A Theoretical Analysis of Migration Strength Selection in SmoothQuant

Zhihang Liu*, Chiu-Wing Sham*, Jiale Li*, Sean Longyu Ma*, Chong Fu†
*School of Computer Science, The University of Auckland, Auckland, New Zealand
†School of Computer Science and Engineering, Northeastern University, Shenyang, China
{zliu604, jli990}@aucklanduni.ac.nz, {sean.ma, b.sham}@auckland.ac.nz, fuchong@mail.neu.edu.cn

***Abstract*—Post-training quantization (PTQ) is essential for efficient inference of large language models, yet remains challenging due to activation outliers with large magnitudes. SmoothQuant mitigates this issue by migrating quantization difficulty from activations to weights using a global migration strength α. However, the choice of α in prior work is empirically motivated and lacks theoretical grounding, while a global α ignores layer heterogeneity. In this paper, we provide a theoretical analysis of SmoothQuant by formulating it as a quantization-noise balancing problem. Under standard quantization noise assumptions, we derive an explicit objective that characterizes the trade-off between activation and weight quantization errors, revealing that the error-minimizing migration strength depends on layer-specific activation and weight statistics. Motivated by this analysis, we propose a layer-wise smoothing strategy in which each layer adopts its own theory-guided α. With minimal modifications to the original method and no additional runtime overhead, the proposed approach consistently reduces layer output error and improves accuracy on LLaMA models under W8A8 quantization, demonstrating the practical effectiveness of the theoretical insights. Our implementation is publicly available.[1]**

***Index Terms*—Post-Training Quantization, Large Language Model, SmoothQuant**

I. Introduction

Large language models (LLMs) have achieved remarkable performance across a wide range of natural language processing tasks, but their deployment is often constrained by substantial memory and computation costs. PTQ has emerged as an attractive solution for efficient inference, as it enables low-bit integer execution without retraining or access to large-scale training data [1]–[3]. However, applying PTQ to LLMs remains challenging due to the presence of activation outliers and highly unbalanced dynamic ranges, which can cause severe accuracy degradation under low-bit quantization [4]–[6].

Recently, hardware complexity is still a concern [7], [8]. Among existing PTQ approaches, SmoothQuant [9] stands out as a representative method that explicitly quantizes activations while preserving hardware-efficient integer execution. Technically, SmoothQuant addresses activation-dominated quantization error by introducing a simple yet effective transformation that redistributes quantization difficulty between activations and weights. Owing to its simplicity, effectiveness, and hardware-friendly design, SmoothQuant has been widely adopted as a core component in practical LLM quantization pipelines. In its original formulation, SmoothQuant introduces a smoothing parameter α to control the migration strength between activations and weights, and applies a single global value across all layers, with $\alpha = 0.5$ commonly used in practice [9].

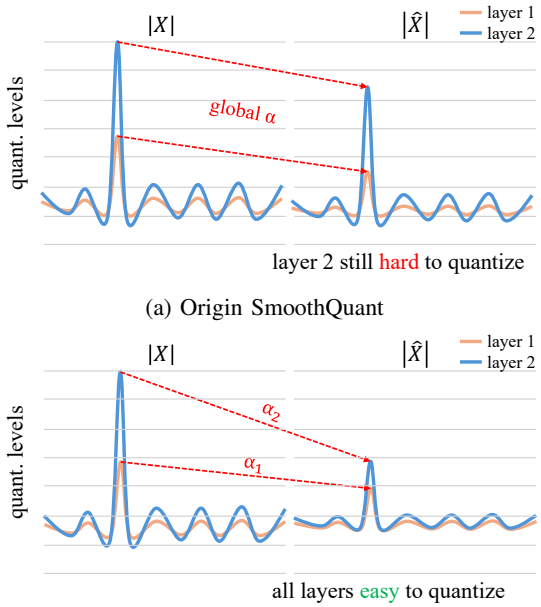


Fig. 1: Global versus layer-wise smoothing in SmoothQuant. (a) With a single global migration strength, smoothing is effective for some layers but insufficient for others. (b) Layer-wise smoothing applies different migration strengths to different layers, reducing outliers and making all layers easier to quantize.

While this design explicitly rebalances the quantization difficulty between activations and weights, the use of a shared migration strength across layers implicitly assumes homogeneous quantization difficulty throughout the network. In practice, however, activation and weight statistics can vary substantially across layers, especially in deep Transformer architectures. Moreover, the choice of the migration strength α has not been theoretically characterized in terms of optimality. These observations raise a fundamental question: what is the optimal migration strength in SmoothQuant, and should it

[1]https://github.com/zhihangliuuoa/analysis-of-smoothquant

be uniform across layers? Answering this question is crucial for understanding the theoretical limits of SmoothQuant and for developing principled post-training quantization strategies beyond fixed parameter choices. Existing extensions and variants of SmoothQuant mainly explore empirical refinements or architectural adaptations [10], but do not provide a theoretical analysis of the migration strength or its optimality.

In this paper, we present a theoretical analysis of SmoothQuant that formulates the migration strength as a quantization error minimization problem. Under a standard quantization noise approximation, we show that the smoothing transformation induces an explicit trade-off between activation and weight quantization errors. This formulation enables an analytical characterization of the migration strength that minimizes the expected output error under the adopted noise model. Our analysis further reveals that the commonly used uniform migration strength is optimal only under restrictive symmetry conditions, and that the error-minimizing migration strength is generally dependent on layer-specific statistics.

Building on this analysis, we show that allowing the migration strength to vary across layers provides a closer approximation to the theoretical optimum. We therefore propose a layer-wise adaptation method that estimates migration strengths using layer-specific activation and weight statistics. Importantly, our approach preserves the key advantages of SmoothQuant: the migration strengths are computed offline and folded into the quantized weights, introducing no additional runtime overhead and remaining fully compatible with existing integer-only inference kernels. Extensive experiments validate our theoretical analysis, demonstrating that the derived formulation captures empirical error trends across layers and provides effective guidance for selecting layer-specific migration strengths, consistently improving performance over uniform smoothing across multiple models.

In summary, our contributions are as follows:

- We provide a theoretical analysis of SmoothQuant that formulates migration strength selection as an explicit quantization error balancing problem under standard noise assumptions.
- We show that the error-minimizing migration strength is generally layer-dependent, and that uniform smoothing arises only as a special case under strong symmetry assumptions.
- We propose a practical layer-wise smoothing strategy that reduces quantization error and improves downstream performance without sacrificing the hardware-friendly properties of SmoothQuant.

II. Background

A. Post-Training Quantization

PTQ aims to reduce the numerical precision of neural network parameters and activations without retraining, enabling efficient deployment on hardware that supports low-bit integer arithmetic. In this work, we focus on uniform symmetric quantization, which is widely adopted in existing integer-only inference kernels.

Method	Year	Precision	Customized kernel
LLM.int8() [4]	2022	W8A16 / W8A8	✗
ZeroQuant [5]	2022	W8A8 (per-token)	✓
GPTQ [11]	2022	W4A16 / W8A16	✓
AWQ [12]	2024	W4A16 / W8A16	✓
SmoothQuant [9]	2023	W8A8	✗

TABLE I: Comparison of representative post-training quantization methods for LLMs.

Given a real-valued tensor $x \in \mathbb{R}^n$, uniform quantization maps x to an integer tensor $\hat{x}$ as

$$\hat{x} = \mathrm{clip}\Big(\Big\lfloor \frac{x}{\Delta} \Big\rceil, q_{\min}, q_{\max}\Big), \tag{1}$$

where Δ denotes the quantization step size, $\lfloor \cdot \rceil$ is the round-to-nearest operator, and $[q_{\min}, q_{\max}]$ specifies the integer range (e.g., $[-128, 127]$ for INT8). The dequantized tensor is given by $\tilde{x} = \Delta\hat{x}$. For symmetric quantization, the step size is determined by the maximum absolute value of the tensor:

$$\Delta = \frac{\max(|x|)}{q_{\max}}. \tag{2}$$

While PTQ avoids the computational cost of retraining, its accuracy is highly sensitive to outliers and extreme values. Large dynamic ranges lead to large step sizes, which in turn increase quantization error for the majority of values. This issue is particularly severe in LLMs, where Transformer-based architectures exhibit highly heterogeneous activation and weight distributions across layers. Activation tensors often contain a small number of outlier values with magnitudes significantly larger than the bulk of the distribution. Such outliers dominate the dynamic range used to determine the quantization step size, causing severe information loss under low-bit quantization. As a result, naive PTQ often leads to unacceptable accuracy degradation when applied to LLMs, motivating the development of specialized quantization techniques that explicitly address activation outliers.

Existing PTQ methods for LLMs can be broadly categorized by whether they quantize activations and whether they preserve hardware-efficient integer execution. Table I summarizes representative PTQ approaches along these dimensions. Weight-only methods such as GPTQ and AWQ avoid activation quantization by retaining floating-point activations, while mixed-precision approaches such as LLM.int8() selectively preserve outlier channels in higher precision. In contrast, SmoothQuant enables hardware-friendly W8A8 quantization by explicitly addressing activation outliers, making it a natural baseline for activation-aware PTQ.

B. SmoothQuant

SmoothQuant [9] addresses the activation outlier problem by redistributing quantization difficulty between activations and weights through a simple scaling transformation. Consider a linear layer

$$Y = XW, \tag{3}$$

where $X \in \mathbb{R}^{B\times d}$ denotes the activation matrix and $W \in \mathbb{R}^{d\times k}$ denotes the weight matrix. SmoothQuant introduces a diagonal scaling matrix $S = \mathrm{diag}(s)$ and rewrites the computation as

$$Y = (XS^{-1})(SW), \tag{4}$$

which is mathematically equivalent to the original formulation. By choosing appropriate scaling factors s, SmoothQuant reduces the dynamic range of activations while correspondingly increasing that of weights, enabling more balanced quantization.

In practice, SmoothQuant parameterizes the scaling factors as

$$s_i = \frac{\max(|X_i|)^{\alpha}}{\max(|W_i|)^{1-\alpha}}, \tag{5}$$

where X_i and W_i denote the activation and weight values associated with the i-th channel, and $\alpha \in [0, 1]$ is a migration strength that controls the trade-off between activation and weight scaling.

In the original SmoothQuant formulation, a single global migration strength α (typically $\alpha = 0.5$) is applied uniformly across all layers to balance activation and weight quantization difficulty. While this design is simple and effective in practice, the choice of α has not been theoretically characterized in terms of optimality. Moreover, applying a shared migration strength across layers implicitly assumes homogeneous quantization difficulty throughout the network. In practice, however, activation and weight statistics can vary substantially across layers and channels, especially in deep Transformer architectures. Fig. 1 conceptually illustrates this effect, showing that a global migration strength may smooth different layers unevenly.

III. Theoretical Analysis

In this section, we provide a rigorous theoretical analysis of SmoothQuant's smoothing mechanism and derive the optimal migration strength for minimizing quantization error. We begin by establishing the theoretical optimum under per-channel smoothing, then analyze the limitations of SmoothQuant's global parameterization, and finally demonstrate why layer-wise adaptation is necessary and effective.

A. Optimal Per-Channel Smoothing Factors

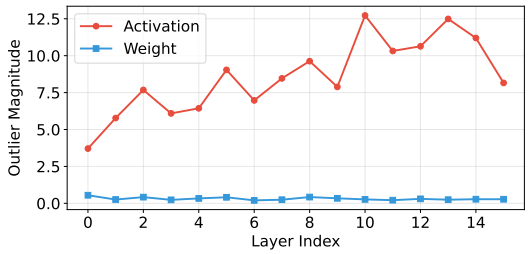
Consider a linear transformation in the ℓ-th layer of a neural network:

$$\mathbf{Y}^{(\ell)} = \mathbf{X}^{(\ell)}\mathbf{W}^{(\ell)} \tag{6}$$

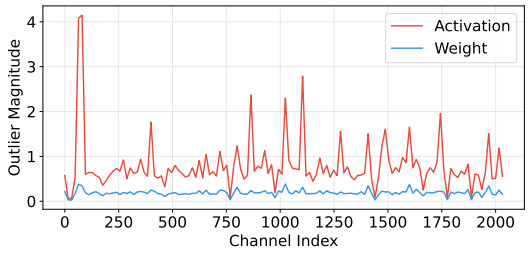
where $\mathbf{X}^{(\ell)} \in \mathbb{R}^{N\times d}$ denotes the input activations, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d\times m}$ denotes the weight matrix, and $\mathbf{Y}^{(\ell)} \in \mathbb{R}^{N\times m}$ is the output. Here, N is the number of tokens, d is the input dimension, and m is the output dimension.

a) Quantization noise model: We adopt the standard uniform quantization model with additive noise:

$$\hat{\mathbf{X}}^{(\ell)} = \mathbf{X}^{(\ell)} + \boldsymbol{\epsilon}_X^{(\ell)}, \quad \hat{\mathbf{W}}^{(\ell)} = \mathbf{W}^{(\ell)} + \boldsymbol{\epsilon}_W^{(\ell)} \tag{7}$$



(a) Outlier magnitudes across layers



(b) Outlier magnitudes across channels (layer 0)

Fig. 2: Layer-wise and channel-wise statistics of activations and weights in LLaMA-3.2-1B, measured on the calibration dataset. (a) Activation outliers vary substantially across layers, while weight outliers remain relatively stable. (b) Activation statistics exhibit larger variability across channels compared to weights, which show consistently smaller ranges.

where the quantization noise satisfies:

$$\boldsymbol{\epsilon}_X^{(\ell)} \sim \mathcal{U}\left(-\frac{\Delta_X^{(\ell)}}{2}, \frac{\Delta_X^{(\ell)}}{2}\right), \quad \boldsymbol{\epsilon}_W^{(\ell)} \sim \mathcal{U}\left(-\frac{\Delta_W^{(\ell)}}{2}, \frac{\Delta_W^{(\ell)}}{2}\right) \tag{8}$$

with variances $\mathbb{E}[(\boldsymbol{\epsilon}_X^{(\ell)})^2] = (\Delta_X^{(\ell)})^2/12$ and $\mathbb{E}[(\boldsymbol{\epsilon}_W^{(\ell)})^2] = (\Delta_W^{(\ell)})^2/12$.

SmoothQuant applies a channel-wise scaling transformation:

$$\mathbf{X}'^{(\ell)} = \mathbf{X}^{(\ell)}\,\mathrm{diag}(\mathbf{s}^{(\ell)})^{-1}, \quad \mathbf{W}'^{(\ell)} = \mathrm{diag}(\mathbf{s}^{(\ell)})\mathbf{W}^{(\ell)} \tag{9}$$

where $\mathbf{s}^{(\ell)} \in \mathbb{R}^d$ is the per-channel smoothing factor. This transformation is mathematically equivalent: $\mathbf{Y}^{(\ell)} = \mathbf{X}'^{(\ell)}\mathbf{W}'^{(\ell)}$.

b) Per-channel quantization error: After smoothing, the quantization step sizes for the j-th channel become:

$$\Delta_{X',j}^{(\ell)} \propto \frac{X_{\max,j}^{(\ell)}}{s_j^{(\ell)}}, \quad \Delta_{W',j}^{(\ell)} \propto s_j^{(\ell)} W_{\max,j}^{(\ell)} \tag{10}$$

where $X_{\max,j}^{(\ell)} = \max_n |X_{nj}^{(\ell)}|$ and $W_{\max,j}^{(\ell)} = \max_i |W_{ji}^{(\ell)}|$ denote the per-channel maximum magnitudes.

The quantization noise variances are:

$$\sigma_{X,j}^{2(\ell)} = \frac{(\Delta_{X',j}^{(\ell)})^2}{12} \propto \frac{(X_{\max,j}^{(\ell)})^2}{(s_j^{(\ell)})^2}$$
$$\sigma_{W,j}^{2(\ell)} = \frac{(\Delta_{W',j}^{(\ell)})^2}{12} \propto (s_j^{(\ell)})^2 (W_{\max,j}^{(\ell)})^2 \tag{11}$$

c) Output error propagation: The expected squared error in the output, neglecting second-order terms, is:

$$\begin{aligned}\mathcal{E}^{(\ell)} &= \mathbb{E}\left[\|\hat{\mathbf{Y}}^{(\ell)} - \mathbf{Y}^{(\ell)}\|_F^2\right] \\ &\approx \mathbb{E}\left[\|\boldsymbol{\epsilon}_{X'}^{(\ell)}\mathbf{W}'^{(\ell)}\|_F^2\right] + \mathbb{E}\left[\|\mathbf{X}'^{(\ell)}\boldsymbol{\epsilon}_{W'}^{(\ell)}\|_F^2\right]\end{aligned} \tag{12}$$

Assuming statistical independence and decomposing channel-by-channel:

$$\mathcal{E}^{(\ell)} \approx \sum_{j=1}^{d}\left[v_{W,j}^{(\ell)} \cdot \sigma_{X,j}^{2(\ell)} + v_{X,j}^{(\ell)} \cdot \sigma_{W,j}^{2(\ell)}\right] \tag{13}$$

where:

- $v_{X,j}^{(\ell)} = \mathbb{E}[(X_j^{(\ell)})^2]$ is the **activation variance** of channel j,
- $v_{W,j}^{(\ell)} = \|\mathbf{W}_{j,:}^{(\ell)}\|_2^2$ is the **weight variance** of row j.

These variance terms act as **importance weights** that modulate the contribution of quantization noise to the output error.

Theorem 1 (Optimal Per-Channel Smoothing)**.** *For each channel j, the smoothing factor that minimizes the quantization-induced error is:*

$$(s_j^{(\ell)})^* = \left(\frac{v_{W,j}^{(\ell)} \cdot (X_{\max,j}^{(\ell)})^2}{v_{X,j}^{(\ell)} \cdot (W_{\max,j}^{(\ell)})^2}\right)^{1/4} \tag{14}$$

Proof. For a fixed channel j, the error contribution is:

$$\mathcal{E}_j^{(\ell)}(s_j) = v_{W,j}^{(\ell)} \cdot \frac{(X_{\max,j}^{(\ell)})^2}{(s_j)^2} + v_{X,j}^{(\ell)} \cdot (s_j)^2 (W_{\max,j}^{(\ell)})^2 \tag{15}$$

Taking the derivative with respect to s_j and setting it to zero:

$$\frac{\partial \mathcal{E}_j^{(\ell)}}{\partial s_j} = -2v_{W,j}^{(\ell)} \cdot \frac{(X_{\max,j}^{(\ell)})^2}{(s_j)^3} + 2v_{X,j}^{(\ell)} \cdot s_j (W_{\max,j}^{(\ell)})^2 = 0 \tag{16}$$

Solving for s_j yields:

$$(s_j)^4 = \frac{v_{W,j}^{(\ell)} \cdot (X_{\max,j}^{(\ell)})^2}{v_{X,j}^{(\ell)} \cdot (W_{\max,j}^{(\ell)})^2} \tag{17}$$

which gives the result in Eq. (14). The second derivative is positive, confirming this is a minimum. □

d) Interpretation: The optimal smoothing factor $(s_j^{(\ell)})^*$ balances the quantization difficulty between activations and weights by considering:

1) **Magnitude ratio**: $X_{\max,j}^{(\ell)}/W_{\max,j}^{(\ell)}$ determines the relative quantization ranges.
2) **Variance ratio**: $v_{W,j}^{(\ell)}/v_{X,j}^{(\ell)}$ weights the importance of each channel's contribution to output error.

e) Theoretical significance: Theorem 1 establishes the **theoretical lower bound** on quantization error achievable through smoothing transformations. However, this per-channel solution requires per-channel activation quantization, which is incompatible with efficient INT8 GEMM kernels (as discussed in Section II). This motivates the need for parameterized approximations.

B. SmoothQuant's Parameterization and Limitation

In its original formulation, SmoothQuant applies a single migration strength to all layers, without explicitly distinguishing potential differences in activation and weight statistics across layers. To understand the implications of this design choice, we analyze the migration strength at the level of an individual layer ℓ, and study how it affects the quantization error given layer-specific activation and weight statistics. To maintain hardware efficiency, SmoothQuant constrains the smoothing factors to a parameterized family indexed by a single scalar migration strength $\alpha^{(\ell)}$:

$$s_j^{(\ell)}(\alpha^{(\ell)}) = \frac{(X_{\max,j}^{(\ell)})^{\alpha^{(\ell)}}}{(W_{\max,j}^{(\ell)})^{1-\alpha^{(\ell)}}}, \quad j = 1, \ldots, d \tag{18}$$

This parameterization enables per-token activation quantization while migrating quantization difficulty from activations to weights in a hardware-friendly manner. The migration strength $\alpha^{(\ell)} \in [0, 1]$ controls this trade-off, with $\alpha^{(\ell)} = 0$ corresponding to no smoothing, $\alpha^{(\ell)} = 1$ to maximum smoothing, and $\alpha^{(\ell)} = 0.5$ representing equal difficulty sharing as adopted in SmoothQuant.

a) Optimal migration strength under SmoothQuant's parameterization: Given this constrained family, we consider the problem of selecting the migration strength that minimizes the quantization error for a fixed layer ℓ:

$$\begin{aligned}\mathcal{E}^{(\ell)} = \arg\min_{\alpha^{(\ell)}} \sum_{j=1}^{d}\Bigg[&v_{W,j}^{(\ell)} \cdot \frac{(X_{\max,j}^{(\ell)})^2}{(s_j^{(\ell)}(\alpha^{(\ell)}))^2} \\ &+ v_{X,j}^{(\ell)} \cdot (s_j^{(\ell)}(\alpha^{(\ell)}))^2 (W_{\max,j}^{(\ell)})^2\Bigg]\end{aligned} \tag{19}$$

Substituting Eq. (18) yields

$$\begin{aligned}\mathcal{E}^{(\ell)} = \arg\min_{\alpha^{(\ell)}} \sum_{j=1}^{d}\Bigg[&v_{W,j}^{(\ell)} \cdot (X_{\max,j}^{(\ell)})^{2(1-\alpha^{(\ell)})} (W_{\max,j}^{(\ell)})^{2(1-\alpha^{(\ell)})} \\ &+ v_{X,j}^{(\ell)} \cdot (X_{\max,j}^{(\ell)})^{2\alpha^{(\ell)}} (W_{\max,j}^{(\ell)})^{2\alpha^{(\ell)}}\Bigg]\end{aligned} \tag{20}$$

Due to heterogeneous channel-wise statistics and the exponential dependence on the migration strength, the resulting optimality condition is transcendental and does not admit a closed-form solution in the general case.

b) A special case and its implication: Under the *uniform channel assumption*, where $X_{\max,j}^{(\ell)} \approx X_{\max}^{(\ell)}$, $W_{\max,j}^{(\ell)} \approx W_{\max}^{(\ell)}$, and $v_{X,j}^{(\ell)} \approx v_X^{(\ell)}$, $v_{W,j}^{(\ell)} \approx v_W^{(\ell)}$ for all j, Eq. (20) simplifies to

$$\begin{aligned}\mathcal{E}^{(\ell)} = \arg\min_{\alpha^{(\ell)}}\Bigg[&v_W^{(\ell)} \cdot (X_{\max}^{(\ell)} W_{\max}^{(\ell)})^{2(1-\alpha^{(\ell)})} \\ &+ v_X^{(\ell)} \cdot (X_{\max}^{(\ell)} W_{\max}^{(\ell)})^{2\alpha^{(\ell)}}\Bigg]\end{aligned} \tag{21}$$

Solving the stationary condition yields

$$v_W^{(\ell)} \cdot (X_{\max}^{(\ell)} W_{\max}^{(\ell)})^{2(1-\alpha^{(\ell)*})} = v_X^{(\ell)} \cdot (X_{\max}^{(\ell)} W_{\max}^{(\ell)})^{2\alpha^{(\ell)*}}. \tag{22}$$

When $v_X^{(\ell)} \approx v_W^{(\ell)}$, we obtain $\alpha^{(\ell)*} \approx 0.5$, recovering SmoothQuant's default choice.

It is important to note that the above analysis characterizes the optimal migration strength at the level of an individual layer ℓ, where the optimal value is defined with respect to layer-specific activation and weight statistics. In contrast, the original SmoothQuant formulation enforces a single global migration strength across all layers, effectively constraining $\alpha^{(\ell)} = \alpha$ for all ℓ. Under this global constraint, the migration strength can be optimal only when the relative quantization sensitivity between activations and weights is identical across layers. When layer-wise statistics differ, as illustrated in Fig. 2, this shared parameterization generally prevents each layer from achieving its own error-minimizing migration strength. This limitation motivates the need for a layer-wise treatment of the migration strength, which we develop in the following section.

C. Layer-wise Optimal Smoothing

Our key insight is that **layer-wise adaptation** of the migration strength provides a principled way to relax the global parameterization constraint while preserving the hardware efficiency of SmoothQuant. By allowing each layer to use its own migration strength $\alpha^{(\ell)*}$, we obtain a closer approximation to the theoretical per-channel optimum (Theorem 1).

In practice, the assumptions underlying a uniform migration strength are violated in real transformer models. As shown in Fig. 2(a), the activation outlier magnitude varies significantly across layers, indicating heterogeneous quantization sensitivity. Consequently, different layers require different levels of migration strength, and using a fixed $\alpha = 0.5$ for all layers is generally suboptimal. Formally, the optimal migration strength for each layer is defined as the minimizer of its layer-specific quantization error:

$$\alpha^{(\ell)*} = \underset{\alpha^{(\ell)} \in [0,1]}{\arg\min} \mathcal{E}^{(\ell)}(\alpha^{(\ell)}) \tag{23}$$

By definition, $\alpha^{(\ell)*}$ minimizes $\mathcal{E}^{(\ell)}$ for each layer:

$$\mathcal{E}^{(\ell)}(\alpha^{(\ell)*}) \leq \mathcal{E}^{(\ell)}(\alpha_{\text{global}}), \quad \forall \ell \tag{24}$$

Summing over all layers:

$$\sum_{\ell=1}^{L} \mathcal{E}^{(\ell)}(\alpha^{(\ell)*}) \leq \sum_{\ell=1}^{L} \mathcal{E}^{(\ell)}(\alpha_{\text{global}}) \tag{25}$$

with strict inequality when at least one layer has $\alpha^{(\ell)*} \neq \alpha_{\text{global}}$. This demonstrates that layer-wise adaptation provably reduces the total quantization error.

Since Eq. (20) does not admit a closed-form solution, we resort to numerical optimization. Algorithm 1 summarizes the complete layer-wise SmoothQuant procedure. Specifically, after collecting layer-wise activation and weight statistics during the standard calibration phase, Algorithm 1 solves a one-dimensional optimization problem for each layer ℓ to obtain its optimal migration strength. The resulting layer-specific migration strengths are then used to compute the smoothing factors in Eq. (18), which are folded into the quantized weights in the same manner as in the original SmoothQuant, introducing no additional runtime overhead.

Algorithm 1 Layer-wise Adaptive SmoothQuant

Require: Pre-trained model $\mathcal{M}$, calibration dataset $\mathcal{D}_{\text{calib}}$
Ensure: Quantized model $\hat{\mathcal{M}}$
// Step 1: Collect statistics
for each layer $\ell = 1, \ldots, L$ **do**
  Compute $X_{\max,j}^{(\ell)}, W_{\max,j}^{(\ell)}, v_{X,j}^{(\ell)}, v_{W,j}^{(\ell)}$ on $\mathcal{D}_{\text{calib}}$
end for
// Step 2: Compute optimal α per layer
for each layer $\ell = 1, \ldots, L$ **do**
  Solve $\alpha^{(\ell)*}$ via Eq. (23)
end for
// Step 3: Apply layer-wise smoothing
for each layer $\ell = 1, \ldots, L$ **do**
  $s_j^{(\ell)} = (X_{\max,j}^{(\ell)})^{\alpha^{(\ell)*}} / (W_{\max,j}^{(\ell)})^{1-\alpha^{(\ell)*}}$
  $\mathbf{X}'^{(\ell)} = \mathbf{X}^{(\ell)} \operatorname{diag}(\mathbf{s}^{(\ell)})^{-1}$, $\mathbf{W}'^{(\ell)} = \operatorname{diag}(\mathbf{s}^{(\ell)}) \mathbf{W}^{(\ell)}$
end for
// Step 4: Quantize to INT8
Quantize $\mathbf{X}'^{(\ell)}$ and $\mathbf{W}'^{(\ell)}$ for all layers
return $\hat{\mathcal{M}}$

We analyze the additional cost of layer-wise optimization compared to the original SmoothQuant. Both methods require the same calibration phase to collect activation statistics. The key difference lies in Step 2 of Algorithm 1:

- **SmoothQuant**: Uses a fixed $\alpha = 0.5$ for all layers, requiring no optimization.
- **Ours**: Solves Eq. (23) independently for each layer using the Brent method [13], a derivative-free one-dimensional optimization algorithm with guaranteed convergence on bounded intervals.

In our case, the objective $\mathcal{E}^{(\ell)}(\alpha)$ is a smooth scalar function defined on the compact interval $\alpha \in [0, 1]$, making it well-suited for Brent's method. The per-layer optimization cost is $O(d \cdot \log(1/\epsilon))$, where d is the channel dimension and ϵ denotes the numerical tolerance used to terminate the one-dimensional search on α. For a model with L layers, the total additional cost is:

$$\text{Additional cost} = O(L \cdot d \cdot \log(1/\epsilon)). \tag{26}$$

Compared to the calibration cost $O(L \cdot N \cdot d \cdot m)$, where N is the number of calibration samples and m is the output dimension, the additional optimization overhead is negligible, since $\log(1/\epsilon) \ll N \cdot m$ in practice. Therefore, the layer-wise adaptation introduces only marginal offline overhead.

IV. Experiments

We validate our theoretical analysis and demonstrate the effectiveness of layer-wise adaptive SmoothQuant through comprehensive experiments. Our evaluation addresses three key hypotheses:

- **H1 (Layer Heterogeneity)**: Different layers exhibit distinct empirically preferred α values, demonstrating that a uniform smoothing factor is suboptimal.(Section IV-B).
- **H2 (Theoretical Foundation)**: The derived quantization error function captures the α-dependent trade-off between activation and weight quantization errors, and exhibits layer-wise trends consistent with empirical observations. (Section IV-C).
- **H3 (Effectiveness)**: Layer-wise adaptation informed by the theoretical analysis reduces quantization error and improves downstream task performance compared to uniform smoothing.(Section IV-D).

A. Experimental Setup

a) Models and datasets: We evaluate on the Llama family of models [14], with parameter sizes ranging from 1B to 13B. For calibration, we use 512 random samples from the Pile validation set [15]. We evaluate zero-shot task performance on seven benchmarks: LAMBADA [16], HellaSwag [17], PIQA [18], WinoGrande [19], OpenBookQA [20], RTE [21], and COPA [22].

b) Baselines: We compare against:

- **SmoothQuant (global α=0.5)**: Original method with fixed $\alpha = 0.5$ for all layers [9].
- **Ours (layer-wise α)**: Our method with per-layer $\alpha^{(\ell)*}$ from Eq. (23).

c) Implementation details: We use per-token activation quantization and per-channel weight quantization. All experiments use INT8 quantization with symmetric uniform quantization. For layer-wise optimization, we use the Brent method with tolerance $\epsilon = 10^{-4}$ and bounds $[0.0, 1.0]$.

B. H1: Layer Heterogeneity Analysis

We first empirically examine whether different layers exhibit heterogeneous activation and weight characteristics that may affect quantization behavior. Fig 2 presents statistics collected from the LLaMA-3-1B model on the calibration dataset. Fig 2(a) shows the layer-wise outlier magnitude across layers 0–15, where the magnitude is measured as the maximum absolute value. We observe that activation outliers exhibit significantly larger magnitudes than weights across all layers. Moreover, activation magnitudes vary substantially across layers, with fluctuations spanning multiple levels, whereas weight magnitudes remain relatively small and stable throughout the network. This pronounced imbalance and variability indicate that different layers face distinct activation quantization challenges.

To examine how such heterogeneity manifests in practical quantization behavior, we evaluate the empirical output error under different smoothing factors α. Specifically, we compute the normalized quantization error as

$$\mathcal{E}(\alpha) = \frac{\|Y_{\mathrm{FP16}} - Y_{\mathrm{INT8}}(\alpha)\|_F^2}{\|Y_{\mathrm{FP16}}\|_F^2}, \tag{27}$$

where Y_{FP16} and $Y_{\mathrm{INT8}}(\alpha)$ denote the layer outputs before and after W8A8 quantization, respectively.

Fig. 3 reports the empirical error curves for LLaMA models of different scales (1B, 7B, 8B, and 13B). For each model, five layers are uniformly sampled to cover early, middle, and late stages of the network. Across all models and layers, the empirical quantization error consistently exhibits a U-shaped dependence on α, indicating a trade-off between activation and weight quantization errors. While the error-minimizing α values generally concentrate in the range of 0.4–0.6, consistent with prior observations in SmoothQuant, the optimal α varies noticeably across layers. These results demonstrate that a single global migration strength cannot simultaneously minimize quantization error for all layers, motivating the need for layer-wise adaptation.

C. H2: Validation of Theoretical Error Function

We next evaluate whether the derived theoretical error function can explain the empirical dependence of quantization error on the smoothing factor α. Fig. 4 compares the theoretical error curves with the corresponding empirical output errors for selected layers, same as those layers in Fig. 3.

Fig. 4(a–e) show results for the five layers of the LLaMA-1B model. For layers (0,4,12), the theoretical error exhibits a clear U-shaped dependence on α, and the locations of the minima closely align with those observed in empirical measurements. This indicates that, for these layers, the derived error function successfully captures the dominant trade-off between activation and weight quantization errors.

In contrast, for layers (8) and (15), the theoretical error becomes monotonic over the $\alpha \in [0, 1]$ range, while the empirical error remains U-shaped. These layers exhibits a monotonically decreasing trend, indicating activation-dominated behavior. The mismatch between theoretical and empirical minima in these cases reflects the simplified assumptions of the theoretical model, which does not account for factors such as clipping effects, residual accumulation, and non-Gaussian activation distributions. Nevertheless, the monotonic trends still provide informative guidance on the relative sensitivity of each layer to activation versus weight quantization. Fig. 4(f–j) present the same comparison for the LLaMA-13B model.

Overall, these results demonstrate that while the theoretical error function does not always exactly predict the empirical optimum, it consistently captures the α-dependent trade-off structure and provides meaningful, layer-specific guidance for selecting migration strengths.

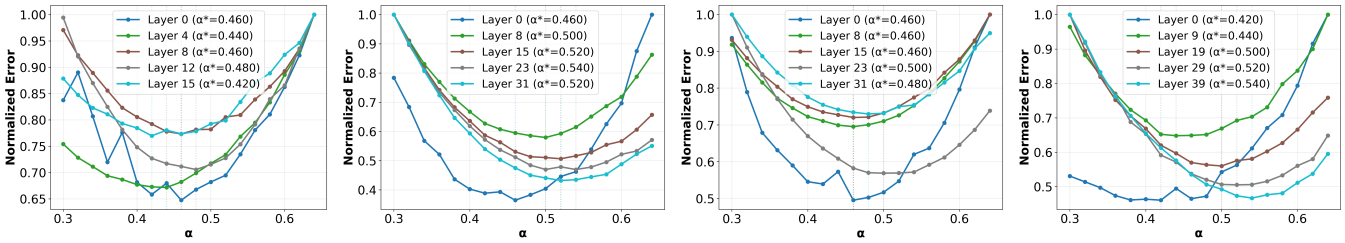
D. H3: Effectiveness of Layer-wise Smoothing

We finally evaluate whether the proposed layer-wise smoothing strategy leads to practical benefits in terms of activation stability and downstream task performance.

Fig. 5 reports the layer-wise coefficient of variation (CV) of activations for LLaMA models of different scales.

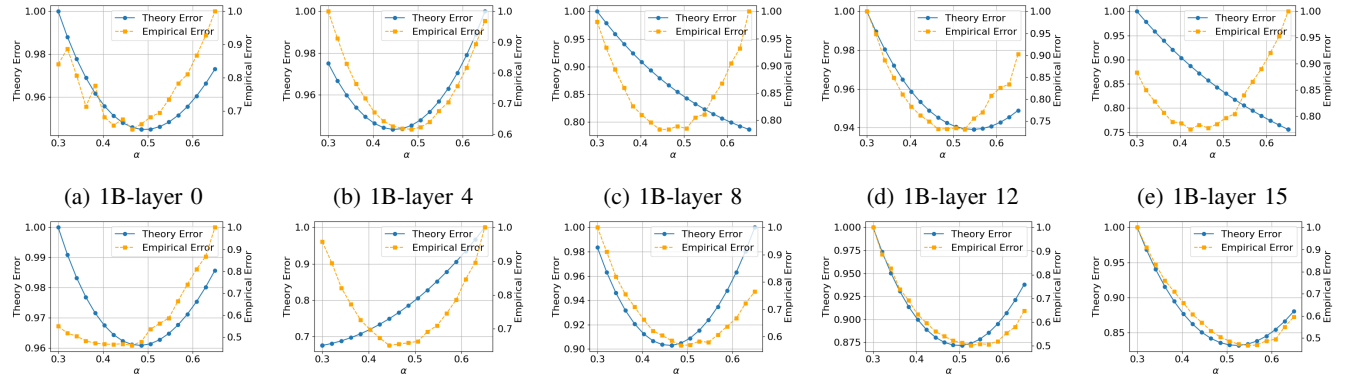
$$\mathrm{CV}^{(\ell)} = \frac{\mathrm{std}(\{X_{\max,j}^{(\ell)}\}_{j=1}^{d})}{\mathrm{mean}(\{X_{\max,j}^{(\ell)}\}_{j=1}^{d})}. \tag{28}$$

We compare the original model, SmoothQuant with a global migration strength ($\alpha = 0.5$), and the proposed layer-wise



(a) 1B (b) 7B (c) 8B (d) 13B

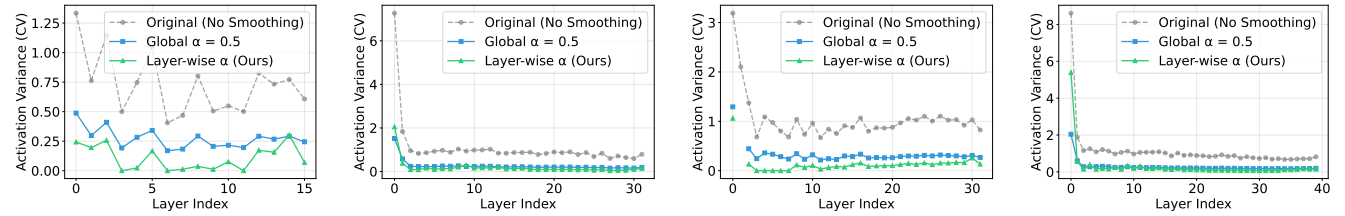
Fig. 3: Empirical quantization error as a function of the migration strength α for LLaMA models of different scales (1B, 7B, 8B, and 13B). For each model, five layers are uniformly sampled from early, middle, and late stages. Across all models and layers, the error exhibits a clear U-shaped dependence on α, reflecting the trade-off between activation and weight quantization errors. While the error-minimizing α typically lies around 0.4–0.6, its exact value varies across layers.



(a) 1B-layer 0 (b) 1B-layer 4 (c) 1B-layer 8 (d) 1B-layer 12 (e) 1B-layer 15

(f) 13B-layer 0 (g) 13B-layer 9 (h) 13B-layer 19 (i) 13B-layer 29 (j) 13B-layer 39

Fig. 4: Derived theoretical error VS empirical error



(a) 1B (b) 7B (c) 8B (d) 13B

Fig. 5: optimal alpha for different layers to minimize layer output error

smoothing strategy. For the LLaMA-1B model, the original activations exhibit substantial layer-wise variability, indicating strong heterogeneity. While global smoothing already reduces activation variance, the layer-wise strategy consistently achieves further reduction across layers, resulting in the lowest overall variability. This suggests that fine-grained, layer-specific migration strengths are particularly beneficial for smaller models where activation statistics vary significantly across layers.

For larger models (7B, 8B, and 13B), the original activation variance is already more stable across layers, and global smoothing effectively suppresses most variability. In these cases, the additional improvement from layer-wise smoothing is more modest, indicating that global smoothing captures much of the benefit when activation statistics are relatively homogeneous. Nevertheless, layer-wise smoothing consistently achieves equal or lower activation variance than global smoothing across all models.

Table II summarizes downstream task accuracy under W8A8 quantization. Although improvements vary across individual tasks, layer-wise smoothing consistently achieves higher average accuracy compared to uniform smoothing for all evaluated models. These results demonstrate that the proposed layer-wise strategy effectively translates the theoretical insights into practical performance improvements.

TABLE II: Zero-shot accuracy comparison across models with uniform vs. layer-wise α in SmoothQuant. Layer-wise optimal α consistently improves or maintains average accuracy compared to uniform $\alpha = 0.5$.

Model	**Method**	LAMBADA	HellaSwag	PIQA	WinoGrande	OpenBookQA	RTE	COPA	**Average↑**
1B	Uniform α	74.60%	47.00%	74.40%	57.70%	36.00%	52.35%	67.00%	58.44%
	Layer-wise α	73.70%	46.50%	74.10%	56.70%	36.80%	52.35%	69.00%	**58.45%**
7B	Uniform α	88.10%	53.20%	76.80%	61.50%	45.40%	54.51%	65.00%	63.50%
	Layer-wise α	87.80%	55.70%	77.20%	61.70%	45.00%	54.51%	70.00%	**64.56%**
8B	Uniform α	81.40%	57.10%	78.40%	69.70%	42.60%	52.71%	73.00%	64.99%
	Layer-wise α	81.00%	57.80%	78.00%	69.90%	43.60%	52.35%	73.00%	**65.09%**
13B	Uniform α	88.50%	56.90%	78.60%	62.80%	46.00%	52.71%	66.00%	64.50%
	Layer-wise α	89.00%	56.80%	78.80%	62.50%	45.80%	53.07%	70.00%	**65.14%**

V. Conclusion

In this paper, we present a theoretical analysis of SmoothQuant and identify the conditions under which its global scaling strategy is suboptimal. By formulating the quantization error minimization problem, we derive the per-channel optimal scaling and show that SmoothQuant can be viewed as a constrained approximation of this solution. Relaxing the global constraint leads naturally to a layer-wise scaling strategy that better accounts for inter-layer heterogeneity. Extensive experiments across different model sizes demonstrate that layer-wise scaling consistently reduces activation variance and quantization error, translating into improved downstream performance. Overall, our work bridges the gap between empirical heuristics and theoretical optimality in post-training quantization.

References

[1] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2704–2713.

[2] R. Banner, Y. Nahshan, and D. Soudry, "Post training 4-bit quantization of convolutional networks for rapid-deployment," *Advances in neural information processing systems*, vol. 32, 2019.

[3] Z. Liu, J. Li, C.-W. Sham, S. L. Ma, and C. Fu, "Lightfsa: A lightweight financial sentiment analysis model," in *Advanced Intelligent Computing Technology and Applications*. Singapore: Springer Nature Singapore, 2025, pp. 332–343.

[4] T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer, "Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale," *Advances in neural information processing systems*, vol. 35, pp. 30 318–30 332, 2022.

[5] Z. Yao, R. Yazdani Aminabadi, M. Zhang, X. Wu, C. Li, and Y. He, "Zeroquant: Efficient and affordable post-training quantization for large-scale transformers," *Advances in neural information processing systems*, vol. 35, pp. 27 168–27 183, 2022.

[6] Y. Bondarenko, M. Nagel, and T. Blankevoort, "Understanding and overcoming the challenges of efficient transformer quantization," *arXiv preprint arXiv:2109.12948*, 2021.

[7] Chiu-Wing Sham, Wai-Chiu Wong, and E. R. Y. Young, "Congestion estimation with buffer planning in floorplan design," in *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, March 2002, pp. 696–701.

[8] C. W. Sham and E. F. Y. Young, "Congestion prediction in early stages of physical design," *Acm Transactions on Design Automation of Electronic Systems*, vol. 14, no. 1, pp. 1–18, 2009.

[9] G. Xiao, J. Lin, M. Seznec, H. Wu, J. Demouth, and S. Han, "Smoothquant: Accurate and efficient post-training quantization for large language models," in *International conference on machine learning*. PMLR, 2023, pp. 38 087–38 099.

[10] J. Pan, C. Wang, K. Zheng, Y. Li, Z. Wang, and B. Feng, "Smoothquant+: Accurate and efficient 4-bit post-training weightquantization for llm," *arXiv preprint arXiv:2312.03788*, 2023.

[11] E. Frantar, S. Ashkboos, T. Hoefler, and D. Alistarh, "Gptq: Accurate post-training quantization for generative pre-trained transformers," *arXiv preprint arXiv:2210.17323*, 2022.

[12] J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, "Awq: Activation-aware weight quantization for on-device llm compression and acceleration," *Proceedings of machine learning and systems*, vol. 6, pp. 87–100, 2024.

[13] R. P. Brent, *Algorithms for minimization without derivatives*. Courier Corporation, 2013.

[14] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[15] L. Gao, S. Biderman, S. Black, L. Golding, T. Hoppe, C. Foster, J. Phang, H. He, A. Thite, N. Nabeshima *et al.*, "The pile: An 800gb dataset of diverse text for language modeling," *arXiv preprint arXiv:2101.00027*, 2020.

[16] D. Paperno, G. Kruszewski, A. Lazaridou, N.-Q. Pham, R. Bernardi, S. Pezzelle, M. Baroni, G. Boleda, and R. Fernández, "The lambada dataset: Word prediction requiring a broad discourse context," in *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, 2016, pp. 1525–1534.

[17] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi, "Hellaswag: Can a machine really finish your sentence?" *arXiv preprint arXiv:1905.07830*, 2019.

[18] Y. Bisk, R. Zellers, J. Gao, Y. Choi *et al.*, "Piqa: Reasoning about physical commonsense in natural language," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 7432–7439.

[19] K. Sakaguchi, R. L. Bras, C. Bhagavatula, and Y. Choi, "Winogrande: An adversarial winograd schema challenge at scale," *Communications of the ACM*, vol. 64, no. 9, pp. 99–106, 2021.

[20] T. Mihaylov, P. Clark, T. Khot, and A. Sabharwal, "Can a suit of armor conduct electricity? a new dataset for open book question answering," *arXiv preprint arXiv:1809.02789*, 2018.

[21] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," in *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, 2018, pp. 353–355.

[22] M. Roemmele, C. A. Bejan, and A. S. Gordon, "Choice of plausible alternatives: An evaluation of commonsense causal reasoning." in *AAAI spring symposium: logical formalizations of commonsense reasoning*, 2011, pp. 90–95.