CALIBRATING ATTENTION-BASED TOKEN PRUNING FOR EFFICIENT MULTI-IMAGE SYNTHESIS IN UMMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Unified Multimodal Models (UMMs) enable powerful multi-image synthesis but incur substantial computation from long visual contexts. While token pruning offers a direct route to accelerating inference, existing methods are primarily adapted from understanding tasks, suffering from insufficient reference image exploration, confused conditional guidance, and pooling-induced ambiguity that degrades generation quality. To address these issues, we propose CATP, the first training-free context-compression method tailored to UMM generation, which **C**alibrates the discrepancies of general **A**ttention-based **T**oken **P**runing for better quality-efficiency trade-off. Concretely, building on independent layer-wise pruning during denoising, we introduce Guidance-Consistent Pruning (GCP) to synchronize pruning decisions across conditioning branches, and a Taylor-Calibrated Attention Metric (TCAM) that derives a theoretically grounded second-order correction for pooling-induced bias. In addition, a Fast Scoring Operator (FSO) fuses the calibration into tiled online Softmax to reduce pruning cost. Extensive experiments across representative UMMs and benchmarks demonstrate that CATP consistently outperforms state-of-the-art token-reduction methods, achieving up to 2.09$\times$ attention-core speedup. Code will be released upon paper acceptance.

1 INTRODUCTION

Unified Multimodal Models (UMMs), which integrate understanding and generation within a shared backbone (Deng et al., 2025; Diao et al., 2026), have emerged as a prevalent paradigm for enhancing both capabilities (Zhao et al., 2025). Leveraging their strength in handling complex inputs, multi-image synthesis, which generates a target image from multiple references (Chen et al., 2026; Wei et al., 2026), has recently attracted growing interest in broad applications including fashion design (Choi et al., 2024), personalized content production (Wang et al., 2025), etc. However, this typically requires concatenating visual tokens from multiple high-resolution images, yielding long sequences that incur substantial attention computation during denoising (Chen et al., 2026; He et al., 2025).

In general, such long-context designs are originally intended to support diverse capabilities of foundation models (Deng et al., 2025), whereas a specific task may exploit only a subset of the available information (Zhang et al., 2025b). Therefore, reducing the effective sequence length is a direct route to improving inference efficiency (Bolya et al., 2023; Chen et al., 2024b). Among strategies, attention-based token pruning offers an effective and general baseline, leveraging native attention score to discard less informative tokens (Yao et al., 2026; Wu et al., 2026b). Nevertheless, existing schemes are primarily developed for understanding-oriented models (Chen et al., 2024b; Wu et al., 2026a), while the few direct adaptations to UMMs overlook generation-specific characteristics (Ke et al., 2026; Chen et al., 2026), potentially discarding visual regions critical to faithful synthesis.

Specifically, we identify three key limitations: 1) **Static pruning across timesteps and layers.** Understanding tasks usually produce short responses (Liu et al., 2024), with most information aggregated into a few tokens at early layers (Wu et al., 2026a), making one-shot pruning throughout inference feasible (Chen et al., 2024b). In contrast, visual generation attends to complementary regions across denoising timesteps and layers, as in Fig. 1(a). Especially under aggressive token budgets, a fixed pruning fails to cover the full reference content, yielding insufficient exploration of visual cues. 2) **Inconsistent compression across conditioning branches.** Classifier-Free Guidance (CFG) (Ho & Salimans, 2021), natively instantiated in UMMs (Deng et al., 2025), performs mul-

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.

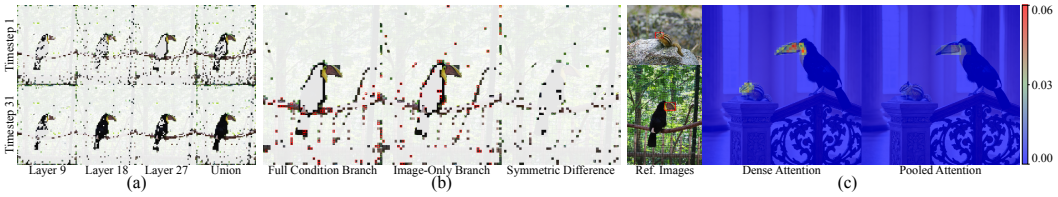


Figure 1: Empirical observations on Bagel using the general attention metric. (a) Top-10% tokens at timesteps 1/31 across layers 9/18/27, with their union in the last column. (b) Top-10% tokens from the (Left) full-condition and (Middle) image-only branches, with (Right) their symmetric difference; red boxes indicate branch-specific tokens. (c) (Left) Reference images with manually selected synthesis-critical regions highlighted in red boxes. (Middle/Right) Heatmaps of the maximum attention from generated-image locations to these regions under dense and pooled attention, respectively.

tiple forward passes with different input sequences, naturally producing branch-specific attention scores and motivating independent pruning. However, as shown in Fig. 1(b), this yields mismatched token selections whose symmetric difference carries useless information yet is amplified in the CFG guidance residual, perturbing guided velocity prediction and compromising generation fidelity. 3) **Ambiguous token selection under pooled query representations.** Adapting pruning to varying context exploration requires frequent importance estimation, while exact attention-based metric with the still-lengthy denoising queries incurs substantial overhead, undermining the goal of acceleration. Average pooling on queries is commonly adopted to reduce this cost (Zhang et al., 2025a). However, as observed in Fig. 1(c), it smooths fine-grained attention responses to critical regions, obscuring their distinction from less relevant content and thus yielding less reliable context retention.

To address these issues, we propose CATP, a novel training-free framework for efficient multi-image synthesis in UMMs. Starting from the general attention-based token pruning, CATP identifies generation-specific discrepancies and calibrates them at both conditioning-branch and importance-estimation levels, thereby improving generation fidelity. Specifically, CATP prunes reference-image tokens in the KV cache across all layers at predefined timesteps, adapting compression to varying reference explorations. Building on this, we introduce Guidance-Consistent Pruning (GCP), which formulates the cross-branch mismatch of independent compression and synchronizes pruning decisions with the full-condition branch to preserve clean guidance. A second-order Taylor expansion then characterizes the bias of attention metric with pooled-query, leading to a Taylor-Calibrated Attention Metric (TCAM) for more reliable identification of critical reference tokens. Additionally, a Fast Scoring Operator (FSO) is developed with a custom Triton kernel. It integrates the calibration into tiled online Softmax computation, substantially reducing the scoring overhead.

In summary, our main contributions are three-fold:

1) We analyze discrepancies in general attention-based token pruning for UMM multi-image synthesis and propose a training-free acceleration framework, dubbed CATP. To the best of our knowledge, CATP makes the first investigation of token pruning specifically tailored to UMM generation.

2) We develop Guidance-Consistent Pruning (GCP), Taylor-Calibrated Attention Metric (TCAM) and Fast Scoring Operator (FSO), which address the rigidity, inconsistency, and ambiguity of general schemes, jointly enabling a flexible, accurate and efficient online context compression.

3) We conduct extensive experiments on representative UMMs and benchmarks, showing that CATP delivers superior generation quality, with up to 2.09$\times$ attention-core acceleration.

2 Related Work

2.1 Unified Multimodal Models and Multi-Image Synthesis

Unified Multimodal Models (UMMs) enable understanding and generation within a single architecture (Zhao et al., 2025). Recent prevalent paradigm adopts a Large Language Model (LLM) as the backbone (Zhao et al., 2025), where text generation remains autoregressive process, while images are synthesized via iterative flow matching in latent (Deng et al., 2025) or pixel space (Diao et al., 2026). For image generation, conditioning inputs are encoded and concatenated into a token sequence, cached as fixed KV states, and repeatedly attended to by generated visual tokens throughout

denoising (Deng et al., 2025; Cao et al., 2025). Multi-image synthesis leverages UMMs' strengths in complex instruction understanding and high-quality generation (Wei et al., 2026; Oshima et al., 2026). However, multiple reference images substantially lengthen the input sequence, underscoring the practical necessity of efficient inference, especially in attention computation (Chen et al., 2026).

2.2 Efficient Inference via Token Reduction

Token Pruning. Token pruning has presented clear efficiency gains for long-context tasks (Yao et al., 2026), exploiting the task-irrelevant redundancy inherent in input sequences (Wu et al., 2026a). Initially explored in understanding tasks, methods develop various criteria for token selection (Yang et al., 2025; Alvar et al., 2025). Among them, attention-based pruning offers a strong baseline across LLMs (Wu et al., 2026a), Multimodal Large Language Models (MLLMs) (Chen et al., 2024b), and the understanding branch of UMM (Ke et al., 2026), accelerating either prefilling through early token removal (Chen et al., 2024b) or decoding via KV cache pruning (Wan et al., 2024).

Token Reduction for Visual Generation. Efficiency in visual generation has been studied from multiple perspectives, such as timestep reduction (Fan et al., 2026) and cross-step cache reusing (Liu et al., 2025; Zhao et al., 2026), while token reduction is orthogonal to them. Most methods originally target Diffusion Transformers (DiTs) (Peebles & Xie, 2023), typically reducing the computed sequence by merging and subsequently unmerging denoising tokens (Bolya & Hoffman, 2023; Fang et al., 2025). ToPi (Lin et al., 2026) more closely follows the aforementioned token pruning paradigm, but shares pruning decisions across layers, overlooking layer-wise differences in information demand. The few attempts on UMMs largely transfer understanding-oriented techniques (Chen et al., 2026), neglecting generation-specific characteristics. In this work, we systematically analyze these distinctive properties and develop token pruning specifically tailored to UMM generation.

3 Methodology

3.1 Framework Overview

In context of mainstream UMMs, multiple reference images and an instruction are encoded into visual and text token sequences $\boldsymbol{I}$ and $\boldsymbol{T}$, respectively, and concatenated to form a context $\boldsymbol{C}=[\boldsymbol{I};\boldsymbol{T}]$ of length N. Prefilling $\boldsymbol{C}$ through the L-layer, θ-parameterized backbone $f_\theta(\cdot)$ produces layer-wise representations $\{\boldsymbol{C}^l\}_{l=1}^{L}$, cached as fixed KV states. Starting from noisy visual tokens $\boldsymbol{Z}_T$ of length M, generation proceeds for T iterative flow-based denoising steps as Eq. (1), where the intermediate visual states $\boldsymbol{Z}_t^l$ attend to $\boldsymbol{C}^l$ within the attention blocks at each layer, and $\boldsymbol{Z}_0$ is ultimately decoded into the output image.

$$\boldsymbol{Z}_{t-1} = f_\theta(\boldsymbol{Z}_t; t, \{\boldsymbol{C}^l\}_{l=1}^{L}),\ t = T, \dots, 1. \tag{1}$$

Token pruning shortens $\boldsymbol{C}^l$ at a predefined timestep t^{p}, reducing attention cost in the current and subsequent denoising steps. Motivated by the dynamic reference exploration discussed in Sec. 1, we adopt the common independent layer-wise pruning (Wan et al., 2024) as the baseline framework. Specifically, each token $\boldsymbol{C}_j^l$ is assigned an importance score s_j^l, and the top-B candidates are retained accordingly as:

$$\hat{\boldsymbol{C}}^{t\le t^{\mathrm{p}},l} = \{\boldsymbol{C}_j^l | j \in \mathrm{TopK}(\{s_j^l\}_{j=1}^{N}, B)\}. \tag{2}$$

As established in prior work (Lin et al., 2026), the attention from denoising visual tokens to reference tokens effectively reflects the native information flow during image synthesis, providing a principled criterion for token pruning. Formally, let $\boldsymbol{Q}=\{\boldsymbol{q}_m\}_{m=1}^{M}$ denote the queries projected from $\boldsymbol{Z}_{t^{\mathrm{p}}}^l$, and $\boldsymbol{K}=\{\boldsymbol{k}_j\}_{j=1}^{N}$ the keys of $\boldsymbol{C}^l$, with the superscript l omitted for clarity. Accordingly, the baseline importance score s_j^l is defined as the average attention received by token j over all queries:

$$s_j^l = \frac{1}{M}\sum\nolimits_{\boldsymbol{q}\in\boldsymbol{Q}} \mathrm{Attn}(\boldsymbol{q})_j, \quad \mathrm{Attn}(\boldsymbol{q}) = \mathrm{Softmax}\left(\frac{\boldsymbol{q}\boldsymbol{K}^\top}{\sqrt{d}}\right) \in \mathbb{R}^N, \tag{3}$$

where d denotes head dimension, and we omit the head-wise averaging from the notation for brevity.

Building upon this general framework, CATP focuses on accelerating the computation-dominant denoising process, as illustrated in Fig. 2. By further investigating the aforementioned limitations of vanilla attention-based pruning, we develop GCP to determine $\hat{\boldsymbol{C}}^{t\le t^{\mathrm{p}},l}$ across condition branches and TCAM with FSO to refine s_j^l, thereby enabling more accurate and efficient context compression.

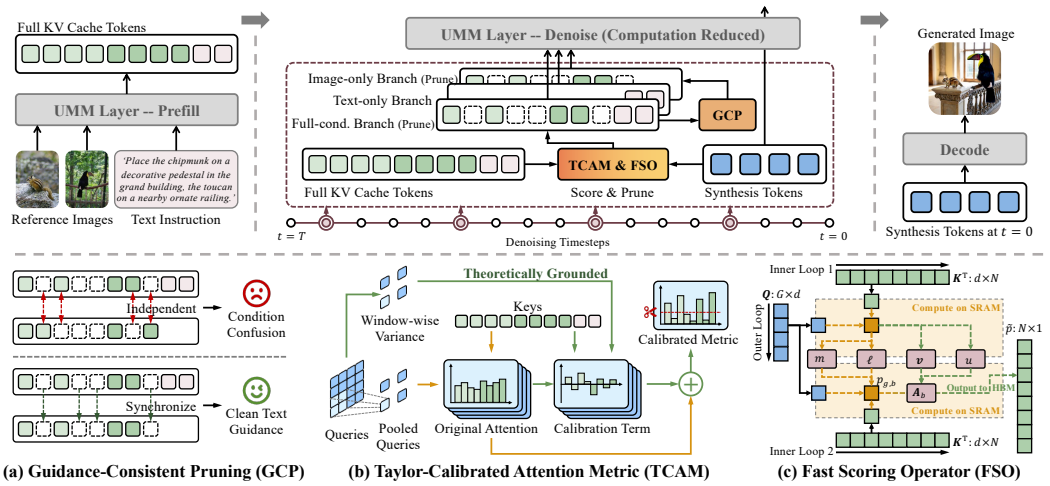


Figure 2: Framework overview. (Top) UMM inference pipeline with CATP. It accelerates denoising by pruning visual KV cache tokens across layers at predefined timesteps. (Bottom) Illustration of the main modules. (a) GCP synchronizes pruning decisions from the full-condition to image-only branch, avoiding condition confusion from independent pruning. (b) TCAM estimates token importance using pooled-query attention and calibrates it with a theoretically grounded second-order term. (c) FSO efficiently realizes calibrated scoring via a two-pass tiled online-Softmax kernel.

3.2 Guidance-Consistent Pruning

The native CFG mechanism (Ho & Salimans, 2021) instantiates Eq. (1) into three condition branches, each performing a complete forward pass under full, image-only or text-only conditioning:

$$\boldsymbol{V}_t^{\text{full}} = v_\theta(\boldsymbol{Z}_t, t; \boldsymbol{I}, \boldsymbol{T}), \quad \boldsymbol{V}_t^{\text{img}} = v_\theta(\boldsymbol{Z}_t, t; \boldsymbol{I}, \varnothing), \quad \boldsymbol{V}_t^{\text{text}} = v_\theta(\boldsymbol{Z}_t, t; \varnothing, \boldsymbol{T}), \tag{4}$$

where $\boldsymbol{V}$ denotes the velocity field, and $v_\theta(\cdot)$ is the velocity predictor of $f_\theta(\cdot)$. This yields modality-specific guidance residuals $\boldsymbol{G}_t^{\text{img}} = \boldsymbol{V}_t^{\text{full}} - \boldsymbol{V}_t^{\text{text}}$ and $\boldsymbol{G}_t^{\text{text}} = \boldsymbol{V}_t^{\text{full}} - \boldsymbol{V}_t^{\text{img}}$, capturing the incremental contribution of each modality. Accordingly, the guided velocity can be equivalently formulated as $\boldsymbol{V}_t^{\text{cfg}}$, controlled by model-inherent weights λ^{text} and λ^{img}, and then used for denoising update:

$$\boldsymbol{V}_t^{\text{cfg}} = \boldsymbol{V}_t^{\text{full}} + \lambda^{\text{img}} \boldsymbol{G}_t^{\text{img}} + \lambda^{\text{text}} \boldsymbol{G}_t^{\text{text}}, \quad \boldsymbol{Z}_{t-1} = \boldsymbol{Z}_t - \Delta t \boldsymbol{V}_t^{\text{cfg}}. \tag{5}$$

Notably, reference image tokens, which are the primary pruning targets, are shared by both the full-condition and image-only branches. However, as shown in Fig. 1(b), the straightforward strategy of pruning each branch using its native online attention scores yields mismatched token selections. Formally, let $\boldsymbol{I}^{\text{full}}$ and $\boldsymbol{I}^{\text{img}}$ denote the visual tokens retained in the corresponding branches, with the layer index omitted for simplicity. The resulting text-guidance can be decomposed as:

$$\tilde{\boldsymbol{G}}_t^{\text{text}} = \underbrace{v_\theta(\boldsymbol{Z}_t, t; \boldsymbol{I}^{\text{full}}, \boldsymbol{T}) - v_\theta(\boldsymbol{Z}_t, t; \boldsymbol{I}^{\text{full}}, \varnothing)}_{\text{clean text guidance}} + \underbrace{v_\theta(\boldsymbol{Z}_t, t; \boldsymbol{I}^{\text{full}}, \varnothing) - v_\theta(\boldsymbol{Z}_t, t; \boldsymbol{I}^{\text{img}}, \varnothing)}_{\boldsymbol{\delta}_t^{\text{conf}}}. \tag{6}$$

Here, the first term remains clean and valid text guidance, whereas an unexpected condition confusion arises from the second term $\boldsymbol{\delta}_t^{\text{conf}}$, as it introduces a spurious guidance signal from the mismatched visual token sets, further amplified by the non-negligible scale λ^{text}.

To address this issue, we adopt a simple yet effective strategy that synchronizes pruning across branches sharing the same visual context. Concretely, token importance is estimated from the full-condition branch, which has access to complete context, and the resulting pruning decision is reused by the image-only branch, yielding:

$$\hat{\boldsymbol{I}}^{\text{img},t,l} \leftarrow \hat{\boldsymbol{I}}^{\text{full},t,l}, \ \forall\, t, l. \tag{7}$$

This eliminates the confusion term as $\boldsymbol{\delta}_t^{\text{conf}} = 0$, ensuring the clean textual guidance as intended.

3.3 Taylor-Calibrated Attention Metric

Pooling-Induced Bias. Without loss of generality, we focus on a single layer, writing s_j^l in Eq. (3) as s_j for brevity. Although faithful, the sequence length of $\boldsymbol{Q}$ incurs non-negligible computation, un-

dermining the acceleration objective. A common remedy is to spatially downsample $\boldsymbol{Q}$ (Chen et al., 2026; Zhang et al., 2025a). Concretely, $\boldsymbol{Q}$ is partitioned into $G = M/w^2$ non-overlapping $w \times w$ windows $\{\mathcal{W}_g\}_{g=1}^{G}$, each represented by an average-pooled query $\bar{\boldsymbol{q}}_g = \frac{1}{w^2}\sum_{\boldsymbol{q}\in\mathcal{W}_g}\boldsymbol{q}$. This replaces the costly scoring over $\boldsymbol{Q}$ in Eq. (3) with a compact query set, yielding $\bar{s}_j = \frac{1}{G}\sum_{g=1}^{G}\mathrm{Attn}(\bar{\boldsymbol{q}}_g)_j$.

However, interchanging averaging with the nonlinear Softmax operation introduces a window-wise numerical bias between s_j and $\bar{s}_j$. Considering a single window $\mathcal{W}_g$, the exact discrepancy is:

$$\Delta_{g,j} = \frac{1}{w^2}\sum_{\boldsymbol{q}\in\mathcal{W}_g}\mathrm{Attn}(\boldsymbol{q})_j - \mathrm{Attn}(\bar{\boldsymbol{q}}_g)_j. \tag{8}$$

Taylor-Based Metric Calibration. Given Eq. (8), our goal is to derive a lightweight estimate of $\Delta_{g,j}$ that compensates $\bar{s}_j$ toward s_j, thereby improving the accuracy of token selection.

First, let $\boldsymbol{\epsilon}_{g,i} = \boldsymbol{q}_{g,i} - \bar{\boldsymbol{q}}_g$ denote the deviation of each query $\boldsymbol{q}_{g,i} \in \mathcal{W}_g$ from $\bar{\boldsymbol{q}}_g$. Due to the spatial locality of visual features (Liu et al., 2021), $\boldsymbol{\epsilon}_{g,i}$ is typically small, motivating a Taylor approximation of $\mathrm{Attn}(\cdot)_j$ around $\bar{\boldsymbol{q}}_g$. Defining $p_{g,j} = \mathrm{Attn}(\bar{\boldsymbol{q}}_g)_j$, and the second-order expansion gives:

$$\mathrm{Attn}(\boldsymbol{q}_{g,i})_j \approx p_{g,j} + \nabla p_{g,j}^{\top}\boldsymbol{\epsilon}_{g,i} + \frac{1}{2}\boldsymbol{\epsilon}_{g,i}^{\top}\boldsymbol{H}_{g,j}\boldsymbol{\epsilon}_{g,i}, \tag{9}$$

where $\nabla p_{g,j}$ and $\boldsymbol{H}_{g,j}$ are the gradient vector and Hessian matrix of $\mathrm{Attn}(\cdot)_j$ at $\bar{\boldsymbol{q}}_g$, respectively. Subsequently, averaging Eq. (9) over $\mathcal{W}_g$ yields an approximation of Eq. (8) as:

$$\Delta_{g,j} \approx \frac{1}{2}\,\mathrm{Tr}(\boldsymbol{H}_{g,j}\boldsymbol{\Sigma}_g), \quad \boldsymbol{\Sigma}_g = \frac{1}{w^2}\sum_{i=1}^{w^2}\boldsymbol{\epsilon}_{g,i}\boldsymbol{\epsilon}_{g,i}^{\top}, \tag{10}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace of a matrix, and $\boldsymbol{\Sigma}_g$ is the covariance matrix of query deviations.

Furthermore, define the attention-weighted key mean as $\bar{\boldsymbol{k}}_g = \sum_{j=1}^{N} p_{g,j}\boldsymbol{k}_j$ and denote the diagonal entries of $\boldsymbol{\Sigma}_g$ by $(\sigma_{g,1}^2, \ldots, \sigma_{g,d}^2)$. As detailed in Sec. B.1, $\boldsymbol{H}_{g,j}$ admits a closed-form expression in Eq. (11), while the weak cross-channel correlations enable $\boldsymbol{\Sigma}_g \approx \sigma_g^2\boldsymbol{I}$, where $\sigma_g^2 = \frac{1}{d}\sum_{c=1}^{d}\sigma_{g,c}^2$.

$$\boldsymbol{H}_{g,j} = \frac{p_{g,j}}{d}[(\boldsymbol{k}_j - \bar{\boldsymbol{k}}_g)(\boldsymbol{k}_j - \bar{\boldsymbol{k}}_g)^{\top} - \sum_{n=1}^{N} p_{g,n}(\boldsymbol{k}_n - \bar{\boldsymbol{k}}_g)(\boldsymbol{k}_n - \bar{\boldsymbol{k}}_g)^{\top}]. \tag{11}$$

Substituting these results into Eq. (10) yields an efficient computable approximation:

$$\Delta_{g,j} \approx \frac{p_{g,j}}{2d}(r_{g,j} - \bar{r}_g), \quad r_{g,j} = \sigma_g^2\|\boldsymbol{k}_j - \bar{\boldsymbol{k}}_g\|_2^2, \quad \bar{r}_g = \sum_{n=1}^{N} p_{g,n}r_{g,n}. \tag{12}$$

Finally, aggregating the window-wise corrections formulates the calibrated importance score $\tilde{s}_j$ as Eq. (13), where we introduce the hyperparameter λ to control the strength of correction.

$$\tilde{s}_j = \frac{1}{G}\sum_{g=1}^{G}\tilde{p}_{g,j}, \quad \tilde{p}_{g,j} = p_{g,j} + \lambda\frac{p_{g,j}}{2d}(r_{g,j} - \bar{r}_g). \tag{13}$$

Remark. This derivation indicates that pooling-induced bias is jointly governed by local query variation and key dispersion within the corresponding attention view. Larger query variance implies greater information loss from pooling, while larger centered key distance makes attention response more sensitive to such variation, both requiring stronger correction. This also echoes prior heuristic use of variance for regional importance (Fang et al., 2025; Wu et al., 2025b), while our derivation further reveals its theoretical role in modulating importance rather than directly estimating it.

3.4 Fast Scoring Operator

While parallelizable across windows and keys, the calibration still involves a sequential dependency $p_{g,j} \to \bar{\boldsymbol{k}}_g \to r_{g,j} \to \bar{r}_g \to \tilde{p}_{g,j}$, where the repeated $p_{g,j}$-weighted aggregations require multiple accesses to large-scale matrices, incurring non-negligible memory traffic. Therefore, we develop a tiled window-wise scoring operator that fuses the correction into the native computation of $p_{g,j}$.

First, as detailed in the Sec. B.2, $\tilde{p}_{g,j}$ can be equivalently reformulated as:

$$\tilde{p}_{g,j} = p_{g,j}[1 + \frac{\lambda\sigma_g^2}{2d}(\|\boldsymbol{k}_j\|_2^2 - 2\boldsymbol{k}_j^{\top}\bar{\boldsymbol{k}}_g + 2\|\bar{\boldsymbol{k}}_g\|_2^2 - \sum_{n=1}^{N} p_{g,n}\|\boldsymbol{k}_n\|_2^2)]. \tag{14}$$

This avoids the explicit dependence on $\bar{r}_g$, allowing the required statistics to be accumulated alongside online Softmax. Concretely, as summarized in Algo. 1, we adopt a tiled two-pass procedure

Algorithm 1 Inner Loop of the Fast Scoring Operator (FSO) (Native attention; Taylor correction).

Input: Pooled query $\bar{\boldsymbol{q}}_g$, variance σ_g^2, key matrix $\boldsymbol{K} \in \mathbb{R}^{N\times d}$ in HBM; $\boldsymbol{K}$ is partitioned into $T_K = \lceil N/B_K \rceil$ blocks $\{\boldsymbol{K}_b\}_{b=1}^{T_K}$, each of size $B_K \times d$; hyperparameter λ; $\frac{1}{\sqrt{d}}$ is absorbed into $\boldsymbol{K}$ and omitted for brevity.

Output: Calibrated importance scores $\tilde{\boldsymbol{p}}_g \in \mathbb{R}^N$ in HBM.

1: Load $\bar{\boldsymbol{q}}_g, \sigma_g^2$ from HBM to on-chip SRAM.
2: On chip, initialize $(m_0, \ell_0) = (-\infty, 0)$ and $(\boldsymbol{v}, u) = (\boldsymbol{0}, 0)$ in SRAM.
3: *Pass 1: online Softmax with key-moment accumulation.*
4: **for** $b = 1, \ldots, T_K$ **do**
5: Load $\boldsymbol{K}_b$ from HBM to on-chip SRAM.
6: On chip, compute $\boldsymbol{z}_b \leftarrow \bar{\boldsymbol{q}}_g \boldsymbol{K}_b^\top$, $m_b \leftarrow \max(m_{b-1}, \max(\boldsymbol{z}_b))$.
7: On chip, compute $\boldsymbol{e}_b \leftarrow \exp(\boldsymbol{z}_b - m_b)$, $\alpha \leftarrow \exp(m_{b-1} - m_b)$, $\ell_b \leftarrow \alpha \ell_{b-1} + \mathrm{sum}(\boldsymbol{e}_b)$.
8: On chip, compute $\boldsymbol{v} \leftarrow \alpha \boldsymbol{v} + \boldsymbol{e}_b \boldsymbol{K}_b$, $u \leftarrow \alpha u + \boldsymbol{e}_b \,\mathrm{RowSqNorm}(\boldsymbol{K}_b)^\top$.
9: **end for**
10: On chip, compute $\bar{\boldsymbol{k}}_g \leftarrow \boldsymbol{v}/\ell_{T_K}$, $u \leftarrow u/\ell_{T_K}$.
11: *Pass 2: probability recomputation with Taylor calibration.*
12: **for** $b = 1, \ldots, T_K$ **do**
13: Reload $\boldsymbol{K}_b$ from HBM to on-chip SRAM.
14: On chip, compute $\boldsymbol{z}_b \leftarrow \bar{\boldsymbol{q}}_g \boldsymbol{K}_b^\top$, $\boldsymbol{p}_{g,b} \leftarrow \exp(\boldsymbol{z}_b - m_{T_K})/\ell_{T_K}$.
15: On chip, compute $\boldsymbol{A}_b \leftarrow \mathrm{RowSqNorm}(\boldsymbol{K}_b) - 2\bar{\boldsymbol{k}}_g \boldsymbol{K}_b^\top + 2\|\bar{\boldsymbol{k}}_g\|_2^2 - u$, $\tilde{\boldsymbol{p}}_{g,b} \leftarrow \boldsymbol{p}_{g,b} \odot (1 + \frac{\lambda}{2}\sigma_g^2 \boldsymbol{A}_b)$.
16: Accumulate $\tilde{\boldsymbol{p}}_{g,b}$ into the b-th tile of $\tilde{\boldsymbol{p}}_g$ in HBM.
17: **end for**
18: **return** $\tilde{\boldsymbol{p}}_g$

inspired by FlashAttention (Dao, 2024) and Sec. B.2. Notably, without materializing intermediate logits, two passes are minimally required to explicitly recover the normalized probabilities $p_{g,j}$.

During the first pass, the native logit maximum m_b and normalizer ℓ_b are updated online in Static Random-Access Memory (SRAM), and the attention-weighted key mean $\boldsymbol{v}$ and squared-norm statistic u are simultaneously accumulated under the same rescaling scheme, as Lines 6∼8. After traversing all key blocks, normalization by ℓ_{T_K} yields the exact $\bar{\boldsymbol{k}}_g$ and $\sum_{n=1}^{N} p_{g,n}\|\boldsymbol{k}_n\|_2^2 = u$, as Line 10.

The second pass reloads each key block and recomputes the exact $p_{g,j}$ utilizing the finalized statistics m_{T_K} and ℓ_{T_K}, as shown in Line 14. With all window-level statistics available, the correction then depends only on the current key-specific terms $\|\boldsymbol{k}_j\|_2^2$ and $\boldsymbol{k}_j^\top \bar{\boldsymbol{k}}_g$, enabling $\tilde{p}_{g,j}$ to be computed immediately as Line 15 and accumulated back into High Bandwidth Memory (HBM) score tile.

Overall, only calibrated scores are materialized, avoiding separate attention and correction matrices.

4 Experimental Results and Analysis

4.1 Experimental Settings

Models. Following recent advances in UMMs, we evaluate methods on two representative models: Bagel (Deng et al., 2025) and SenseNova-U1 (Diao et al., 2026), with generation performed in latent and pixel spaces, respectively. For Bagel, we use the multi-image fine-tuned checkpoint from (Chen et al., 2026), while SenseNova-U1 adopts its official checkpoint with native multi-image support.

Datasets and Evaluation Metrics. We conduct experiments on three widely used multi-image synthesis benchmarks, following their official splits and metric aggregation rules. Concretely, we report the geometric mean of Image Consistency and Prompt Following scores on the MacroBench (Chen et al., 2026) Customization subset across four reference-number groups (1–3, 4–5, 6–7, and 8–10), the Weighted-Ref-VIEScore for Object-Centric, Human-Centric, Human-Object Interaction (HOI), and Decomposition-and-Recomposition (De&Re) tasks on MICo-Bench (Wei et al., 2026), and the geometric mean of Prompt Following and Subject Consistency scores for the Single/Multiple/Scene categories of OmniContext (Wu et al., 2025a), each further divided into Character (Char.)/Object (Obj.)/Char.+Obj. subcategories. We also include the average score (Avg.) for MacroBench and OmniContext, and the Overall score for MICo-Bench. For metrics involving MLLM-based evaluation, we utilize the open-source Qwen3-VL-32B-Instruct (Bai et al., 2025a) as the evaluator.

Implementation Details. We implement all methods on top of the TorchUMM library (Luo et al., 2026), with FlashAttention (Dao, 2024) the acceleration baseline. Since visual tokens dominate the

Table 1: Comparison of multi-image synthesis performance scores by various methods for Bagel, evaluated on the MacroBench and MICo-Bench datasets. The best results are highlighted in **bold**.

Method	MacroBench					MICo-Bench				
	1–3	4–5	6–7	8–10	Avg.↑	Object	Human	HOI	De&Re	Overall↑
Full Model	8.89	8.51	7.97	7.20	8.14	47.52	40.68	42.47	53.69	46.66
retain 25% tokens										
Text-Aligned	8.21	8.22	8.07	7.44	7.98	52.11	35.77	46.95	52.92	47.64
VisionZip	8.18	8.07	8.00	6.97	7.81	49.65	22.99	38.79	56.40	43.38
SDTM	8.13	7.99	7.68	7.50	7.83	38.96	24.79	25.86	41.52	32.90
ToPi	8.61	8.39	8.12	7.49	8.15	53.48	**40.17**	46.00	**63.69**	51.97
CATP (Ours)	**8.76**	**8.69**	**8.51**	**7.87**	**8.46**	**54.81**	39.87	**49.49**	63.61	**53.22**
retain 10% tokens										
Text-Aligned	7.09	7.30	7.53	6.95	7.22	49.42	23.60	39.72	55.18	43.35
VisionZip	5.44	6.02	6.25	5.70	5.85	37.43	13.92	27.58	46.95	33.00
SDTM	7.21	7.08	6.81	6.47	6.89	31.45	15.18	18.82	28.81	23.41
ToPi	7.32	7.18	7.28	6.86	7.16	46.89	29.29	38.81	56.60	44.20
CATP (Ours)	**8.40**	**8.33**	**8.31**	**7.54**	**8.15**	**52.01**	**29.51**	**42.88**	**58.04**	**46.84**

Table 2: Comparison of performance scores by various methods for Bagel/SenseNova-U1 on the OmniContext dataset with 25% tokens retained. The best results are highlighted in **bold**.

Method	Single		Multiple			Scene			Avg.↑
	Char.	Obj.	Char.	Obj.	Char.+Obj.	Char.	Obj.	Char.+Obj.	
Bagel (Latent Space)									
Full Model	8.50	9.28	7.97	9.22	8.60	7.49	8.21	8.15	8.43
Text-Aligned	8.16	8.77	**7.56**	**9.10**	8.25	7.46	8.23	7.64	8.14
VisionZip	7.06	7.90	6.78	8.64	7.97	7.40	8.50	7.58	7.73
SDTM	7.16	8.32	6.44	7.10	7.56	6.32	7.45	6.97	7.16
ToPi	6.98	8.61	6.82	8.89	8.40	7.08	8.42	8.02	7.90
CATP (Ours)	**8.45**	**9.10**	7.55	8.98	**8.59**	**7.57**	**8.72**	**8.26**	**8.40**
SenseNova-U1 (Pixel Space)									
Full Model	7.75	7.78	7.71	8.85	8.10	7.12	7.06	7.06	7.68
Text-Aligned	7.11	6.96	7.69	8.60	7.43	6.97	7.80	7.32	7.48
VisionZip	7.34	7.51	7.31	8.66	7.92	7.06	7.21	6.89	7.49
SDTM	7.02	7.73	6.84	8.01	7.46	6.89	6.69	6.22	7.11
ToPi	6.09	6.68	6.58	7.40	7.18	6.28	**7.94**	6.81	6.87
CATP (Ours)	**7.42**	**8.06**	**7.76**	**8.76**	**8.40**	**7.52**	7.69	**7.63**	**7.91**

multimodal context, pruning is conducted exclusively on them, including both ViT and VAE tokens in Bagel and patch-level visual tokens in SenseNova-U1. By following (Lin et al., 2026), pruning is performed over the full sequence at denoising timesteps $t^{\mathrm{P}} \in \{49, 39, 29, 19, 9\}$. Identical retention ratio is applied to each reference image. In all experiments, we fix $\lambda = 0.5$ and $w = 4$ for TCAM and $B_K = 64$ for FSO. All other inference settings, *e.g.*, sampling and guidance parameters, follow the original model configurations. All experiments are conducted on one NVIDIA A800 (80GB) GPU.

4.2 Main Results

Since token pruning for UMM generation remains rarely explored, we adapt representative baselines from closely related paradigms underlying UMMs, *i.e.*, MLLMs and DiTs. Specifically, Text-Aligned Pruning is previously applied to UMM-based multi-image synthesis by (Chen et al., 2026). Both it and VisionZip (Yang et al., 2025) are originally developed for MLLM understanding tasks. SDTM (Fang et al., 2025) and ToPi (Lin et al., 2026) are DiT-oriented methods. For fair comparison, we restrict all methods to attention acceleration and adjust each to an equivalent computation reduction as CATP under predefined token retention ratios.

Quantitative Results. As shown in Tab. 1, CATP consistently outperforms compared methods across varying numbers of reference images and diverse generation tasks. Specifically, it surpasses



Figure 3: Qualitative comparison of various methods on Bagel at a 25% token retention ratio. The failure/well-preserved regions are marked with red/green boxes.

the second-best methods by 0.31 and 1.25 points on MacroBench and MICo-Bench, respectively at a 25% retention ratio, and further increases the advantages to 0.93 and 2.64 points respectively at 10% retention ratio, demonstrating more pronounced robustness under aggressive context compression.

Notably, CATP even outperforms the original full model on MacroBench Avg. at a 25% retention ratio, with gains primarily arising in multi-image settings ($\geq$ 6 images) and reaching up to 0.67 points. This stems from the effective redundancy removal. As discussed in prior work, the intrinsic normalization property of attention can dilute individual weights in excessively long sequences (Wu et al., 2026a), making critical reference regions harder to distinguish and thereby yielding confused spatial layouts or degraded fine-grained details. Accordingly, pruning redundant context mitigates this effect by reducing unnecessary attention competition (Chen et al., 2026), while CATP further amplifies this benefit by more accurately retaining generation-critical regions.

Generalization Across Denoising Spaces. Given recent advances in pixel-space UMMs, we further adapt methods to the representative SenseNova-U1 and evaluate on OmniContext. As shown in Tab. 2, CATP maintains notable advantages of 0.26 and 0.42 Avg. points on Bagel and SenseNova-U1 at 25% token retention, respectively, with the margins further widening to 0.5 and 1.2 at 10% retention in Tab. A. These results demonstrate its impressive generalizability across denoising spaces.

Qualitative Results. As shown in Fig. 3, CATP delivers more favorable overall generation quality, particularly in spatial composition, human identity preservation, and critical background details. This stems from the joint effects of our designs, where GCP improves generation fidelity through coherent guidance, and TCAM better preserves fine-grained identity cues by assigning stronger importance corrections to regions with more details, making them more likely to survive pruning.

Further Remarks. Text-Aligned relies on attention signals from text or within reference images, which primarily reflect multimodal understanding rather than direct generation relevance. VisionZip merges similar visual regions, favoring coarse structural preservation but potentially losing identity-critical details. SDTM compresses only denoising tokens, making it better suited to simple generation scenarios but prone to substantial degradation when targeting higher acceleration. While ToPi explicitly targets context compression for generation, it reuses pruning decisions across layers to reduce scoring overhead, overlooking layer-wise differences in context exploration, and the fully computed anchor timesteps further limit the compression scale. In contrast, CATP develops a more accurate and efficient scoring process, enabling layer-wise token evaluation and selection. Under aggressive compression, different layers can exploit complementary critical regions rather than maintaining complete target content throughout, thereby maintaining overall generation quality.

Efficiency. We further assess the acceleration effect of CATP, with FlashAttention as the baseline. For each image count, we fix 1024-resolution inputs, sample 10 examples and report the average latency of the accelerated attention cores across all denoising calls. As shown in Fig. 4(a), atten-

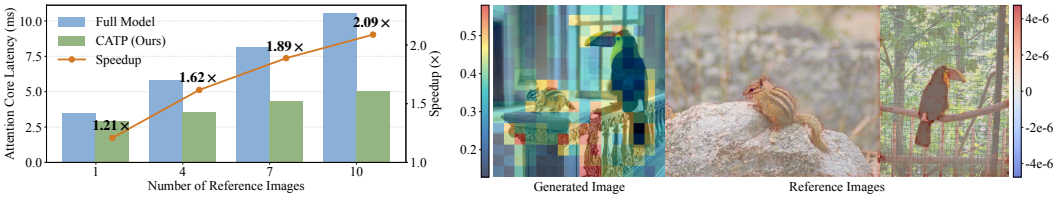


Figure 4: (a) Attention core latency of the full model and CATP, along with the relative speedup of CATP, across varying numbers of reference images. (b) Visualization of (Left) window-wise query variance and (Right) the resulting aggregated calibration term over reference images in TCAM.

Table 3: Ablation of GCP and TCAM modules for Bagel on MacroBench at two retention ratios.

GCP	TCAM	Retain 25% Tokens					Retain 10% Tokens				
		1–3	4–5	6–7	8–10	Avg.↑	1–3	4–5	6–7	8–10	Avg.↑
		8.32	8.19	8.14	7.62	8.07	6.65	6.87	7.25	6.85	6.91
✓		8.75	8.65	8.36	7.45	8.30	8.23	8.28	**8.39**	7.38	8.07
✓	✓	**8.76**	**8.69**	**8.51**	**7.87**	**8.46**	**8.40**	**8.33**	8.31	**7.54**	**8.15**

tion computation becomes increasingly dominant with more reference images, translating context compression into larger practical gains, reaching 2.09× with 10 reference images at 25% retention.

4.3 Ablation Study

On GCP. As shown in Tab. 3, GCP effectively improves conditional-guidance consistency, yielding average gains of 0.23 and 1.16 points at 25% and 10% retention ratios, respectively. Intuitively, more aggressive pruning makes the symmetric difference between conditioning branches increasingly dominated by less important tokens, introducing stronger irrelevant signals into the guidance residual. Therefore, GCP becomes more beneficial at lower retention ratios.

On TCAM. Tab. 3 also indicates that TCAM further improves gains under both settings. As visualized in Fig. 4(b), high-variance denoising regions concentrate around object boundaries and identity-critical details, consistent with the widely recognized properties in prior studies (Fang et al., 2025). TCAM exploits this signal via second-order calibration to strengthen corresponding reference regions, thus promoting the retention of identity-critical tokens for finer-grained synthesis.

On FSO. We further assess the effect of FSO in Tab. 4, by measuring scoring latency with ten 1024-resolution reference images. Concretely, dense attention achieves the best generation quality but incurs prohibitive pruning cost, while query pooling substantially reduces latency at the cost of degraded performance. Given TCAM's additional calibration cost, FSO finally mitigates this by algebraically reformulating it into a fused operator, reducing latency to 5.39 ms without sacrificing quality. Notably, FSO is even faster than the pooling baseline, which owes to fine-grained HBM-level optimization enabling BF16 multiplication with FP32 accumulation.

Table 4: Effect of TCAM and FSO modules on scoring latency (ms) and MacroBench average score (Avg.).

Pooling	TCAM	FSO	Latency↓	Avg.↑
			102.97	8.48
✓			7.24	8.30
✓	✓		30.12	8.46
✓	✓	✓	5.39	8.46

5 Conclusion

In this work, we conduct a pioneering investigation of visual token pruning for efficient UMM multi-image synthesis, and develop a novel training-free method, namely CATP. By performing layer-wise pruning on KV cache tokens, CATP enables adaptive utilization of different reference information. Furthermore, we explicitly preserve clean conditional guidance through Guidance-Consistent Pruning (GCP) and improve the discriminability of critical regions with a theoretically derived Taylor-Calibrated Attention Metric (TCAM), together with a Fast Scoring Operator (FSO) that reduces the scoring cost. Experiments on prevalent latent- and pixel-space UMMs across three multi-image benchmarks demonstrate the effectiveness of CATP, consistently outperforming competitive token-reduction approaches across retention ratios while achieving up to 2.09× attention-core speedup.

AI USE STATEMENT

In this work, we used generative AI tools for assist in the writing of proofs, assist with translation. We have not used generative AI tools for help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, interpret results, and generate synthetic data sets, clean and reformat dataset, support qualitative and thematic data analysis are not applicable to this work. Additionally, we used generative AI tools for sourcing/searching for information, identify relevant literature.

We have reviewed all AI-assisted work. Specifically, we independently verified all AI-polished mathematical formulations and proof-related content step by step to ensure their correctness and consistency with the underlying mathematical definitions. AI-assisted translations were manually reviewed by 2 authors to preserve the intended meaning. Information obtained through AI-assisted search was cross-checked against reliable sources, and all AI-identified literature was verified by consulting the corresponding original publications before being cited or discussed in this manuscript.

We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9392–9401, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025b.

Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 4599–4603, 2023.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *Proceedings of the International Conference on Learning Representations*, 2023.

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Proceedings of the International Conference on Machine Learning*, volume 235, pp. 6562–6595, 2024a.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Proceedings of the European Conference on Computer Vision*, pp. 19–35. Springer, 2024b.

Zhekai Chen, Yuqing Wang, Manyuan Zhang, and Xihui Liu. Macro: Advancing multi-reference image generation with structured long-context data. *arXiv preprint arXiv:2603.25319*, 2026.

Yisol Choi, Sangkyung Kwak, Kyungmin Lee, Hyungwon Choi, and Jinwoo Shin. Improving diffusion models for authentic virtual try-on in the wild. In *Proceedings of the European Conference on Computer Vision*, pp. 206–235. Springer, 2024.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *Proceedings of the International Conference on Learning Representations*, 2024.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Haiwen Diao, Penghao Wu, Hanming Deng, Jiahao Wang, Shihao Bai, Silei Wu, Weichen Fan, Wenjie Ye, Wenwen Tong, Xiangyu Fan, et al. Sensenova-u1: Unifying multimodal understanding and generation with neo-unify architecture. *arXiv preprint arXiv:2605.12500*, 2026.

Xiangyu Fan, Zesong Qiu, Zhuguanyu Wu, Fanzhou Wang, Zhiqian Lin, Tianxiang Ren, Dahua Lin, Ruihao Gong, and Lei Yang. Phased dmd: Few-step distribution matching distillation via score matching within subintervals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41667–41676, 2026.

Haipeng Fang, Sheng Tang, Juan Cao, Enshuo Zhang, Fan Tang, and Tong-Yee Lee. Attend to not attended: Structure-then-detail token merging for post-training dit acceleration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18083–18092, 2025.

Google DeepMind. Gemini 3 flash: Model card. Google DeepMind Model Card, 2025. URL `https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Flash-Model-Card.pdf`.

Xin He, Longhui Wei, Jianbo Ouyang, Minghui Liao, Lingxi Xie, and Qi Tian. Emma: Efficient multimodal understanding, generation, and editing with a unified architecture. *arXiv preprint arXiv:2512.04810*, 2025.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS Workshop on Deep Generative Models and Downstream Applications*, 2021.

Junlong Ke, Zichen Wen, Boxue Yang, Yantai Yang, Xuyang Liu, Chenfei Liao, Zhaorun Chen, Shaobo Wang, and Linfeng Zhang. Flash-unified: A training-free and task-aware acceleration framework for native unified models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Findings*, pp. 9131–9142, 2026.

Junqing Lin, Xingyu Zheng, Pei Cheng, Bin Fu, Jingwei Sun, and Guangzhong Sun. Token pruning for in-context generation in diffusion transformers. *arXiv preprint arXiv:2602.01609*, 2026.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26286–26296, 2024.

Jiacheng Liu, Chang Zou, Yuanhuiyi Lyu, Junjie Chen, and Linfeng Zhang. From reusing to forecasting: Accelerating diffusion models with taylorseers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15853–15863, 2025.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF international conference on computer vision (ICCV)*, pp. 9992–10002. Ieee, 2021.

Yinyi Luo, Wenwen Wang, Hayes Bai, Hongyu Zhu, Hao Chen, Pan He, Marios Savvides, Sharon Li, and Jindong Wang. Torchumm: A unified multimodal model codebase for evaluation, analysis, and post-training. *arXiv preprint arXiv:2604.10784*, 2026.

Yuta Oshima, Daiki Miyake, Kohsei Matsutani, Yusuke Iwasawa, Masahiro Suzuki, Yutaka Matsuo, and Hiroki Furuta. Multibanana: A challenging benchmark for multi-reference text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 448–460, 2026.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. In *Findings of the Association for Computational Linguistics*, pp. 4065–4078, 2024.

Xierui Wang, Siming Fu, Qihan Huang, Wanggui He, and Hao Jiang. Ms-diffusion: Multi-subject zero-shot image personalization with layout guidance. In *Proceedings of the International Conference on Learning Representations*, volume 2025, pp. 95118–95146, 2025.

Xinyu Wei, Kangrui Cen, Hongyang Wei, Zhen Guo, Bairui Li, Zeqing Wang, Jinrui Zhang, and Lei Zhang. Mico-150k: A comprehensive dataset advancing multi-image composition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29695–29706, 2026.

Chenyuan Wu, Jiahao Wang, Pengfei Zheng, et al. Omnigen2: Towards instruction-aligned multimodal generation. *arXiv preprint arXiv:2506.18871*, 2025a.

Zimeng Wu, Jiaxin Chen, and Yunhong Wang. Unified knowledge maintenance pruning and progressive recovery with weight recalling for large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 8550–8558, 2025b.

Zimeng Wu, Donghao Wang, Chaozhe Jin, Jiaxin Chen, and Yunhong Wang. Probe and skip: Self-predictive token skipping for efficient long-context llm inference. *arXiv preprint arXiv:2601.13155*, 2026a.

Zimeng Wu, Yunhong Wang, Donghao Wang, and Jiaxin Chen. Collaborative multi-mode pruning for vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39561–39571, 2026b.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Linli Yao, Long Xing, Yang Shi, Sida Li, Yuanxin Liu, Yuhao Dong, Yi-Fan Zhang, Lei Li, Qingxiu Dong, Xiaoyi Dong, et al. Towards efficient multimodal large language models: A survey on token compression. 2026.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Haocheng Xi, Jun Zhu, Jianfei Chen, et al. Spargeattention: Accurate and training-free sparse attention accelerating any model inference. In *Proceedings of the International Conference on Machine Learning*, 2025a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *Proceedings of the International Conference on Machine Learning*, 2025b.

Qicheng Zhao, Qi Sun, and Zheyu Yan. Seeing the end at step zero: Accelerating diffusion mllms via mlp sparsity-aware truncation. *arXiv preprint arXiv:2607.14557*, 2026.

Shanshan Zhao, Xinjie Zhang, Jintao Guo, Jiakui Hu, Lunhao Duan, Minghao Fu, Yong Xien Chng, Guo-Hua Wang, Qing-Guo Chen, Zhao Xu, et al. Unified multimodal understanding and generation models: Advances, challenges, and opportunities. *arXiv preprint arXiv:2505.02567*, 2025.

A More Experimental Results and Analysis

A.1 Additional Quantitative Results on OmniContext

To complement the 25% retention results reported in Tab. 2, we further evaluate all methods on OmniContext (Wu et al., 2025a) under a more aggressive 10% token retention ratio. As shown in Tab. A, CATP continues to achieve superior performance in most scenarios on both Bagel and SenseNova-U1, further widening its margin over the second-best methods to 0.5 and 1.2 points, respectively. These results provide additional evidence for the robustness of CATP under aggressive acceleration and its generalization across various UMM generation spaces.

Table A: Comparison of performance scores by various methods for Bagel/SenseNova-U1 on the OmniContext dataset with 10% tokens retained. The best results are highlighted in **bold**.

Method	Single		Multiple			Scene			Avg.↑
	Char.	Obj.	Char.	Obj.	Char.+Obj.	Char.	Obj.	Char.+Obj.	
Bagel (Latent Space)									
Full Model	8.50	9.28	7.97	9.22	8.60	7.49	8.21	8.15	8.43
Text-Aligned	**7.06**	7.36	5.17	**8.33**	7.33	6.13	7.95	7.26	7.07
VisionZip	4.38	4.62	3.67	7.50	4.01	4.27	7.11	6.06	5.20
SDTM	6.57	6.23	5.89	5.77	6.68	5.31	6.31	5.75	6.06
ToPi	5.33	6.65	4.16	7.92	6.90	5.15	**8.20**	6.89	6.38
CATP (Ours)	6.74	**8.19**	**6.78**	7.94	**8.25**	**6.89**	8.14	**7.61**	**7.57**
SenseNova-U1 (Pixel Space)									
Full Model	7.75	7.78	7.71	8.85	8.10	7.12	7.06	7.06	7.68
Text-Aligned	5.72	4.78	6.21	7.88	6.56	6.40	**7.94**	6.58	6.51
VisionZip	6.35	6.19	5.54	7.90	6.51	5.69	7.39	6.22	6.47
SDTM	6.36	7.27	6.46	6.90	6.63	6.01	5.76	5.41	6.35
ToPi	4.70	5.64	4.45	6.96	5.38	5.13	7.54	5.87	5.71
CATP (Ours)	**7.31**	**7.45**	**7.84**	**8.55**	**8.18**	**7.04**	7.86	**7.42**	**7.71**

A.2 Further Ablations

On Calibration Strength λ. Since TCAM adopts approximations to avoid computing higher-order terms, its second-order estimate captures the correction trend rather than an exact discrepancy. Therefore, we heuristically introduce λ in Eq. (13) to control the calibration strength, whose effect is ablated in Tab. B. As λ decreases, TCAM approaches the vanilla pooling baseline and provides insufficient emphasis on fine-grained reference regions, whereas an excessively large value biases selection toward high-variance regions and may suppress smoother yet relevant content. Given the overall performance peak at $\lambda = 0.5$, we empirically fix this value for all experiments.

Table B: Effect of the correction strength λ on MacroBench score for Bagel at a 25% token retention ratio. The default setting is underlined.

λ	1–3	4–5	6–7	8–10	Avg.↑
0	8.75	8.65	8.36	7.45	8.30
0.25	8.73	8.70	8.34	7.73	8.38
<u>0.50</u>	8.76	8.69	8.51	7.87	**8.46**
0.75	8.79	8.76	8.49	7.77	8.45
1.00	8.81	8.67	8.37	7.74	8.40
1.25	8.83	8.65	8.35	7.68	8.38

On Window Size w. We assess the effect of the pooling window size w in TCAM. As reported in Tab. C, smaller windows incur less loss of fine-grained cues and therefore yield better generation quality, but at the cost of a longer query sequence with higher scoring latency. Larger windows exhibit the opposite trade-off, reducing scoring overhead while sacrificing more local details. Ac-

cordingly, we empirically choose $w = 4$ as a balanced configuration and fix it across all experiments.

Table C: Effect of the window size w on MacroBench score and scoring latency (ms) for Bagel at a 25% token retention ratio. The default setting is underlined.

w	1–3	4–5	6–7	8–10	Avg.↑	Latency↓
2	8.75	8.73	8.49	7.98	8.49	17.87
<u>4</u>	8.76	8.69	8.51	7.87	8.46	5.39
8	8.68	8.57	8.27	7.68	8.30	2.31

Qualitative Ablation on Main Modules. Beyond the quantitative ablations in Tab. 3, we further visualize the effects of GCP and TCAM in Fig. A. With the general pruning scheme, independent token selection across conditioning branches amplifies irrelevant visual cues through CFG guidance, leading to implausible rendering or distorted content. GCP alleviates this issue by enforcing consistent pruning across branches. TCAM further improves generation by emphasizing information-rich reference regions, yielding local patterns and details that better match the references.

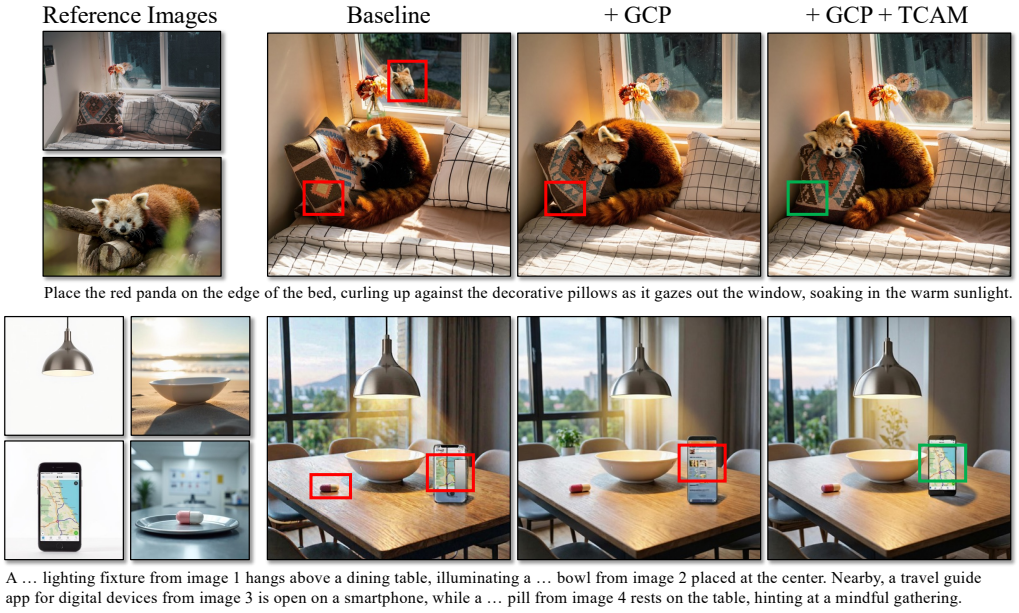


Figure A: Qualitative ablation of GCP and TCAM modules on Bagel at a 25% token retention ratio. The failure/well-preserved regions are marked with red/green boxes.

On Alternatives for Pooling-Induced Bias. For the pooling-induced bias discussed in Sec. 3.3, we further compare our solution (*i.e.* TCAM + FSO) with several alternative designs in Tab. D. intuitively, the inferior performance of the baseline Average Pooling stems from indiscriminate within-window averaging that smooths fine-grained cues. Therefore, a direct alternative is Max Pooling, however, channel-wise maxima may originate from different spatial tokens, disrupting the semantic coherence of the pooled representation and yielding even worse performance. Inspired by our derivation in Eq. (13), which suggests strengthening the contribution of high-variance denoising regions, we further consider an intuitive alternative that leaves queries from the highest-variance windows unpooled, denoted as Mixed Query. Its improved generation quality in Tab. D provides additional empirical support for our motivation, but the enlarged query sequence incurs increased scoring latency. In contrast, TCAM together with FSO achieves gains in both generation quality and scoring efficiency, further validating the effectiveness of our designs.

A.3 Robustness Across MLLM Evaluators

Noting that MLLM-based evaluation can exhibit evaluator-specific biases and judgment inconsistencies (Chen et al., 2024a), we therefore re-evaluate all methods on MacroBench utilizing two

Table D: Effect of various design choices for the pooling-induced bias on MacroBench for Bagel at a 25% retention ratio. The best results are highlighted in **bold** and the default setting is underlined.

Method	1–3	4–5	6–7	8–10	Avg.↑	Latency↓
Average Pooling	8.75	8.65	8.36	7.45	8.30	7.24
Max Pooling	8.59	8.56	8.29	7.50	8.24	7.23
Mixed Query	**8.80**	**8.74**	8.48	7.66	8.42	31.47
TCAM + FSO	8.76	8.69	**8.51**	**7.87**	**8.46**	**5.39**

other representative MLLMs: the open-source Qwen2.5-VL-72B-Instruct (Bai et al., 2025b) and the closed-source Gemini-3-Flash (Google DeepMind, 2025). As shown in Tab. E, CATP still consistently achieves the best performance across both evaluators and distinct token retention ratios, despite noticeable differences in their scoring distributions. This further suggests that CATP more reliably preserves generation quality rather than benefiting from a particular evaluator.

Additionally, it is noteworthy that, under the more capable evaluator, performance with larger numbers of reference images remains far from the maximum score (*i.e.* 10.0). While context compression provides partial recovery, we consider that further gains in such challenging scenarios should primarily stem from stronger native multi-reference generation capabilities of the foundation model, motivating continued research along this direction.

Table E: Comparison of multi-image synthesis performance scores by various methods for Bagel on MacroBench, evaluated by Qwen2.5-VL-72B-Instruct and Gemini-3-Flash. The best results are highlighted in **bold**.

Method	Qwen2.5-VL-72B-Instruct					Gemini-3-Flash				
	1–3	4–5	6–7	8–10	Avg.↑	1–3	4–5	6–7	8–10	Avg.↑
Full Model	8.43	7.65	6.45	5.42	6.99	9.01	8.48	6.93	4.45	7.21
retain 25% tokens										
Text-Aligned	8.07	7.52	6.26	5.34	6.80	8.05	8.34	6.76	4.82	6.99
VisionZip	7.81	7.36	5.88	5.02	6.52	7.43	7.71	6.36	3.72	6.31
SDTM	7.68	6.75	5.28	3.88	5.90	7.84	7.69	6.42	4.59	6.63
ToPi	8.10	7.45	**6.31**	5.18	6.76	8.47	8.33	6.80	4.66	7.06
CATP (Ours)	**8.44**	**7.84**	6.26	**5.62**	**7.04**	**8.87**	**8.70**	**7.54**	**5.25**	**7.59**
retain 10% tokens										
Text-Aligned	7.34	6.46	5.75	4.63	6.05	6.13	6.45	5.41	3.89	5.47
VisionZip	5.57	5.05	4.57	4.18	4.84	4.43	4.07	3.17	2.11	3.45
SDTM	6.71	5.40	4.08	3.05	4.81	6.74	6.37	5.42	3.79	5.58
ToPi	7.13	6.39	5.05	4.64	5.80	6.94	6.34	4.71	3.20	5.30
CATP (Ours)	**8.08**	**7.55**	**6.03**	**5.15**	**6.70**	**8.10**	**8.20**	**7.01**	**4.99**	**7.07**

A.4 Statistical Stability

We further assess the statistical stability of methods on Bagel at a 25% token retention ratio. Specifically, all methods are evaluated on MacroBench over five random seeds for image generation, with the mean and standard deviation reported. As shown in Tab. F, CATP consistently maintains superior performance, improving the Avg. over the second-best method by 0.306 points while exhibiting the lowest Avg. standard deviation of 0.02. These results further demonstrate both the effectiveness and stability of our method.

A.5 Compatibility with Orthogonal Acceleration Methods

CATP accelerates inference from the perspective of context compression, reducing attention computation by removing redundant conditional tokens. As discussed in Sec. 2, several methods optimize efficiency along orthogonal dimensions, motivating an exploration of their joint effect with CATP. Specifically, we combine CATP at a 25% token retention ratio with two representative approaches,

Table F: Comparison of MacroBench scores (mean ± std) by various methods on Bagel at a 25% token retention ratio. The best results are highlighted in **bold**.

Method	1–3	4–5	6–7	8–10	Avg.↑
Full Model	8.810 ± 0.045	8.412 ± 0.083	8.048 ± 0.078	7.256 ± 0.099	8.132 ± 0.047
Text-Aligned	8.210 ± 0.041	8.164 ± 0.060	8.064 ± 0.048	7.406 ± 0.064	7.960 ± 0.027
VisionZip	8.128 ± 0.065	8.010 ± 0.091	7.966 ± 0.168	7.012 ± 0.153	7.780 ± 0.065
SDTM	8.110 ± 0.222	8.048 ± 0.120	7.876 ± 0.157	7.252 ± 0.165	7.822 ± 0.066
ToPi	8.570 ± 0.056	8.278 ± 0.073	8.240 ± 0.087	7.532 ± 0.113	8.152 ± 0.032
CATP (Ours)	**8.814 ± 0.073**	**8.640 ± 0.104**	**8.540 ± 0.039**	**7.838 ± 0.058**	**8.458 ± 0.020**

SDTM (Fang et al., 2025) and TaylorSeer (Liu et al., 2025), both adopting their official default settings. SDTM shortens the denoising sequence, whereas TaylorSeer performs block-wise computation skipping across timesteps, and we therefore report the end-to-end speedup ratio of the entire denoising process relative to the full model, with 10 input reference images.

As shown in Tab. G, both methods alone already improve average generation quality over the full model. Here, SDTM fails to provide a meaningful acceleration, as the fixed context in UMM paradigm renders computation over its targeted denoising sequence non-dominant, leaving the overhead of frequent selection operation insufficiently amortized. Furthermore, when combined with these methods, CATP continues to reduce computational cost, while even improving generation performance by 0.19 and 0.26 points, respectively, primarily due to the elimination of interfering contextual redundancy. We also include representative generation results in Fig. B, where both methods produce erroneous outputs, yet their combination with CATP corrects these failures. These results demonstrate the favorable adaptability of CATP to prevalent techniques, and suggest that appropriate compression may itself provide a promising direction for improving downstream generation quality.

Table G: MacroBench scores and overall denoising speedup of SDTM and TaylorSeer, individually and in combination with CATP, on Bagel. The best results are highlighted in **bold**.

Method	1–3	4–5	6–7	8–10	Avg.↑	Speedup↑
Full Model	8.89	8.51	7.97	7.20	8.14	1.00×
SDTM	**8.95**	8.79	8.41	7.65	8.45	1.04×
SDTM + CATP (Ours)	8.91	**8.98**	**8.63**	**8.06**	**8.64**	**1.41**×
TaylorSeer	8.76	8.48	8.21	7.70	8.29	2.39×
TaylorSeer + CATP (Ours)	**8.78**	**8.95**	**8.65**	**7.81**	**8.55**	**3.06**×

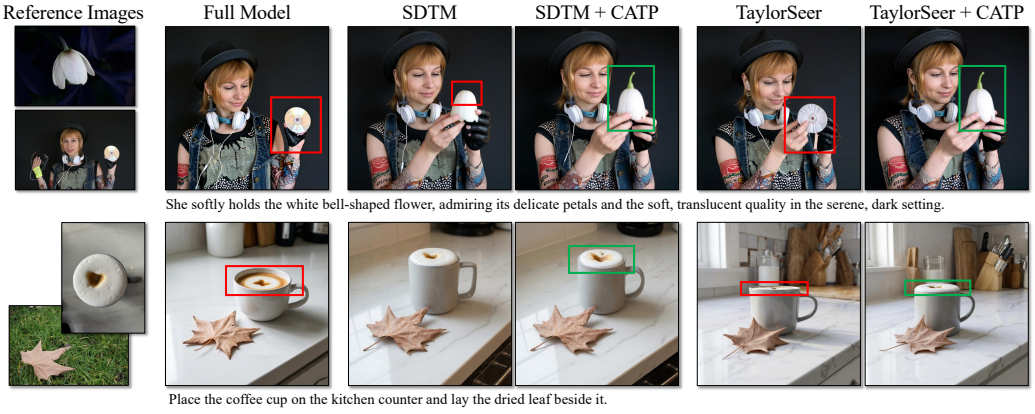


Figure B: Qualitative generation results by SDTM and TaylorSeer, individually and combined with CATP, on Bagel. The failure/well-preserved regions are marked with red/green boxes.

B Detailed Derivations

In this section, we present detailed derivations for TCAM and FSO. Sec. B.1 first formulates the pooling-induced attention discrepancy, then derives its second-order Taylor expansion with combined Hessian and query variance, yielding the calibrated importance score. Sec. B.2 reformulates this calibration into a simplified form, follow by a clarification of the theoretical basis of exact tiled online Softmax and the analogous online accumulation of the required intermediate statistics.

B.1 Derivation of the Taylor Calibration

Pooling-Induced Bias. We begin with the exact attention-based metric s_j computed over the complete query sequence $\boldsymbol{Q}$, corresponding to Eq. (3) in the main text:

$$s_j = \frac{1}{M}\sum\nolimits_{\boldsymbol{q}\in\boldsymbol{Q}} \mathrm{Attn}(\boldsymbol{q})_j, \quad \mathrm{Attn}(\boldsymbol{q}) = \mathrm{Softmax}(\frac{\boldsymbol{q}\boldsymbol{K}^\top}{\sqrt{d}}) \in \mathbb{R}^N. \tag{B.1}$$

Afterward, partitioning $\boldsymbol{Q}$ into $G = M/w^2$ non-overlapping $w \times w$ windows and applying average pooling yields, for the g-th window,

$$\bar{\boldsymbol{q}}_g = \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2} \boldsymbol{q}_{g,i}, \tag{B.2}$$

which further produces the coarsened window-level score:

$$p_{g,j} = \mathrm{Attn}(\bar{\boldsymbol{q}}_g)_j. \tag{B.3}$$

Accordingly, grouping the queries by window, the original dense attention metric of Eq. (B.1) can be equivalently written as:

$$s_j = \frac{1}{G}\sum\nolimits_{g=1}^{G} \hat{p}_{g,j}, \quad \hat{p}_{g,j} = \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2} \mathrm{Attn}(\boldsymbol{q}_{g,i})_j, \tag{B.4}$$

whereas the metric with pooled queries becomes:

$$\bar{s}_j = \frac{1}{G}\sum\nolimits_{g=1}^{G} \mathrm{Attn}(\bar{\boldsymbol{q}}_g)_j = \frac{1}{G}\sum\nolimits_{g=1}^{G} p_{g,j}. \tag{B.5}$$

Although $\bar{s}_j$ can be computed more efficiently, the difference from the exact s_j goes beyond a simple reordering of window-wise averaging and Softmax. Due to the nonlinearity of Softmax, each window incurs an additional non-negligible bias:

$$\Delta_{g,j} = \hat{p}_{g,j} - p_{g,j} = \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2} \mathrm{Attn}(\boldsymbol{q}_{g,i})_j - p_{g,j}, \tag{B.6}$$

which is identical to Eq. (8) of Sec. 3.3, and aggregating these window-level discrepancies gives:

$$s_j = \bar{s}_j + \frac{1}{G}\sum\nolimits_{g=1}^{G} \Delta_{g,j}. \tag{B.7}$$

Therefore, recovering the accurate importance score reduces to efficiently estimating the window-level correction $\Delta_{g,j}$, which motivates the following derivation.

Taylor Expansion for Discrepancy Approximation. For the g-th window, denote the deviation of each query $\boldsymbol{q}_{g,i}$ from the average-pooled representation $\bar{\boldsymbol{q}}_g$ as:

$$\boldsymbol{\epsilon}_{g,i} = \boldsymbol{q}_{g,i} - \bar{\boldsymbol{q}}_g, \tag{B.8}$$

Given the definition in Eq. (B.2), these deviations sum to zero within the corresponding window:

$$\sum\nolimits_{i=1}^{w^2} \boldsymbol{\epsilon}_{g,i} = \sum\nolimits_{i=1}^{w^2} \boldsymbol{q}_{g,i} - w^2\bar{\boldsymbol{q}}_g = \boldsymbol{0}. \tag{B.9}$$

Due to the spatial locality of visual representations, queries within each window typically lie close to their pooled center. This motivates a local Taylor expansion of $\mathrm{Attn}(\boldsymbol{q}_{g,i})_j$ around $\bar{\boldsymbol{q}}_g$. Using Eq. (B.3), the expansion simplifies to:

$$\begin{aligned}\mathrm{Attn}(\boldsymbol{q}_{g,i})_j = \mathrm{Attn}(\bar{\boldsymbol{q}}_g + \boldsymbol{\epsilon}_{g,i})_j &= p_{g,j} + \nabla p_{g,j}^\top \boldsymbol{\epsilon}_{g,i} + \frac{1}{2}\boldsymbol{\epsilon}_{g,i}^\top \boldsymbol{H}_{g,j}\boldsymbol{\epsilon}_{g,i} + \boldsymbol{R}^{(3)} \\ &\approx p_{g,j} + \nabla p_{g,j}^\top \boldsymbol{\epsilon}_{g,i} + \frac{1}{2}\boldsymbol{\epsilon}_{g,i}^\top \boldsymbol{H}_{g,j}\boldsymbol{\epsilon}_{g,i},\end{aligned} \tag{B.10}$$

where $\nabla p_{g,j} = \nabla_{\boldsymbol{q}} \operatorname{Attn}(\boldsymbol{q})_j|_{\boldsymbol{q}=\bar{\boldsymbol{q}}_g}$ and $\boldsymbol{H}_{g,j} = \nabla^2_{\boldsymbol{q}} \operatorname{Attn}(\boldsymbol{q})_j|_{\boldsymbol{q}=\bar{\boldsymbol{q}}_g}$ denote the first-order derivative and the second-order Hessian matrix, respectively. Terms beyond the second order, represented by $\boldsymbol{R}^{(3)}$, are omitted due to their negligible magnitude.

Furthermore, subtracting $p_{g,j}$ from both sides of Eq. (B.10) and averaging over all queries within the window yields:

$$\begin{aligned}\Delta_{g,j} &= \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2}(\operatorname{Attn}(\boldsymbol{q}_{g,i})_j - p_{g,j}) \\ &\approx \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2}(\nabla p_{g,j}^\top \boldsymbol{\epsilon}_{g,i}) + \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2}(\frac{1}{2}\boldsymbol{\epsilon}_{g,i}^\top \boldsymbol{H}_{g,j}\boldsymbol{\epsilon}_{g,i}) \\ &= \nabla p_{g,j}^\top \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2}\boldsymbol{\epsilon}_{g,i} + \frac{1}{2w^2}\sum\nolimits_{i=1}^{w^2}(\boldsymbol{\epsilon}_{g,i}^\top \boldsymbol{H}_{g,j}\boldsymbol{\epsilon}_{g,i}).\end{aligned} \tag{B.11}$$

Given Eq. (B.9), the first-order term vanishes exactly. Meanwhile, by defining the population covariance of the within-window query deviations as:

$$\boldsymbol{\Sigma}_g = \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2}\boldsymbol{\epsilon}_{g,i}\boldsymbol{\epsilon}_{g,i}^\top, \tag{B.12}$$

and applying the identity $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} = \operatorname{Tr}(\boldsymbol{A}\boldsymbol{x}\boldsymbol{x}^\top)$, where $\operatorname{Tr}(\cdot)$ is the matrix trace, the second-order term can be rewritten as $\frac{1}{2}\operatorname{Tr}(\boldsymbol{H}_{g,j}\boldsymbol{\Sigma}_g)$. Therefore, Eq. (B.11) yields

$$\Delta_{g,j} \approx \frac{1}{2}\operatorname{Tr}(\boldsymbol{H}_{g,j}\boldsymbol{\Sigma}_g). \tag{B.13}$$

Hessian of the Softmax Probability. For notational convenience, we denote the attention logit associated with $\boldsymbol{k}_j$ as:

$$z_j(\boldsymbol{q}) = \frac{\boldsymbol{q}^\top \boldsymbol{k}_j}{\sqrt{d}}, \tag{B.14}$$

such that the Softmax-normalized attention probability in Eq. (B.1) becomes:

$$p_j \equiv \operatorname{Attn}(\boldsymbol{q})_j = \frac{\exp(z_j(\boldsymbol{q}))}{\sum_{n=1}^N \exp(z_n(\boldsymbol{q}))}. \tag{B.15}$$

Here for brevity, we temporarily omit the window index g and evaluate all quantities at a generic query $\boldsymbol{q}$. The derivative of Softmax *w.r.t* its logits is:

$$\frac{\partial p_j}{\partial z_n} = p_j(\mathbb{I}(j=n) - p_n), \tag{B.16}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Applying the chain rule to Eq. (B.16) with Eq. (B.14) gives:

$$\begin{aligned}\nabla_{\boldsymbol{q}} p_j &= \sum\nolimits_{n=1}^N \frac{\partial p_j}{\partial z_n}\nabla_{\boldsymbol{q}} z_n = \sum\nolimits_{n=1}^N \frac{\boldsymbol{k}_n}{\sqrt{d}} p_j(\mathbb{I}(j=n) - p_n) \\ &= \frac{p_j}{\sqrt{d}}(\boldsymbol{k}_j - \sum\nolimits_{n=1}^N p_n \boldsymbol{k}_n) \\ &\equiv \frac{p_j}{\sqrt{d}}\boldsymbol{c}_j,\end{aligned} \tag{B.17}$$

where we define the attention-weighted mean key $\bar{\boldsymbol{k}}$ and the centered key vectors $\boldsymbol{c}_j$ as:

$$\bar{\boldsymbol{k}} = \sum\nolimits_{n=1}^N p_n \boldsymbol{k}_n, \quad \boldsymbol{c}_j = \boldsymbol{k}_j - \bar{\boldsymbol{k}}. \tag{B.18}$$

Adopting Eq. (B.17) and Eq. (B.18), we further prepare the Jacobian of $\bar{\boldsymbol{k}}$ *w.r.t* $\boldsymbol{q}$ as:

$$\begin{aligned}\nabla_{\boldsymbol{q}} \bar{\boldsymbol{k}} &= \sum\nolimits_{n=1}^N \boldsymbol{k}_n(\nabla_{\boldsymbol{q}} p_n)^\top \\ &= \frac{1}{\sqrt{d}}\sum\nolimits_{n=1}^N p_n \boldsymbol{k}_n(\boldsymbol{k}_n - \bar{\boldsymbol{k}})^\top \\ &= \frac{1}{\sqrt{d}}(\sum\nolimits_{n=1}^N p_n \boldsymbol{k}_n \boldsymbol{k}_n^\top - \bar{\boldsymbol{k}}\bar{\boldsymbol{k}}^\top),\end{aligned} \tag{B.19}$$

Differentiating Eq. (B.17) to the second-order gradient and noting that $\nabla_{\boldsymbol{q}}\boldsymbol{c}_j = -\nabla_{\boldsymbol{q}}\bar{\boldsymbol{k}}$ now gives:

$$\begin{aligned}\nabla^2_{\boldsymbol{q}} p_j &= \frac{1}{\sqrt{d}}\nabla_{\boldsymbol{q}}(p_j\boldsymbol{c}_j) = \frac{1}{\sqrt{d}}(\boldsymbol{c}_j(\nabla_{\boldsymbol{q}}p_j)^\top + p_j\nabla_{\boldsymbol{q}}\boldsymbol{c}_j)\\ &= \frac{1}{\sqrt{d}}(\boldsymbol{c}_j(\nabla_{\boldsymbol{q}}p_j)^\top - p_j\nabla_{\boldsymbol{q}}\bar{\boldsymbol{k}})\\ &= \frac{p_j}{d}(\boldsymbol{c}_j\boldsymbol{c}_j^\top - (\sum\nolimits_{n=1}^{N} p_n\boldsymbol{k}_n\boldsymbol{k}_n^\top - \bar{\boldsymbol{k}}\bar{\boldsymbol{k}}^\top))\\ &= \frac{p_j}{d}[(\boldsymbol{k}_j-\bar{\boldsymbol{k}})(\boldsymbol{k}_j-\bar{\boldsymbol{k}})^\top - \sum\nolimits_{n=1}^{N} p_n(\boldsymbol{k}_n-\bar{\boldsymbol{k}})(\boldsymbol{k}_n-\bar{\boldsymbol{k}})^\top].\end{aligned} \tag{B.20}$$

Restoring the window index and evaluating at $\bar{\boldsymbol{q}}_g$ gives $p_j \to p_{g,j}$ and $\bar{\boldsymbol{k}} \to \bar{\boldsymbol{k}}_g = \sum_{n=1}^{N} p_{g,n}\boldsymbol{k}_n$. This yields the following closed-form Hessian, which is identical to Eq. (11) in Sec. 3.3.

$$\boldsymbol{H}_{g,j} = \frac{p_{g,j}}{d}[(\boldsymbol{k}_j-\bar{\boldsymbol{k}}_g)(\boldsymbol{k}_j-\bar{\boldsymbol{k}}_g)^\top - \sum\nolimits_{n=1}^{N} p_{g,n}(\boldsymbol{k}_n-\bar{\boldsymbol{k}}_g)(\boldsymbol{k}_n-\bar{\boldsymbol{k}}_g)^\top]. \tag{B.21}$$

Contracting Hessian with Query Covariance. We first denote the diagonal entries of $\boldsymbol{\Sigma}_g$ by $\sigma^2_{g,c}$, where $c = 1, \ldots, d$, which also corresponds to the channel-wise variances as:

$$\sigma^2_{g,c} = \frac{1}{w^2}\sum\nolimits_{i=1}^{w^2} \boldsymbol{\epsilon}^2_{g,i,c}. \tag{B.22}$$

Since cross-channel correlations among the query deviations are typically weak, the off-diagonal covariance terms can be neglected. To further reduce sensitivity to channel-wise variance fluctuations, we approximate the covariance with an isotropic form:

$$\boldsymbol{\Sigma}_g \approx \sigma_g^2\boldsymbol{I}, \quad \sigma_g^2 = \frac{1}{d}\sum\nolimits_{c=1}^{d}\sigma^2_{g,c}. \tag{B.23}$$

Substituting Eq. (B.23) and Eq. (B.20) into Eq. (B.13), and adopting the identity $\mathrm{Tr}(\boldsymbol{x}\boldsymbol{x}^\top) = \|\boldsymbol{x}\|_2^2$, the approximated discrepancy gives:

$$\begin{aligned}\Delta_{g,j} &\approx \frac{\sigma_g^2}{2}\,\mathrm{Tr}(\boldsymbol{H}_{g,j})\\ &= \frac{p_{g,j}\sigma_g^2}{2d}\,\mathrm{Tr}[(\boldsymbol{k}_j-\bar{\boldsymbol{k}}_g)(\boldsymbol{k}_j-\bar{\boldsymbol{k}}_g)^\top - \sum\nolimits_{n=1}^{N} p_{g,n}(\boldsymbol{k}_n-\bar{\boldsymbol{k}}_g)(\boldsymbol{k}_n-\bar{\boldsymbol{k}}_g)^\top]\\ &= \frac{p_{g,j}}{2d}(\sigma_g^2\|\boldsymbol{k}_j-\bar{\boldsymbol{k}}_g\|_2^2 - \sum\nolimits_{n=1}^{N} p_{g,n}\sigma_g^2\|\boldsymbol{k}_n-\bar{\boldsymbol{k}}_g\|_2^2)\\ &\equiv \frac{p_{g,j}}{2d}(r_{g,j}-\bar{r}_g),\end{aligned} \tag{B.24}$$

where we define:

$$r_{g,j} = \sigma_g^2\|\boldsymbol{k}_j-\bar{\boldsymbol{k}}_g\|_2^2, \quad \bar{r}_g = \sum\nolimits_{n=1}^{N} p_{g,n}r_{g,n}. \tag{B.25}$$

Calibrated Importance Score. Revisiting the global calibration target in Eq. (B.7), the approximated window-wise discrepancy in Eq. (B.24) yields the calibrated window-level score:

$$\tilde{p}_{g,j} = p_{g,j} + \lambda\frac{p_{g,j}}{2d}(r_{g,j}-\bar{r}_g), \tag{B.26}$$

where λ controls the correction strength. Finally, the overall calibrated importance score formulates:

$$\tilde{s}_j = \frac{1}{G}\sum\nolimits_{g=1}^{G}\tilde{p}_{g,j}. \tag{B.27}$$

This corresponds to Eq. (13) in Sec. 3.3, augmenting the efficient pooled attention with a modest additional second-order correction to improve token discriminability for more reliable pruning.

B.2 Derivation of the Fast Scoring Operator

Simplified Window-Wise Correction. We further simplify Eq. (13), equivalently given by Eq. (B.26), in preparation for practical implementation.

First, the centered squared distance $r_{g,j}$ can be expanded as:

$$r_{g,j} = \sigma_g^2 \|\boldsymbol{k}_j - \bar{\boldsymbol{k}}_g\|_2^2 = \sigma_g^2 (\|\boldsymbol{k}_j\|_2^2 - 2\boldsymbol{k}_j^\top \bar{\boldsymbol{k}}_g + \|\bar{\boldsymbol{k}}_g\|_2^2). \tag{B.28}$$

Then, given that $\sum_{n=1}^N p_{g,n} = 1$ and $\sum_{n=1}^N p_{g,n}\boldsymbol{k}_n = \bar{\boldsymbol{k}}_g$, the weighted mean distance $\bar{r}_g$ is simplified as:

$$\begin{aligned} \bar{r}_g &= \sum\nolimits_{n=1}^N p_{g,n} r_{g,n} \\ &= \sigma_g^2 (\sum\nolimits_{n=1}^N p_{g,n} \|\boldsymbol{k}_n\|_2^2 - 2(\sum\nolimits_{n=1}^N p_{g,n}\boldsymbol{k}_n)^\top \bar{\boldsymbol{k}}_g + \|\bar{\boldsymbol{k}}_g\|_2^2 \sum\nolimits_{n=1}^N p_{g,n}) \\ &= \sigma_g^2 (\sum\nolimits_{n=1}^N p_{g,n} \|\boldsymbol{k}_n\|_2^2 - \|\bar{\boldsymbol{k}}_g\|_2^2). \end{aligned} \tag{B.29}$$

Subtracting Eq. (B.29) from Eq. (B.28) and substituting the result into Eq. (B.26) ultimately yields:

$$\tilde{p}_{g,j} = p_{g,j}[1 + \frac{\lambda\sigma_g^2}{2d}(\|\boldsymbol{k}_j\|_2^2 - 2\boldsymbol{k}_j^\top \bar{\boldsymbol{k}}_g + 2\|\bar{\boldsymbol{k}}_g\|_2^2 - \sum\nolimits_{n=1}^N p_{g,n}\|\boldsymbol{k}_n\|_2^2)]. \tag{B.30}$$

This is algebraically equivalent to the TCAM formulation. It eliminates the explicit computation of $\bar{r}_g$, requiring only the window-level statistics $\bar{\boldsymbol{k}}_g$ and $u_g = \sum_{n=1}^N p_{g,n}\|\boldsymbol{k}_n\|_2^2$, both of which can be accumulated online as in Algo. 1.

Exactness of Online Softmax. To establish the basis for the tiled implementation, we first review the exactness of online Softmax. Without loss of generality, we consider a single query $\boldsymbol{q}$ within window g and let $\boldsymbol{z} = \{z_j\}_{j=1}^N$ denote its attention logits.

Softmax is invariant to a uniform shift of its logits, *i.e.* for any scalar c,

$$\mathrm{Softmax}(\boldsymbol{z})_j = \frac{\exp(z_j - c)}{\sum_{n=1}^N \exp(z_n - c)}. \tag{B.31}$$

In particular, choosing $c = \max_j z_j$ improves numerical stability without changing the resulting probabilities. Online Softmax exploits this property by dynamically updating the reference maximum as key blocks are processed.

Specifically, consistent with the notations in Algo. 1, let the key sequence $\boldsymbol{K}$ be partitioned into T_K blocks, each of size $B_K \times d$. After processing the first b blocks, define the running maximum m_b and the normalization factor ℓ_b as:

$$m_b = \max_{j=1,\ldots,bB_K} z_j, \quad \ell_b = \sum\nolimits_{j=1}^{bB_K} \exp(z_j - m_b). \tag{B.32}$$

When processing a new block, the running maximum changes from m_{b-1} to m_b, and all previously accumulated terms can then be exactly rescaled to the new reference maximum using:

$$\alpha = \exp(m_{b-1} - m_b), \tag{B.33}$$

yielding:

$$\ell_b = \alpha\ell_{b-1} + \sum\nolimits_{j=(b-1)B_K+1}^{bB_K} \exp(z_j - m_b). \tag{B.34}$$

This recurrence preserves the exact normalization throughout block-wise accumulation. After the first pass till the final block, the resulting m_{T_K} and ℓ_{T_K} recover each attention probability as:

$$p_j = \frac{\exp(z_j - m_{T_K})}{\ell_{T_K}} = \mathrm{Softmax}(\boldsymbol{z})_j. \tag{B.35}$$

Thus, tiled online computation is algebraically equivalent to standard Softmax, differing only in the accumulation procedure.

Online Accumulation of TCAM Statistics. Given the shift invariance of Eq. (B.31), any statistic aggregated with normalized Softmax weights is likewise invariant to a uniform logit shift, *e.g.* for arbitrary $\{\boldsymbol{a}_j\}_{j=1}^N$ and scalar c,

$$\sum\nolimits_{j=1}^{N} p_j \boldsymbol{a}_j = \frac{\sum_{j=1}^{N} \exp(z_j - c)\boldsymbol{a}_j}{\sum_{n=1}^{N} \exp(z_n - c)}. \tag{B.36}$$

Building on this, the two attention-weighted statistics required by TCAM, *i.e.* $\bar{\boldsymbol{k}}_g = \sum_{j=1}^{N} p_{g,j}\boldsymbol{k}_j$ and $\sum_{n=1}^{N} p_{g,n}\|\boldsymbol{k}_n\|_2^2$, can be accumulated jointly within the first online pass. Concretely, after processing the first b key blocks, we maintain the corresponding unnormalized running statistics:

$$\boldsymbol{v}_b = \sum\nolimits_{j=1}^{bB_K} \exp(z_j - m_b)\boldsymbol{k}_j, \quad u_b = \sum\nolimits_{j=1}^{bB_K} \exp(z_j - m_b)\|\boldsymbol{k}_j\|_2^2. \tag{B.37}$$

Both follow the same rescaling rule when the running maximum changes, *i.e.*,

$$\begin{aligned} \boldsymbol{v}_b &= \alpha_b \boldsymbol{v}_{b-1} + \sum\nolimits_{j=(b-1)B_K+1}^{bB_K} \exp(z_j - m_b)\boldsymbol{k}_j, \\ u_b &= \alpha_b u_{b-1} + \sum\nolimits_{j=(b-1)B_K+1}^{bB_K} \exp(z_j - m_b)\|\boldsymbol{k}_j\|_2^2. \end{aligned} \tag{B.38}$$

After the final block, normalization by ℓ_{T_K} exactly recovers the required statistics as:

$$\bar{\boldsymbol{k}}_g = \frac{\boldsymbol{v}_{T_K}}{\ell_{T_K}} = \sum\nolimits_{j=1}^{N} p_{g,j}\boldsymbol{k}_j, \quad \sum\nolimits_{n=1}^{N} p_{g,n}\|\boldsymbol{k}_n\|_2^2 = \frac{u_{T_K}}{\ell_{T_K}}. \tag{B.39}$$

With these finalized statistics, a second traversal over the key blocks recomputes the exact probabilities and directly applies the calibration, yielding the final TCAM scores. Consequently, neither the full attention matrix nor the centered-distance matrix needs to be materialized.