot d)$ and $O(M \cdot D_{\mathrm{tot}})$) and FFT work arrays (size $O(\max_{k\in\mathcal{K}} D_k)$), none of which scales with $N_q N_k$. □