# NOT ALL TOKENS ARE EQUAL: RETHINKING QUANTIZATION FOR DIFFUSION LANGUAGE MODELS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Quantization is a standard approach for large language models' deployment. Most existing quantization methods for large language models are designed for autoregressive models and the quantization objective is based on layer-wise mean square reconstruction error. In this paper, we show that such objective is mismatched to diffusion language models. Unlike autoregressive models, diffusion language models distinguish between visible tokens, which provide conditioning context, and masked tokens, which are the actual prediction targets. This structural asymmetry induces a fundamentally different quantization objective. Our analysis shows that minimizing the standard layer-wise mean square reconstruction error can yield a loose bound for quantization error in diffusion language models. Motivated by the above analysis, we explicitly disentangle visible and masked tokens as two groups and propose MV-Quant, which minimizes the worst-group reconstruction error. We further prove that, for an $n$-layer transformer, MV-Quant yields a tighter quantization error bound than the standard MSE objective with the same error tolerance. Extensive experiments across multiple diffusion language models and multiple benchmarks validate the effectiveness of our proposed MV-Quant.

## 1 INTRODUCTION

Large language models (LLMs) have made breakthrough progress in many fields but their deployment remains challenging due to substantial memory and computational costs. Quantization has therefore become a standard technique for reducing these costs. Most existing LLM quantization algorithms (Lin et al., 2024; Frantar et al., 2023; Xiao et al., 2023) adopted in current models (Yang et al., 2025; Dubey et al., 2024; Liu et al., 2024a) are based on layer-wise reconstruction, where the quantized weights are selected to minimize the discrepancy between the activations produced by the quantized and full-precision weights:

$$\arg\min_{\widehat{W}} \|WX - \widehat{W}X\|_F^2, \tag{1}$$

where $X$ is the input, $W$ and $\widehat{W}$ are the full-precision and quantized weights, respectively. This objective treats all tokens equally, which is well aligned with the autoregressive prediction structure, where the model predicts at every position and each token is used to predict the next token.

Recently, diffusion language models (DLMs) (Nie et al., 2025; Ye et al., 2025; Zhu et al., 2025; Khanna et al., 2025; Wu et al., 2026; Cheng et al., 2025) have emerged as a compelling alternative to autoregressive generation (Yang et al., 2025; Dubey et al., 2024; Achiam et al., 2023), achieving competitive quality while supporting parallel decoding and flexible any-order generation. The rapid progress of DLMs makes their efficient deployment an increasingly pressing problem, and quantization is a natural candidate.

Existing quantization algorithms (Xu and Yang, 2025; Zhang et al., 2025) for DLMs share the same mean square objective as ARM's quantization methods (such as AWQ (Lin et al., 2024) and GPTQ (Frantar et al., 2023)). However, DLMs differ from autoregressive models in a way that is crucial for quantization: at every denoising step, the input sequence consists of two structurally distinct token groups, namely *visible tokens* that provide conditioning context and *masked tokens* that serve as prediction targets. These two token groups contribute differently to the final quantization error. In particular, visible-token activations influence generation indirectly through conditioning



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and attention, whereas masked-token activations directly affect the denoising output. Consequently, quantization errors on the two groups contribute unevenly to the final output error, and the standard objective in (1), which assigns uniform importance to every token, is no longer well aligned with the true quantization goal. This raises a fundamental question:

*How to design a quantization objective that is better aligned with DLM?*

To address this question, we first develop a detailed quantization error analysis for DLMs. We show that the output error of a quantized DLM layer depends asymmetrically on two sources: reconstruction error on masked-token activations and reconstruction error on visible-token activations. These two terms enter the error propagation with different coefficients whose relative magnitudes vary across layers and samples. As a result, minimizing the MSE reconstruction objective in (1) can be a poor surrogate for the true quantization objective.

Motivated by this mismatch, we propose Masked–Visible Quantization (MV-Quant), a new objective for DLM quantization. Instead of averaging reconstruction errors over all token positions, MV-Quant disentangles the reconstruction errors on visible and masked tokens and minimizes the worst-group error. Intuitively, since the error coefficients of the two groups are unknown and input-dependent, minimizing their average treats them equally and risks leaving one group with a large error. If this group receives a large coefficient in the final error bound, the resulting quantization error can be substantially large. Minimizing the worst-group error avoids this by keeping the errors of both groups small simultaneously, ensuring robustness regardless of which coefficient dominates. We further theoretically prove that MV-Quant yields a tighter upper bound on the quantization error than the standard MSE loss with the same error tolerance. Building on this objective, we develop a diffusion-aware quantization framework based on AWQ (Lin et al., 2024) and evaluate it on multiple DLMs and benchmark tasks.

Our contributions are summarized as follows:

- We provide a detailed quantization error analysis for DLMs, showing that the visible and masked token groups contribute differently to the quantization error and that the standard layer-wise MSE objective yields a loose bound on the final quantization error.
- We propose MV-Quant, a quantization objective that disentangles visible-token and masked-token reconstruction and explicitly controls the worst-group error, with a theoretical guarantee that it achieves a tighter error bound than the standard MSE objective.
- We conduct extensive experiments on multiple DLMs and benchmarks, demonstrating that MV-Quant reduces quantization error and improves downstream performance over existing methods.

## 2 Preliminary

### 2.1 Diffusion Language Models

Diffusion models have demonstrated remarkable success in continuous domains (Sohl-Dickstein et al., 2015; Ho et al., 2020; Karras et al., 2022). Building on these foundations, diffusion language models have been developed to train discrete diffusion models directly within the categorical vocabulary space (Sohl-Dickstein et al., 2015; Austin et al., 2021a; Lou et al., 2023; Nie et al., 2025; Ye et al., 2025). Given an original data sample $\boldsymbol{x}^0 \in \mathbb{R}^N$, the forward process progressively corrupts $\boldsymbol{x}^0$ to a sequence of latent variables $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^T$ (with each $\boldsymbol{x}^t \in \mathbb{R}^N$) by iteratively applying noise. The corresponding joint distribution is:

$$q(\boldsymbol{x}^{1:T}) = q(\boldsymbol{x}^0) \prod_{t=1}^{T} q(\boldsymbol{x}^t \mid \boldsymbol{x}^{t-1}),$$

where $q(\boldsymbol{x}^t \mid \boldsymbol{x}^s) = \mathrm{Cat}(\boldsymbol{x}^t; \mathbf{Q}_t \boldsymbol{x}^s)$ is the categorical distribution and $\mathbf{Q}_t$ is the transition matrix at time step $t$. On the other hand, the backward process learns to gradually denoise the masked sequence back to the original data distribution by iteratively predicting masked tokens as $t$ moves from $T$ to 0, with the corresponding joint distribution:

$$p_{\boldsymbol{\theta}}(\boldsymbol{x}^{0:T}) \coloneqq p(\boldsymbol{x}_T) \prod_{t=1}^{T} p_{\boldsymbol{\theta}}(\boldsymbol{x}^{t-1} | \boldsymbol{x}^t).$$

The model parameter $\boldsymbol{\theta}$ is optimized by minimizing the negative log-likelihood of the clean data $\boldsymbol{x}^0$. When the model uses the absorbing kernel, learning the denoising process leads to minimizing the Negative Evidence Lower Bound (NELBO) of log likelihood (Nie et al., 2025):

$$\mathcal{L}(\boldsymbol{\theta}) = -\mathbb{E}_{\boldsymbol{x}^0,\boldsymbol{\epsilon},t} \sum_{i=1}^{n} \mathbf{1}_{[\boldsymbol{x}^0[i]=M]} \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^0[i] \mid \boldsymbol{x}^t), \tag{2}$$

where $\boldsymbol{x}^0[i]$ is the $i$th element of $\boldsymbol{x}^0$ and $M$ denotes the mask token.

Recent research typically adopts the absorbing kernel strategy, which maps input tokens to a mask token during the forward pass, and has focused on scaling masked diffusion models to the billion-parameter regime (Nie et al., 2025; Ye et al., 2025; Khanna et al., 2025; Zhu et al., 2025). This line of work can be categorized by their training and initialization strategies: (i) Training from scratch: The LLaDA series (Nie et al., 2025; Zhu et al., 2025) trains diffusion models from scratch, utilizing direct mask prediction objectives. (ii) AR Initialization: Conversely, the Dream series (Xie et al., 2025; Ye et al., 2025), SDAR (Cheng et al., 2025), and Fast-dLLM v2 (Wu et al., 2026) leverage pre-trained autoregressive models (ARMs) for initialization. These models are fine-tuned as diffusion models using a shift mask prediction objective.

### 2.2 LLM Quantization

Quantization is a standard approach to reduce the memory and computation costs of large language models. Quantization-aware training (QAT) (Chen et al., 2025; Liu et al., 2024b) incorporates quantization directly into the training process and can lead to accurate quantized models. However, it is often impractical due to its high training cost. On the other hand, post-training quantization (PTQ), which quantizes a fully trained model without retraining or fine-tuning, is more commonly used. PTQ methods for autoregressive transformers include second-order or Hessian-inspired methods (e.g., GPTQ (Frantar et al., 2023)), activation-aware schemes (e.g., AWQ (Lin et al., 2024)), activation-smoothing methods (e.g., SmoothQuant (Xiao et al., 2023)), and calibration-based methods (e.g., OmniQuant (Shao et al., 2023)). In these methods, the objective is defined over the full sequence processed by the model, and the quantization quality is assessed through layer-wise or block-wise reconstruction of activations. Recently, PTQ is also adapted to DLLMs. DLLMQuant (Xu and Yang, 2025) uses a temporal-mask adaptive calibration dataset and certainty-guided weight quantization. Quant-dLLM (Zhang et al., 2025) considers a distinct mixed-precision quantization setting with an average 2-bit budget.

### 2.3 Transformer

Consider an $n$-layer transformer. Let $H^{(0)} \in \mathbb{R}^{T\times d}$ be the embedding of an input sequence $X \in \mathbb{R}^{T\times d}$ with length $T$ and dimension $d$. For each layer $l \in \{1,\dots,n\}$, the hidden states are updated via a mapping of the form: $H^{(l)} = \mathcal{F}^{(l)}(H^{(l-1)}) + H^{(l-1)}$. At layer $l$, the query, key, and value matrices are computed as linear projections of $H^{(l-1)}$ (Huang et al., 2024; Li et al., 2024; Vaswani et al., 2017):

$$Q^{(l)} = H^{(l-1)}W_Q^{(l)}, \quad K^{(l)} = H^{(l-1)}W_K^{(l)}, \quad V^{(l)} = H^{(l-1)}W_V^{(l)},$$

where $W_Q^{(l)}, W_K^{(l)} \in \mathbb{R}^{d\times d_k}$ and $W_V^{(l)} \in \mathbb{R}^{d\times d_v}$. The attention probability matrix $P^{(l)} \in \mathbb{R}^{T\times T}$ is defined via the row-wise softmax of scaled dot-product scores:

$$P^{(l)} = \mathrm{Softmax}\left(\frac{Q^{(l)}(K^{(l)})^\top}{\sqrt{d_k}}\right). \tag{3}$$

Finally, by incorporating the output projection $W_O^{(l)} \in \mathbb{R}^{d_v\times d}$ and a residual connection, the layer update is: $H^{(l)} = H^{(l-1)} + P^{(l)}V^{(l)}W_O^{(l)}$.

## 3 Quantization Error Analysis for DLMs

We first analyze how layer-wise quantization errors propagate and accumulate through a $n$-layer transformer. At layer $l$, its input hidden state already contains the errors accumulated from previous

layers. Let $H^{(l)}$ and $\widehat{H}^{(l)}$ be the full-precision and quantized hidden states at layer $l$, respectively:

$$H^{(l)} = \mathcal{F}^{(l)}\left(H^{(l-1)}\right) + H^{(l-1)}; \quad \widehat{H}^{(l)} = \widehat{\mathcal{F}}^{(l)}\left(\widehat{H}^{(l-1)}\right) + \widehat{H}^{(l-1)},$$

where $\mathcal{F}^{(l)}$ (resp. $\widehat{\mathcal{F}}^{(l)}$) is the full-precision (resp. quantized) transformations at layer $l$. The quantization error at layer $l$ can be decomposed as:

$$\widehat{H}^{(l)} - H^{(l)} = \underbrace{\widehat{H}^{(l-1)} - H^{(l-1)}}_{\text{previous layer quant error}} + \underbrace{\widehat{\mathcal{F}}^{(l)}(\widehat{H}^{(l-1)}) - \widehat{\mathcal{F}}^{(l)}(H^{(l-1)})}_{\text{propagation of input error}} + \underbrace{\widehat{\mathcal{F}}^{(l)}(H^{(l-1)}) - \mathcal{F}^{(l)}(H^{(l-1)})}_{\text{new quantization error introduced at layer } l}. \tag{4}$$

The first term on the RHS corresponds to the quantization error in previous layer, the second term is on how the quantization error accumulated up to layer $l-1$ propagates to layer $l$, and the third term is on the new quantization error introduced at layer $l$.

Define the quantization error at layer $l$ as $E^{(l)} := \widehat{H}^{(l)} - H^{(l)}$, The following Proposition characterizes how $E^{(l)}$ propagates across transformer layers by applying first-order Taylor expansion on $\widehat{\mathcal{F}}^{(l)}$

**Proposition 3.1.** *With first-order Taylor approximation on $\widehat{\mathcal{F}}^{(l)}$, the quantization error at layer $l$ satisfies*

$$E^{(l)} = [I + \nabla_{H^{(l-1)}} \widehat{\mathcal{F}}^{(l)}(H^{(l-1)})]E^{(l-1)} + \widehat{\mathcal{F}}^{(l)}(H^{(l-1)}) - \mathcal{F}^{(l)}(H^{(l-1)}), \tag{5}$$

*where $I$ is the identity matrix.*

To simplify presentation, let $\xi^{(l)} := \widehat{\mathcal{F}}^{(l)}(H^{(l-1)}) - \mathcal{F}^{(l)}(H^{(l-1)})$ be the new quantization error introduced at layer $l$, and $T^{(l)} := [I + \nabla_{H^{(l-1)}} \widehat{\mathcal{F}}^{(l)}(H^{(l-1)})]$ be the error propagation matrix in (5).

For DLM, the error can be naturally decomposed into two groups, one for the masked tokens and the other for the visible tokens. Let $\mathcal{M}$ and $\mathcal{V}$ be the corresponding index sets, and $P_m \in \{0,1\}^{|\mathcal{M}| \times L}$ and $P_v \in \{0,1\}^{|\mathcal{V}| \times L}$ be the corresponding row selection matrices, where $(P_m)_{r,i} = \mathbf{1}\{i = \mathcal{M}_r\}$ and $(P_v)_{r,i} = \mathbf{1}\{i = \mathcal{V}_r\}$. Equation (5) can be equivalently written as:

$$\begin{bmatrix} E_m^{(l)} \\ E_v^{(l)} \end{bmatrix} = \begin{bmatrix} T_{mm}^{(l)} & T_{mv}^{(l)} \\ T_{vm}^{(l)} & T_{vv}^{(l)} \end{bmatrix} \begin{bmatrix} E_m^{(l-1)} \\ E_v^{(l-1)} \end{bmatrix} + \begin{bmatrix} \xi_m^{(l)} \\ \xi_v^{(l)} \end{bmatrix},$$

where $E_m^{(l)} = P_m E^{(l)}$, $E_v^{(l)} = P_v E^{(l)}$, $\xi_m^{(l)} = P_m \xi^{(l)}$, $\xi_v^{(l)} = P_v \xi^{(l)}$, $T_{mm}^{(l)} = P_m T^{(l)} P_m^\top$, $T_{mv}^{(l)} = P_m T^{(l)} P_v^\top$, $T_{vm}^{(l)} = P_v T^{(l)} P_m^\top$, and $T_{vv}^{(l)} = P_v T^{(l)} P_v^\top$. By recursion on (5), we obtain the following.

**Proposition 3.2.** *$E_m^{(n)} = \sum_{l=1}^{n} (A_l \xi_m^{(l)} + B_l \xi_v^{(l)})$, where $A_l = P_m(\prod_{j=l+1}^{n} T_j)P_m^\top$ and $B_l = P_m(\prod_{j=l+1}^{n} T_j)P_v^\top$ for $l < n$. Hence,*

$$\|E_m^{(n)}\|_F \le \sum_{l=1}^{n} \|A_l \xi_m^{(l)} + B_l \xi_v^{(l)}\|_F. \tag{6}$$

Because DLMs only make prediction on masked tokens, Proposition 3.2 shows that the final quantization error of an $n$-layer transformer is bounded by a sum of weighted layer-wise quantization errors in the hidden states of masked and visible tokens. Therefore, minimizing the quantization error introduced at each layer provides a principled way to control the accumulated quantization error. The following Theorem shows that the error contribution from layer $l$ is upper bounded by the quantization errors induced by the attention projections $\Theta := \{W_Q, W_K, W_V, W_O\}$.

**Theorem 3.1.** *For $i \in \{Q, K, V, O\}$, let $X_i^{(l)}$ be the inputs to the projection matrices $W_i^{(l)}$. With first-order Taylor approximation, the error contribution of layer-$l$ is upper bounded by:*

$$\left\| A_l \xi_m^{(l)} + B_l \xi_v^{(l)} \right\|_F \le \sum_{i \in \{Q,K,V,O\}} \alpha_i^{(l)} \left( (1 - \beta_i^{(l)}) \epsilon_{i,m}^{(l)} + \beta_i^{(l)} \epsilon_{i,v}^{(l)} \right) \triangleq \mathrm{UB}(\Theta^{(l)}, \alpha^{(l)}, \beta^{(l)}), \tag{7}$$

*where $\epsilon_{i,\tau}^{(l)} = \|X_{i,\tau}^{(l)} \Delta W_i^{(l)}\|_F$ is the quantization error for $W_i$ on the mask tokens ($\tau = m$) and visible tokens ($\tau = v$). respectively. $\alpha_i^{(l)} > 0$ and $\beta_i^{(l)} \in [0,1]$ represent the error propagation coefficient for projection matrix $i$ at layer $l$ and the group importance ratio between visible and mask tokens at layer $l$, respectively. The detailed definitions of $\alpha_i^{(l)}$ and $\beta_i^{(l)}$ can be found in Section C.*

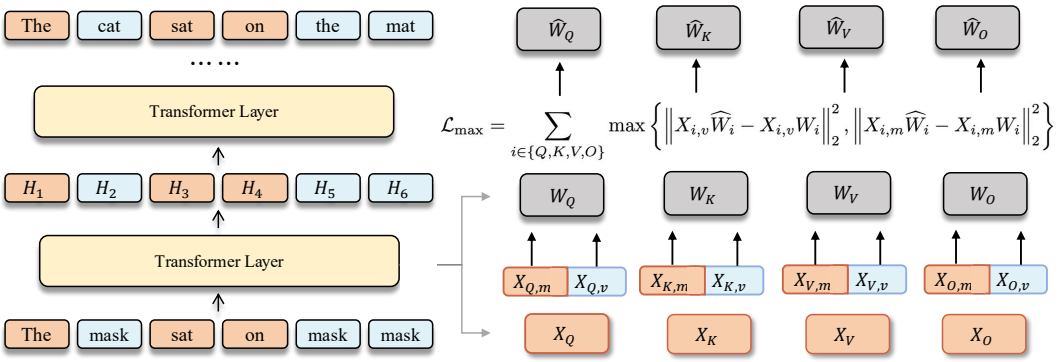


Figure 1: Overview of MV-Quant. For the projection matrices $W_Q$, $W_K$, $W_V$, and $W_O$, our method explicitly partitions the corresponding input activations into masked and visible token groups, and quantize the model by minimizing the worst-group reconstruction loss.

*Remark* 3.2. The coefficients $\beta_i^{(l)}$ are data and network dependent. Specifically, $\beta_Q^{(l)}$ depends on the norms of $A_l$, $B_l$, the key and value activations, and the softmax Jacobian blocks $\{\mathcal{J}_\tau^{(l),a\leftarrow b}\}_{\tau,a,b\in\{m,v\}}$; $\beta_K^{(l)}$ depends on the norms of $A_l$, $B_l$, the query and value activations, and the same Jacobian blocks; $\beta_V^{(l)}$ depends on the norms of $A_l$, $B_l$, and the attention-probability blocks $\{P_{\tau,b}^{(l)}\}_{\tau,b\in\{m,v\}}$; and $\beta_O^{(l)}$ depends only on the norms of $A_l$ and $B_l$.

At layer $l$, let $S_m^{(l)}$ (resp. $S_v^{(l)}$) be the attention score submatrix corresponding to the masked (resp. visible) token query rows attending to all key tokens. Let $\mathcal{J}_\tau^{(l),\,a\leftarrow b}$ (where $\tau, a, b$ denotes either the masked group $m$ or visible group $v$) be the corresponding block of the row-wise softmax Jacobian for query group $\tau$, characterizing how score perturbations associated with key group $b$ affect the attention probabilities assigned to key group $a$.

Theorem 3.1 shows that the relative importance assigned to mask and visible token quantization errors, denoted by $\beta^{(l)} = [\beta_i^{(l)}]_{i\in\{Q,K,V,O\}}$, differs significantly across $Q$, $K$, $V$, and $O$. These coefficients are determined by the corresponding activation and Jacobian. Consequently, as the model processes diverse inputs, the induced $\beta^{(l)}$ dynamically shifts across different sequences.

**Corollary 3.3.** *The quantization error of $n$-layer transformer in (6) can be upper-bounded as:*

$$\|E_m^{(n)}\|_F \le \sum_{l=1}^{n} \left\|A_l\xi_m^{(l)} + B_l\xi_v^{(l)}\right\|_F \le n\mathrm{UB}(\Theta^{(k)}, \alpha^{(k)}, \beta^{(k)}),$$

*where* $k = \arg\max_l\{\mathrm{UB}(\Theta^{(l)}, \alpha^{(l)}, \beta^{(l)})\}$.

Consequently, minimizing the standard MSE reconstruction error, which implicitly assigns a fixed and uniform importance to all token groups, is not necessarily aligned with the true quantization objective in diffusion language models. This mismatch becomes particularly severe when the true weighting places more emphasis on the token group with larger reconstruction error, leading to a substantially larger propagated quantization error.

## 4 Proposed MV-Quant

The standard MSE-based quantization objective obtains the quantized weights by minimizing the reconstruction error over all token activations:

$$\mathcal{L}_{\mathrm{MSE}} = \sum_{i\in\{Q,K,V,O\}} \left\|X_i\widehat{W}_i - X_iW_i\right\|_F^2, \tag{8}$$

which treats all token positions uniformly. While this is effective for autoregressive language models, the masked and visible tokens in diffusion language models play different roles during

denoising. Masked tokens are directly predicted by the model, whereas visible tokens mainly serve as conditioning context. Therefore, minimizing the averaged reconstruction error over all tokens may lead to large reconstruction errors on one token group.

Motivated by the analysis in Section 3, we propose to directly minimize the higher quantization error across the two token groups. This avoids relying on a fixed token-group weighting and instead directly controls the largest reconstruction error that may dominate the propagated quantization error. For each projection $i \in \{Q, K, V, O\}$, let $X_{i,m}$ and $X_{i,v}$ denote the masked-token and visible-token components of $X_i$, respectively. We propose the following MV-Quant objective:

$$\mathcal{L}_{\max} = \sum_{i \in \{Q,K,V,O\}} \max \left\{ \left\| X_{i,v}\widehat{W}_i - X_{i,v}W_i \right\|_F^2, \left\| X_{i,m}\widehat{W}_i - X_{i,m}W_i \right\|_F^2 \right\}. \tag{9}$$

This proposed objective can be interpreted as a commonly used Tchebycheff (min–max) scalarization in multi-objective optimization, and can also be interpreted as a robust surrogate for the coefficient-weighted objective when the relative importance of the two token groups is unknown or input-dependent. Compared with the standard MSE objective in (8), this prevents the quantization error from being dominated by either masked or visible tokens (Figure 1). Specifically, since $B_n$ is 0, the final-layer $W_Q$ quantized only on $X_m$, as predictions are made only for masked tokens rather than visible tokens.

The following Theorem shows that under the same per-group error tolerance, MV-Quant yields a tighter upper bound on the layer-wise quantization error than the standard MSE objective with the same error tolerance.

**Theorem 4.1.** *Let $I = \{Q, K, V, O\}$. For each projection $i \in I$, let $\epsilon_{i,m}^{(l)}$ and $\epsilon_{i,v}^{(l)}$ be the masked-token and visible-token reconstruction errors in layer $l$ defined in Theorem 3.1, respectively. Consider the layer-wise quantization-error upper bound in (7). With the same error tolerance $\epsilon$, the MSE-based objective and MV-Quant objective yield the upper bounds*

$$\left\| A_l \xi_m^{(l)} + B_l \xi_v^{(l)} \right\|_F \le \sqrt{2}\epsilon \sum_{i \in I} \alpha_i^{(l)} \sqrt{\left(1 - \beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2} := \mathrm{UB}_{\mathrm{mse}},$$

$$\left\| A_l \xi_m^{(l)} + B_l \xi_v^{(l)} \right\|_F \le \epsilon \sum_{i \in I} \alpha_i^{(l)} := \mathrm{UB}_{\max},$$

*where $\alpha_i^{(l)}, \beta_i^{(l)}$ are as defined in Theorem 3.1. Moreover, the upper bound induced by MV-Quant is no larger than that induced by MSE:*

$$\frac{\mathrm{UB}_{\max}}{\mathrm{UB}_{\mathrm{mse}}} = \frac{\epsilon \sum_{i \in I} \alpha_i^{(l)}}{\sqrt{2}\epsilon \sum_{i \in I} \alpha_i^{(l)} \sqrt{\left(1 - \beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2}} \le 1.$$

*The equality holds if and only if $\beta_i = \frac{1}{2}$ for all $i$'s such that $\alpha_i > 0$.*

Theorem 4.1 shows that MV-Quant provides a tighter upper bound than MSE loss. In particular, unless the masked and visible token contributions are exactly balanced, i.e., $\beta_i^{(l)} = \frac{1}{2}$ for all projections with nonzero $\alpha_i$, the inequality is strict. We denote the resulting ratio by $\kappa_l = (\sum_{i \in I} \alpha_i^{(l)}) / (\sqrt{2} \sum_{i \in I} \alpha_i^{(l)} \sqrt{\left(1 - \beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2})$. Thus, for a single transformer layer, the MV-Quant objective can have a strictly tighter quantization-error bound than the standard MSE objective by a factor $\kappa_l$. This supports the use of token-disentangled MV-Quant for DLM quantization, where controlling the worst-case error across masked and visible token groups is more aligned with the DLM objective. For an $n$-layer transformer, since its quantization error is upper bounded by the weighted sum of single layer quantization error, we have the following corollary.

**Corollary 4.2.** *Let $I = \{Q, K, V, O\}$. For each projection $i \in I$, let $\epsilon_{i,m}^{(l)}$ and $\epsilon_{i,v}^{(l)}$ denote the masked-token and visible-token reconstruction errors in layer $l$, respectively. Consider the layer-wise quantization-error upper bound in (6). With the same error tolerance $\epsilon$, the MSE-based objective and MV-Quant objective yield the upper bounds $\left\| E_m^{(n)} \right\|_F \le$*

Table 1: Fast-dLLM v2 7B 4-bit quantization results on various benchmarks. The best results are in bold, and the second-best are underlined. MV-Quant achieves the smallest accuracy drop and the highest average accuracy.

| | *GSM8K* | *Math* | *HumanEval* | *MBPP* | *GPQA* | *MMLU* | *IFEval* | Avg. |
|---|---|---|---|---|---|---|---|---|
| FP | 82.1 | 58.9 | 64.0 | 63.2 | 31.9 | 68.0 | 75.4 | 63.4 |
| RTN | 79.1 | 53.9 | 61.0 | 59.3 | 28.1 | 65.5 | 73.8 | 60.1 |
| GPTQ | 80.5 | 52.9 | 63.0 | <u>60.7</u> | 27.9 | 66.4 | 69.5 | 60.1 |
| AWQ | <u>80.6</u> | <u>55.7</u> | **64.7** | 59.0 | <u>31.9</u> | <u>67.0</u> | <u>73.9</u> | <u>61.8</u> |
| MV-Quant | **81.9** | **56.3** | <u>64.6</u> | **62.4** | **33.3** | **67.4** | **77.1** | **63.3** |

Table 2: SDAR 8B 4-bit quantization results on various benchmarks. The best results are in bold, and the second-best are underlined. MV-Quant achieves the smallest accuracy drop and the highest average accuracy.

| | *GSM8K* | *Math* | *HumanEval* | *MBPP* | *GPQA* | *MMLU* | *IFEval* | Avg. |
|---|---|---|---|---|---|---|---|---|
| FP | 89.1 | 67.7 | 77.4 | 74.9 | 36.9 | 71.7 | 69.4 | 69.6 |
| RTN | 86.7 | 61.5 | <u>70.7</u> | 72.0 | <u>33.3</u> | 61.9 | 62.5 | 64.1 |
| GPTQ | 86.9 | 60.3 | 67.1 | 67.7 | 32.3 | 67.8 | 62.9 | 63.6 |
| AWQ | <u>87.9</u> | <u>65.6</u> | 68.9 | **75.4** | 31.3 | <u>70.8</u> | <u>66.5</u> | <u>66.6</u> |
| MV-Quant | **88.0** | **66.3** | **73.2** | <u>73.8</u> | **35.4** | **71.7** | **69.3** | **68.2** |

$\sqrt{2}\,\epsilon \sum_{l=1}^{n} \sum_{i \in I} \alpha_i^{(l)} \sqrt{\left(1-\beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2} := \mathrm{UB}_{\mathrm{mse}}$ *and* $\left\|E_m^{(n)}\right\|_F \le \epsilon \sum_{l=1}^{n} \sum_{i \in I} \alpha_i^{(l)} := \mathrm{UB}_{\max}$. *Consequently, the upper bound induced by MV-Quant is no larger than the MSE upper bound:* $\mathrm{UB}_{\max}/\mathrm{UB}_{\mathrm{mse}} \le \max_{1 \le l \le n} \kappa_l \le 1$. *The equality* $\mathrm{UB}_{\max}/\mathrm{UB}_{\mathrm{mse}} = 1$ *holds if and only if* $\beta_i^{(l)} = \frac{1}{2}$ *for every* $l$ *and every* $i \in I$ *such that* $\alpha_i^{(l)} > 0$.

## 5 Experiments

In this section, we fist validate the effectiveness of the proposed MV-Quant. We first show that the MV-Quant objective yields a tighter error bound than the standard mean-squared-error objective. We then compare MV-Quant with existing quantization methods and demonstrate that it achieves state-of-the-art performance across diverse downstream tasks. Finally, we show that MV-Quant reduces memory consumption and improves inference throughput for current dLLMs.

### 5.1 Setup

**Models and Hardware.** Evaluation is performed on three representative DLM models: LLaDA-1.5 8B (Zhu et al., 2025), Fast-dLLM v2 7B (Wu et al., 2026), and SDAR 8B (Cheng et al., 2025). All experiments are run on NVIDIA A6000 GPUs.

**Benchmarks.** We evaluate the quantized models on a comprehensive suite of tasks, covering diverse aspects of language modeling and reasoning abilities. The evaluation suite includes (i) mathematical reasoning tasks: *GSM8K* (Cobbe et al., 2021) and *MATH* (Saxton and Hill, 2019), (ii) code generation tasks: *HumanEval* (Chen et al., 2021) and *MBPP* (Austin et al., 2021b), (iii) knowledge-intensive benchmarks: *GPQA* (Rein et al., 2024) and *MMLU* (Hendrycks et al., 2021), and (iv) instruction-following task *IFEval* (Zhou et al., 2023). Following standard evaluation protocols, we report accuracy for *GSM8K*, *MATH*, *GPQA*, *MMLU*; pass@1 for code generation benchmarks *HumanEval* and *MBPP*; and instruction-following accuracy for *IFEval*.

**Baselines.** We compare MV-Quant with the (i) full-precision model (FP), (ii) AWQ (Lin et al., 2024), (iii) GPTQ (Frantar et al., 2023), (iv) RTN (Dettmers et al., 2022), and (v) DLLMQuant (Xu

Table 3: LLaDA 1.5 8B 4-bit quantization results on various benchmarks. The best results are in bold, and the second-best are underlined. MV-Quant achieves the smallest accuracy drop and the highest average accuracy.

| | *GSM8K* | *Math* | *HumanEval* | *MBPP* | *GPQA* | *MMLU* | *IFEval* | Avg. |
|---|---|---|---|---|---|---|---|---|
| FP | 83.6 | 38.1 | 51.2 | 38.8 | 34.8 | 63.6 | 64.6 | 53.5 |
| RTN | 79.2 | 36.6 | 42.6 | 35.4 | 25.8 | 22.9 | 62.2 | 43.5 |
| GPTQ | 77.7 | 36.2 | 43.2 | 36.4 | <u>29.8</u> | **63.7** | 63.9 | 50.1 |
| AWQ | <u>80.2</u> | <u>37.0</u> | 44.0 | <u>37.6</u> | **30.8** | 63.1 | **64.7** | <u>51.1</u> |
| DLLMQuant | 74.3 | – | <u>44.6</u> | – | – | 57.3 | – | – |
| MV-Quant | **80.4** | **37.2** | **45.1** | **38.4** | **30.8** | <u>63.5</u> | <u>64.3</u> | **51.4** |

Table 4: SDAR 8B 3-bit quantization results on various benchmarks. The best results are in bold, and the second-best are underlined. MV-Quant achieves the smallest accuracy drop and the highest average accuracy.

| | *GSM8K* | *Math* | *HumanEval* | *MBPP* | *GPQA* | *MMLU* | *IFEval* | Avg. |
|---|---|---|---|---|---|---|---|---|
| FP | 89.1 | 67.7 | 77.4 | 74.9 | 36.9 | 71.7 | 69.4 | 69.6 |
| RTN | 8.3 | 4.6 | 34.2 | 39.4 | 23.7 | 31.0 | 22.5 | 23.4 |
| GPTQ | 80.2 | 47.3 | 51.8 | 58.5 | **33.8** | <u>65.3</u> | <u>57.7</u> | 56.4 |
| AWQ | <u>80.4</u> | <u>54.9</u> | <u>62.8</u> | <u>64.6</u> | <u>27.8</u> | 61.1 | 57.6 | <u>58.5</u> |
| MV-Quant | **83.1** | **55.9** | **64.6** | **67.2** | 26.8 | **66.6** | **59.4** | **60.5** |

and Yang, 2025). For a fair comparison, following Lin et al. (2024), we use pileval[1] as the calibration dataset. Since DLLMQuant does not release its code, we directly report the results from their paper for comparison. For the other baselines, we quantize the models using their official github codes. Compared with AWQ, MV-Quant only replaces the original MSE reconstruction objective with our proposed objective, while keeping all other components, calibration data, quantization settings, and optimization procedures unchanged. Quant-dLLM (Zhang et al., 2025) considers a distinct mixed-precision quantization setting with an average 2-bit budget and is therefore not directly comparable to our setting.

### 5.2 Test Accuracy

First, we evaluate the most commonly used INT4 quantization setting (Lin et al., 2024; Frantar et al., 2023). The 4-bit quantization results on Fast-dLLMv2-7B, SDAR-8B and LLaDA1.5-8B, are shown in Table 1, Table 2, and Table 3, respectively. As can be seen, MV-Quant achieves minimal accuracy loss and the highest average accuracy among the quantized methods across all three models. On Fast-dLLM v2 7B, SDAR-8B, and LLaDA-1.5-8B, MV-Quant achieves average accuracies of 63.3, 68.2, and 51.4 after 4-bit quantization, compared to the corresponding full-precision accuracies of 63.4, 69.6, and 53.5, resulting in small drops of only 0.1, 1.4, and 2.1 points, respectively.

Since the optimization objective of MV-Quant provides a tighter bound on the quantization error, MV-Quant improves the average accuracy over the strongest AWQ baseline by $+1.5$, $+1.6$, and $+0.3$ points on Fast-dLLM v2 7B, SDAR-8B, and LLaDA-1.5-8B, respectively. These results show that MV-Quant consistently preserves the accuracy of diffusion language models under aggressive 4-bit quantization while outperforming existing post-training quantization baselines.

Next, we evaluate 3-bit quantization. As 3-bit quantization lacks native Tensor Core support (NVIDIA, 2020) and is not commonly used in real-world deployment, we only run it on SDAR-8B. Table 4 shows the results. As can be seen, MV-Quant again outperforms the baselines, and improves over the strongest AWQ baseline by $+2.0$ points on average.

To validate whether MV-Quant reduces the quantization error as studied in our analysis, we measure the 4-bit quantization errors of both AWQ (Lin et al., 2024) and MV-Quant on Fast-dLLM v2 7B. Specifically, we replace the standard MSE reconstruction loss in AWQ with our proposed objective,

[1] https://huggingface.co/datasets/mit-han-lab/pile-val-backup

Table 5: Memory consumption and throughput comparison between full-precision and quantized models on Fast-dLLM v2 and SDAR using *GSM8K* and *HumanEval*. Memory is measured in MiB, and throughput is measured in tokens per second (TPS).

| Model | Dataset | Precision | Memory (MiB) | TPS | Speedup | Mem. Red. |
|---|---|---|---|---|---|---|
| Fast-dLLM v2 | *GSM8K* | FP | 15,083 | 91 | 1.34× | 61% |
| | | MV-Quant | 5,883 | 122 | | |
| | *HumanEval* | FP | 15,179 | 95 | 1.38× | 60% |
| | | MV-Quant | 6,123 | 131 | | |
| SDAR | *GSM8K* | FP | 15,893 | 50 | 1.42× | 61% |
| | | MV-Quant | 6,164 | 71 | | |
| | *HumanEval* | FP | 16,006 | 46 | 1.39× | 61% |
| | | MV-Quant | 6,289 | 64 | | |

while keeping all other components unchanged. MV-Quant yields $73{,}216$ while AWQ yields $74{,}752$. This indicates that our objective better controls the quantization error, providing empirical support for the proposed theoretical analysis.

### 5.3 Memory and Speedup

In this section, we focus on 4-bit kernel inference because existing AWQ inference implementations only provide optimized INT4 kernels (Lin et al., 2024). We evaluate the efficiency of 4-bit quantization in terms of GPU memory consumption and inference throughput. Results are shown in Table 5. As can be seen, for Fast-dLLMv2-7B, quantization substantially reduces the memory while consistently improving decoding speed. On *GSM8K* dataset, the peak memory usage decreases from 15,083 MiB to 5,883 MiB, leading to a 61.0% reduction, while the throughput increases from 91 to 122 tokens per second (TPS), yielding a 1.34× speedup. On *HumanEval* dataset, the memory usage is reduced from 15,179 MiB to 6,123 MiB, with throughput improving from 95 to 131 TPS, resulting in a 59.7% memory saving and a 1.38× speedup.

For SDAR-8B, we observe a similar efficiency gain. On *GSM8K* dataset, 4-bit quantization reduces the peak memory consumption from 15,893 MiB to 6,164 MiB, achieving a 61.2% memory saving. On *HumanEval* dataset, the peak memory usage decreases from 16,006 MiB to 6,289 MiB, while the throughput improves from 45.96 to 64.11 TPS, corresponding to a 60.7% memory saving and a 1.39× speedup.

These results demonstrate that the proposed quantization algorithm provides substantial memory reduction and non-trivial inference acceleration, making diffusion language models more practical for resource-constrained deployment.

## 6 Conclusion

In this work, we theoretically investigate the post-training quantization for diffusion language models and presented MV-Quant, a new quantization objective for diffusion language model quantization. Our analysis shows that quantization errors in transformers accumulate as propagated layer-wise errors, and that in DLMs each layer's error decomposes into masked-token and visible-token components with input- and projection-dependent importance. This makes the standard MSE reconstruction objective a potentially suboptimal surrogate. By directly minimizing the worst reconstruction error across masked and visible tokens, MV-Quant provides a tighter upper bound on both layer-wise and end-to-end quantization error than MSE, except in the balanced case. Experiments on Fast-dLLMv2-7B further show that replacing the MSE calibration loss in AWQ with MV-Quant reduces the measured quantization error. These results highlight the importance of DLM-specific quantization objectives and suggest worst-group error control as a simple and principled approach for diffusion language models quantization.

## AI USE STATEMENT

In this work, we used generative AI tools solely to aid and polish the writing, including improving grammar, wording, and readability. We did not use generative AI tools for research ideation or execution, retrieval or discovery of related work, drafting substantive content, generating synthetic datasets, or proving mathematical claims. All AI-assisted edits were reviewed and verified by the authors. We take full responsibility for the final content of this work.

## REFERENCES


Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *NeurIPS*, 2021a.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021b.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. In *ACL*, 2025.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, et al. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Llm.int8 (): 8-bit matrix multiplication for transformers at scale. *NeurIPS*, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407, 2024.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *ICLR*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *ICLR*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020.

Yu Huang, Yuan Cheng, and Yingbin Liang. In-context convergence of transformers. In *ICML*, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

Zihao Li, Yuan Cao, Cheng Gao, Yihan He, Han Liu, Jason Klusowski, Jianqing Fan, and Mengdi Wang. One-layer transformer provably learns one-nearest neighbor in context. *NeurIPS*, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Machine Learning and Systems*, 2024.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024a.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. Llm-qat: Data-free quantization aware training for large language models. In *ACL Findings*, 2024b.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. In *NeurIPS*, 2025.

NVIDIA. Nvidia ampere architecture in-depth. *Tech Report*, 2020. URL https://developer.nvidia.com/blog/nvidia-ampere-architecture-in-depth/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *COLM*, 2024.

Grefenstette Saxton and Kohli Hill. Analysing mathematical reasoning abilities of neural models. *arXiv:1904.01557*, 2019.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. Omniquant: Omnidirectionally calibrated quantization for large language models. *ICLR*, 2023.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dllm v2: Efficient block-diffusion llm. In *ICLR*, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, 2023.

Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, et al. Dream-coder 7b: An open diffusion language model for code. *arXiv preprint arXiv:2509.01142*, 2025.

Chen Xu and Dawei Yang. Dllmquant: Quantizing diffusion-based large language models. *Tech report arXiv:2508.14090*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *Tech report arXiv:2505.09388*, 2025.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Tianao Zhang, Zhiteng Li, Xianglong Yan, Haotong Qin, Yong Guo, and Yulun Zhang. Quant-dllm: Post-training extreme low-bit quantization for diffusion large language models. *ICLR*, 2025.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *Tech report arXiv:2311.07911*, 2023.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, et al. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. In *NeurIPS*, 2025.

## A Proof of Proposition 3.1

*Proof.* Using the decomposition in Equation (4), we obtain

$$E^{(l)} = E^{(l-1)} + \underbrace{\widehat{\mathcal{F}}^{(l)}\left(\widehat{H}^{(l-1)}\right) - \widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right)}_{\text{propagation of the input error}} + \underbrace{\widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right) - \mathcal{F}^{(l)}\left(H^{(l-1)}\right)}_{\xi^{(l)}}. \tag{10}$$

By definition,

$$\xi^{(l)} := \widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right) - \mathcal{F}^{(l)}\left(H^{(l-1)}\right) \tag{11}$$

is the new quantization error introduced at layer $l$.

Next, apply the first-order Taylor expansion of $\widehat{\mathcal{F}}^{(l)}$ around $H^{(l-1)}$:

$$\widehat{\mathcal{F}}^{(l)}\left(\widehat{H}^{(l-1)}\right) - \widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right) \approx \nabla_{H^{(l-1)}}\widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right)\left(\widehat{H}^{(l-1)} - H^{(l-1)}\right). \tag{12}$$

Since $\widehat{H}^{(l-1)} - H^{(l-1)} = E^{(l-1)}$, substituting into Equation (10) yields

$$\begin{aligned} E^{(l)} &= E^{(l-1)} + \nabla_{H^{(l-1)}}\widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right)E^{(l-1)} + \xi^{(l)} \\ &= \left[I + \nabla_{H^{(l-1)}}\widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right)\right]E^{(l-1)} + \xi^{(l)}. \end{aligned} \tag{13}$$

Defining

$$T^{(l)} := \left[I + \nabla_{H^{(l-1)}}\widehat{\mathcal{F}}^{(l)}\left(H^{(l-1)}\right)\right], \tag{14}$$

we obtain

$$E^{(l)} = T^{(l)}E^{(l-1)} + \xi^{(l)}, \tag{15}$$

□

## B Proof of Proposition 3.2

*Proof.* The layer-wise error recursion can be written compactly as

$$E^{(l)} = T^{(l)}E^{(l-1)} + \xi^{(l)}.$$

Assume zero initial error, i.e., $E_m^{(0)} = E_v^{(0)} = 0$. We first unroll the recursion:

$$E^{(n)} = \sum_{l=1}^{n} T^{(n)}T^{(n-1)}\cdots T^{(l+1)}\xi^{(l)},$$

where the product is treated as the identity operator when $l = n$.

To see this, the first few layers satisfy

$$E^{(1)} = \xi^{(1)},$$
$$E^{(2)} = T^{(2)}\xi^{(1)} + \xi^{(2)},$$

and

$$E^{(3)} = T^{(3)}T^{(2)}\xi^{(1)} + T^{(3)}\xi^{(2)} + \xi^{(3)}.$$

Thus, by induction, the general unrolled form is

$$E^{(n)} = \sum_{l=1}^{n} T^{(n)}T^{(n-1)}\cdots T^{(l+1)}\xi^{(l)}.$$

Now define the accumulated propagation operator from layer $l+1$ to layer $n$ as

$$\Phi^{(n,l+1)} = T^{(n)}T^{(n-1)}\cdots T^{(l+1)}.$$

Since each $T^{(j)}$ is partitioned according to masked and visible tokens, $\Phi^{(n,l+1)}$ is also a block operator:

$$\Phi^{(n,l+1)} = \begin{bmatrix} \Phi_{mm}^{(n,l+1)} & \Phi_{mv}^{(n,l+1)} \\ \Phi_{vm}^{(n,l+1)} & \Phi_{vv}^{(n,l+1)} \end{bmatrix}.$$

Therefore, the contribution of the layer-$l$ local error to the final masked-token error is

$$\left[\Phi^{(n,l+1)}\xi^{(l)}\right]_m = \begin{bmatrix} \Phi_{mm}^{(n,l+1)} & \Phi_{mv}^{(n,l+1)} \end{bmatrix} \begin{bmatrix} \xi_m^{(l)} \\ \xi_v^{(l)} \end{bmatrix}.$$

Hence,

$$\left[\Phi^{(n,l+1)}\xi^{(l)}\right]_m = \Phi_{mm}^{(n,l+1)}\xi_m^{(l)} + \Phi_{mv}^{(n,l+1)}\xi_v^{(l)}.$$

Taking the masked block of $E^{(n)}$, we obtain

$$E_m^{(n)} = \sum_{l=1}^{n} \left(\Phi_{mm}^{(n,l+1)}\xi_m^{(l)} + \Phi_{mv}^{(n,l+1)}\xi_v^{(l)}\right).$$

Finally, define

$$A_l = \Phi_{mm}^{(n,l+1)} = \left[T^{(n)}T^{(n-1)}\cdots T^{(l+1)}\right]_{mm},$$

and

$$B_l = \Phi_{mv}^{(n,l+1)} = \left[T^{(n)}T^{(n-1)}\cdots T^{(l+1)}\right]_{mv}.$$

When $l = n$, the product is empty, we define $A_n = I_m$ and $B_n = 0$. Substituting these definitions gives

$$E_m^{(n)} = \sum_{l=1}^{n} A_l\xi_m^{(l)} + \sum_{l=1}^{n} B_l\xi_v^{(l)}.$$

where

$$A_l = \left[T^{(n)}T^{(n-1)}\cdots T^{(l+1)}\right]_{mm}, \qquad B_l = \left[T^{(n)}T^{(n-1)}\cdots T^{(l+1)}\right]_{mv},$$

Therefore,

$$\left\|E_m^{(n)}\right\|_F = \left\|\sum_{l=1}^{n} A_l\xi_m^{(l)} + \sum_{l=1}^{n} B_l\xi_v^{(l)}\right\|_F \tag{16}$$

$$\leq \sum_{l=1}^{n} \left\|A_l\xi_m^{(l)} + B_l\xi_v^{(l)}\right\|_F \tag{17}$$

□

## C Proof of Theorem 3.1

*Proof.* For simplicity, we suppress the layer index $(l)$ throughout the proof.

Let

$$\xi = \widehat{\mathcal{F}}(H) - \mathcal{F}(H)$$

denote the single-layer quantization error evaluated at the same input hidden state $H$. Partition the output rows into visible-token and masked-token blocks:

$$\xi = \begin{bmatrix} \xi_v \\ \xi_m \end{bmatrix}.$$

**Step 1: First-order expansion of the visible and masked output blocks.** For a single self-attention layer,

$$O = PVW_O, \qquad P = \mathrm{Softmax}(S), \qquad S = \frac{QK^\top}{\sqrt{d_k}}.$$

Under the first-order approximation of the visible-token attention output with respect to the perturbations in $P$, $V$, and $W_O$, the visible-output perturbation satisfies

$$\begin{aligned} B\xi_v = {} & B(\Delta P_{v,v})V_vW_O + BP_{v,v}(\Delta V_v)W_O + BP_{v,v}V_v\Delta W_O \\ & + B(\Delta P_{v,m})V_mW_O + BP_{v,m}(\Delta V_m)W_O + BP_{v,m}V_m\Delta W_O, \end{aligned} \tag{18}$$

and the masked-output perturbation satisfies

$$\begin{aligned} A\xi_m = {} & A(\Delta P_{m,v})V_vW_O + AP_{m,v}(\Delta V_v)W_O + AP_{m,v}V_v\Delta W_O \\ & + A(\Delta P_{m,m})V_mW_O + AP_{m,m}(\Delta V_m)W_O + AP_{m,m}V_m\Delta W_O. \end{aligned} \tag{19}$$

**Step 2: Bound the $Q$- and $K$-induced terms.** For each query group $r \in \{m, v\}$, partition the attention scores and probabilities according to visible and masked key positions:

$$S_r = [S_{r,v}, S_{r,m}], \qquad P_r = [P_{r,v}, P_{r,m}].$$

Since the row-wise softmax jointly normalizes over all key positions, its Jacobian is generally not block diagonal with respect to the visible/masked key partition. Therefore, the first-order perturbation takes the form

$$\Delta P_{r,v} \approx \mathcal{J}_r^{v\leftarrow v}[\Delta S_{r,v}] + \mathcal{J}_r^{v\leftarrow m}[\Delta S_{r,m}], \tag{20}$$

$$\Delta P_{r,m} \approx \mathcal{J}_r^{m\leftarrow v}[\Delta S_{r,v}] + \mathcal{J}_r^{m\leftarrow m}[\Delta S_{r,m}], \tag{21}$$

where $\mathcal{J}_r^{a\leftarrow b}$ denotes the corresponding block of the row-wise softmax Jacobian, mapping perturbations of score columns in key group $b$ to probability columns in group $a$.

Define

$$j_r^{a\leftarrow b} := \left\|\mathcal{J}_r^{a\leftarrow b}\right\|_{F\to F},$$

where $\|\cdot\|_{F\to F}$ denotes the induced operator norm under the Frobenius norm.

For $r \in \{m, v\}$, the score perturbations are

$$\Delta S_{r,v} = \frac{(\Delta Q_r)K_v^\top + Q_r(\Delta K_v)^\top}{\sqrt{d_k}}, \tag{22}$$

$$\Delta S_{r,m} = \frac{(\Delta Q_r)K_m^\top + Q_r(\Delta K_m)^\top}{\sqrt{d_k}}. \tag{23}$$

Using

$$\Delta Q_r = X_r\Delta W_Q, \qquad \Delta K_v = X_v\Delta W_K, \qquad \Delta K_m = X_m\Delta W_K,$$

we obtain

$$\|\Delta S_{r,v}\|_F \le \frac{1}{\sqrt{d_k}}\left(\|K_v\|_F\epsilon_{Q,r} + \|Q_r\|_F\epsilon_{K,v}\right), \tag{24}$$

$$\|\Delta S_{r,m}\|_F \le \frac{1}{\sqrt{d_k}}\left(\|K_m\|_F\epsilon_{Q,r} + \|Q_r\|_F\epsilon_{K,m}\right). \tag{25}$$

Combining Equations (20), (21), (24) and (25), the masked-query terms satisfy

$$\begin{aligned} \|A(\Delta P_{m,v})V_vW_O\|_F \le {} & \frac{\|A\|_F\|W_O\|_F\|V_v\|_F}{\sqrt{d_k}}\Big[j_m^{v\leftarrow v}\big(\|K_v\|_F\epsilon_{Q,m} + \|Q_m\|_F\epsilon_{K,v}\big) \\ & + j_m^{v\leftarrow m}\big(\|K_m\|_F\epsilon_{Q,m} + \|Q_m\|_F\epsilon_{K,m}\big)\Big], \end{aligned} \tag{26}$$

$$\begin{aligned} \|A(\Delta P_{m,m})V_mW_O\|_F \le {} & \frac{\|A\|_F\|W_O\|_F\|V_m\|_F}{\sqrt{d_k}}\Big[j_m^{m\leftarrow v}\big(\|K_v\|_F\epsilon_{Q,m} + \|Q_m\|_F\epsilon_{K,v}\big) \\ & + j_m^{m\leftarrow m}\big(\|K_m\|_F\epsilon_{Q,m} + \|Q_m\|_F\epsilon_{K,m}\big)\Big]. \end{aligned} \tag{27}$$

Similarly, the visible-query terms satisfy

$$\|B(\Delta P_{v,v})V_vW_O\|_F \le \frac{\|B\|_F\|W_O\|_F\|V_v\|_F}{\sqrt{d_k}}\Big[j_v^{v\leftarrow v}\big(\|K_v\|_F\epsilon_{Q,v} + \|Q_v\|_F\epsilon_{K,v}\big) + j_v^{v\leftarrow m}\big(\|K_m\|_F\epsilon_{Q,v} + \|Q_v\|_F\epsilon_{K,m}\big)\Big], \tag{28}$$

$$\|B(\Delta P_{v,m})V_mW_O\|_F \le \frac{\|B\|_F\|W_O\|_F\|V_m\|_F}{\sqrt{d_k}}\Big[j_v^{m\leftarrow v}\big(\|K_v\|_F\epsilon_{Q,v} + \|Q_v\|_F\epsilon_{K,v}\big) + j_v^{m\leftarrow m}\big(\|K_m\|_F\epsilon_{Q,v} + \|Q_v\|_F\epsilon_{K,m}\big)\Big]. \tag{29}$$

Collecting the terms induced by $\Delta W_Q$ gives

$$\Big\|A\xi_m^{(Q)} + B\xi_v^{(Q)}\Big\|_F \le c_{Q,m}\epsilon_{Q,m} + c_{Q,v}\epsilon_{Q,v}, \tag{30}$$

where

$$c_{Q,m} = \frac{\|A\|_F\|W_O\|_F}{\sqrt{d_k}}\Big[\|V_v\|_F\big(j_m^{v\leftarrow v}\|K_v\|_F + j_m^{v\leftarrow m}\|K_m\|_F\big) + \|V_m\|_F\big(j_m^{m\leftarrow v}\|K_v\|_F + j_m^{m\leftarrow m}\|K_m\|_F\big)\Big], \tag{31}$$

$$c_{Q,v} = \frac{\|B\|_F\|W_O\|_F}{\sqrt{d_k}}\Big[\|V_v\|_F\big(j_v^{v\leftarrow v}\|K_v\|_F + j_v^{v\leftarrow m}\|K_m\|_F\big) + \|V_m\|_F\big(j_v^{m\leftarrow v}\|K_v\|_F + j_v^{m\leftarrow m}\|K_m\|_F\big)\Big]. \tag{32}$$

Likewise, collecting the terms induced by $\Delta W_K$ gives

$$\Big\|A\xi_m^{(K)} + B\xi_v^{(K)}\Big\|_F \le c_{K,m}\epsilon_{K,m} + c_{K,v}\epsilon_{K,v}, \tag{33}$$

where

$$c_{K,m} = \frac{\|W_O\|_F}{\sqrt{d_k}}\Bigg[\|A\|_F\|Q_m\|_F\Big(j_m^{v\leftarrow m}\|V_v\|_F + j_m^{m\leftarrow m}\|V_m\|_F\Big) + \|B\|_F\|Q_v\|_F\Big(j_v^{v\leftarrow m}\|V_v\|_F + j_v^{m\leftarrow m}\|V_m\|_F\Big)\Bigg], \tag{34}$$

$$c_{K,v} = \frac{\|W_O\|_F}{\sqrt{d_k}}\Bigg[\|A\|_F\|Q_m\|_F\Big(j_m^{v\leftarrow v}\|V_v\|_F + j_m^{m\leftarrow v}\|V_m\|_F\Big) + \|B\|_F\|Q_v\|_F\Big(j_v^{v\leftarrow v}\|V_v\|_F + j_v^{m\leftarrow v}\|V_m\|_F\Big)\Bigg]. \tag{35}$$

**Step 3: Bound the $V$- and $O$-induced terms.** Since

$$\Delta V_v = X_v\Delta W_V, \qquad \Delta V_m = X_m\Delta W_V,$$

the $V$-induced terms in Equations (18) and (19) satisfy

$$\|AP_{m,v}(\Delta V_v)W_O\|_F \le \|A\|_F\|P_{m,v}\|_F\|W_O\|_F\epsilon_{V,v}, \tag{36}$$

$$\|BP_{v,v}(\Delta V_v)W_O\|_F \le \|B\|_F\|P_{v,v}\|_F\|W_O\|_F\epsilon_{V,v}, \tag{37}$$

$$\|AP_{m,m}(\Delta V_m)W_O\|_F \le \|A\|_F\|P_{m,m}\|_F\|W_O\|_F\epsilon_{V,m}, \tag{38}$$

$$\|BP_{v,m}(\Delta V_m)W_O\|_F \le \|B\|_F\|P_{v,m}\|_F\|W_O\|_F\epsilon_{V,m}. \tag{39}$$

Hence,

$$\Big\|A\xi_m^{(V)} + B\xi_v^{(V)}\Big\|_F \le c_{V,m}\epsilon_{V,m} + c_{V,v}\epsilon_{V,v}, \tag{40}$$

where

$$c_{V,m} = \left(\|A\|_F \|P_{m,m}\|_F + \|B\|_F \|P_{v,m}\|_F\right) \|W_O\|_F, \tag{41}$$
$$c_{V,v} = \left(\|A\|_F \|P_{m,v}\|_F + \|B\|_F \|P_{v,v}\|_F\right) \|W_O\|_F. \tag{42}$$

For the $O$-induced terms, by definition,

$$\epsilon_{O,m} = \|(PV)_m \Delta W_O\|_F, \qquad \epsilon_{O,v} = \|(PV)_v \Delta W_O\|_F.$$

Since

$$(PV)_m = P_{m,v} V_v + P_{m,m} V_m, \tag{43}$$
$$(PV)_v = P_{v,v} V_v + P_{v,m} V_m, \tag{44}$$

we have

$$\xi_m^{(O)} = (PV)_m \Delta W_O, \tag{45}$$
$$\xi_v^{(O)} = (PV)_v \Delta W_O. \tag{46}$$

Therefore,

$$\left\| A\xi_m^{(O)} + B\xi_v^{(O)} \right\|_F \le \|A(PV)_m \Delta W_O\|_F + \|B(PV)_v \Delta W_O\|_F \tag{47}$$
$$\le \|A\|_F \|(PV)_m \Delta W_O\|_F + \|B\|_F \|(PV)_v \Delta W_O\|_F \tag{48}$$
$$= \|A\|_F \epsilon_{O,m} + \|B\|_F \epsilon_{O,v}. \tag{49}$$

Hence,

$$\left\| A\xi_m^{(O)} + B\xi_v^{(O)} \right\|_F \le c_{O,m}\epsilon_{O,m} + c_{O,v}\epsilon_{O,v}, \tag{50}$$

where

$$c_{O,m} = \|A\|_F, \tag{51}$$
$$c_{O,v} = \|B\|_F. \tag{52}$$

**Step 4: Collect all four projection errors.** Under the first-order approximation, the local perturbation decomposes as

$$\xi_\tau = \xi_\tau^{(Q)} + \xi_\tau^{(K)} + \xi_\tau^{(V)} + \xi_\tau^{(O)}, \qquad \tau \in \{m, v\}.$$

Therefore, by the triangle inequality,

$$\|A\xi_m + B\xi_v\|_F = \left\| \sum_{i \in \{Q,K,V,O\}} \left( A\xi_m^{(i)} + B\xi_v^{(i)} \right) \right\|_F \tag{53}$$
$$\le \sum_{i \in \{Q,K,V,O\}} \left\| A\xi_m^{(i)} + B\xi_v^{(i)} \right\|_F \tag{54}$$
$$\le \sum_{i \in \{Q,K,V,O\}} \left( c_{i,m}\epsilon_{i,m} + c_{i,v}\epsilon_{i,v} \right). \tag{55}$$

Finally, define

$$\alpha_i := c_{i,m} + c_{i,v}, \qquad \beta_i := \frac{c_{i,v}}{c_{i,m} + c_{i,v}}. \tag{56}$$

Since $c_{i,m}, c_{i,v} \ge 0$, we have $\alpha_i \ge 0$ and $\beta_i \in [0, 1]$ whenever $\alpha_i > 0$. Moreover,

$$c_{i,m}\epsilon_{i,m} + c_{i,v}\epsilon_{i,v} = \alpha_i \left( (1 - \beta_i)\epsilon_{i,m} + \beta_i \epsilon_{i,v} \right).$$

Substituting this identity into Equation (55) gives

$$\|A\xi_m + B\xi_v\|_F \le \sum_{i \in \{Q,K,V,O\}} \alpha_i \left( (1 - \beta_i)\epsilon_{i,m} + \beta_i \epsilon_{i,v} \right) = \mathrm{UB}(\Theta, \alpha, \beta). \tag{57}$$

This completes the proof. □

## D Proof of Theorem 4.1

*Proof.* Starting from Equation (7), we have

$$\left\|A_l\xi_m^{(l)} + B_l\xi_v^{(l)}\right\|_F \le \sum_{i\in I}\alpha_i^{(l)}\left((1-\beta_i^{(l)})\epsilon_{i,m}^{(l)} + \beta_i^{(l)}\epsilon_{i,v}^{(l)}\right). \tag{58}$$

Since each reconstruction error is a norm,

$$\epsilon_{i,m}^{(l)} \ge 0, \qquad \epsilon_{i,v}^{(l)} \ge 0, \qquad \forall i \in I. \tag{59}$$

**MSE-based objective.** Suppose the MSE-based objective satisfies the tolerance

$$\frac{\left(\epsilon_{i,m}^{(l)}\right)^2 + \left(\epsilon_{i,v}^{(l)}\right)^2}{2} \le \epsilon^2, \qquad \forall i \in I. \tag{60}$$

Equivalently,

$$\left(\epsilon_{i,m}^{(l)}\right)^2 + \left(\epsilon_{i,v}^{(l)}\right)^2 \le 2\epsilon^2. \tag{61}$$

For each fixed $i \in I$, by the Cauchy–Schwarz inequality,

$$\begin{aligned}
&(1-\beta_i^{(l)})\epsilon_{i,m}^{(l)} + \beta_i^{(l)}\epsilon_{i,v}^{(l)} \\
&\le \sqrt{\left(1-\beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2}\sqrt{\left(\epsilon_{i,m}^{(l)}\right)^2 + \left(\epsilon_{i,v}^{(l)}\right)^2} \\
&\le \sqrt{2}\epsilon\sqrt{\left(1-\beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2}.
\end{aligned} \tag{62}$$

Substituting Equation (62) into Equation (58) gives

$$\left\|A_l\xi_m^{(l)} + B_l\xi_v^{(l)}\right\|_F \le \sqrt{2}\epsilon\sum_{i\in I}\alpha_i^{(l)}\sqrt{\left(1-\beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2} \coloneqq \mathrm{UB}_{\mathrm{mse}}. \tag{63}$$

**MV-Quant objective.** Suppose the MV-Quant objective satisfies the tolerance

$$\max\left\{\left(\epsilon_{i,m}^{(l)}\right)^2, \left(\epsilon_{i,v}^{(l)}\right)^2\right\} \le \epsilon^2, \qquad \forall i \in I. \tag{64}$$

Since the reconstruction errors are nonnegative, this is equivalent to

$$\epsilon_{i,m}^{(l)} \le \epsilon, \qquad \epsilon_{i,v}^{(l)} \le \epsilon. \tag{65}$$

Therefore, for every $i \in I$,

$$\begin{aligned}
(1-\beta_i^{(l)})\epsilon_{i,m}^{(l)} + \beta_i^{(l)}\epsilon_{i,v}^{(l)} &\le (1-\beta_i^{(l)})\epsilon + \beta_i^{(l)}\epsilon \\
&= \epsilon.
\end{aligned} \tag{66}$$

Substituting Equation (66) into Equation (58) yields

$$\left\|A_l\xi_m^{(l)} + B_l\xi_v^{(l)}\right\|_F \le \epsilon\sum_{i\in I}\alpha_i^{(l)} \coloneqq \mathrm{UB}_{\max}. \tag{67}$$

**Comparison of the two bounds.** For any $\beta_i^{(l)} \in [0,1]$,

$$\left(1-\beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2 = 2\left(\beta_i^{(l)} - \frac{1}{2}\right)^2 + \frac{1}{2} \ge \frac{1}{2}. \tag{68}$$

Hence,

$$\sqrt{2}\sqrt{\left(1-\beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2} \ge 1. \tag{69}$$

Multiplying by $\alpha_i^{(l)} \geq 0$ and summing over $i \in I$ gives

$$\sqrt{2} \sum_{i \in I} \alpha_i^{(l)} \sqrt{\left(1 - \beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2} \geq \sum_{i \in I} \alpha_i^{(l)}. \tag{70}$$

Therefore,

$$\frac{\mathrm{UB}_{\max}}{\mathrm{UB}_{\mathrm{mse}}} = \frac{\sum_{i \in I} \alpha_i^{(l)}}{\sqrt{2} \sum_{i \in I} \alpha_i^{(l)} \sqrt{\left(1 - \beta_i^{(l)}\right)^2 + \left(\beta_i^{(l)}\right)^2}} \leq 1, \tag{71}$$

and thus

$$\mathrm{UB}_{\max} \leq \mathrm{UB}_{\mathrm{mse}}. \tag{72}$$

Equality holds if and only if

$$\beta_i^{(l)} = \frac{1}{2} \tag{73}$$

for every $i \in I$ such that $\alpha_i^{(l)} > 0$. □