REGATE: Learning Faster and Better with Fewer Tokens in MLLMs

Chaoyu Li, Yogesh Kulkarni, Pooyan Fazli
Arizona State University, Arizona, USA
{chaoyuli, ykulka10, pooyan}@asu.edu
https://people-robots.github.io/regate

Abstract

The computational cost of training multimodal large language models (MLLMs) grows rapidly with the number of processed tokens. Existing efficiency methods mainly target inference via token reduction or merging, offering limited benefits during training. We introduce REGATE (**R**eference-**G**uided **A**daptive **T**oken **E**lision), an adaptive token pruning method for accelerating MLLM training. REGATE adopts a teacher-student framework, in which a frozen teacher LLM provides per-token guidance losses that are fused with an exponential moving average of the student's difficulty estimates. This adaptive scoring mechanism dynamically selects informative tokens while skipping redundant ones in the forward pass, substantially reducing computation without altering the model architecture. Across three representative MLLMs, REGATE matches the peak accuracy of standard training on MVBench up to **2**× **faster**, using only **38%** of the tokens. With extended training, it even surpasses the baseline across multiple multimodal benchmarks, cutting total token usage by over **41%**.

1 Introduction

Multimodal large language models (MLLMs) face significant challenges due to the high computational cost of training (Jin et al., 2024). A key bottleneck is the self-attention mechanism, whose complexity grows quadratically with input sequence length (Vaswani et al., 2017). This issue is especially severe in video tasks (Kulkarni and Fazli, 2024, 2025), where frames are tokenized into extremely long sequences. Consequently, training MLLMs on large-scale datasets requires substantial computing resources, limiting accessibility and slowing progress. Improving training efficiency is therefore essential for enabling MLLMs to scale to longer contexts, support richer modalities, and learn from larger datasets that would otherwise be infeasible to process.

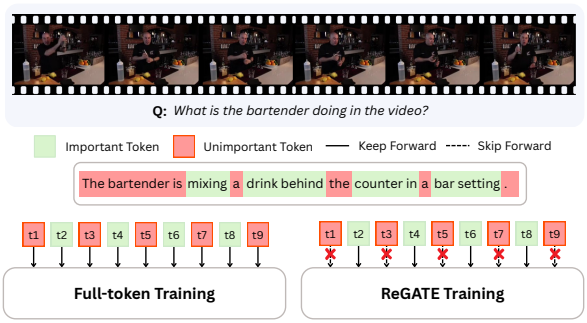


Figure 1: In training MLLMs, **REGATE** identifies important textual tokens (light green) and selectively propagates them, while skipping unimportant ones (red).

Several strategies have been proposed to speed up inference in MLLMs, including token pruning (Ye et al., 2025; Arif et al., 2025; Alvar et al., 2025; Lee et al., 2025), token merging (Chen et al., 2024; Dhouib et al., 2025), and token compression (Yang et al., 2025c). While these methods reduce FLOPs during inference, they rely on prescored or frozen tokens and cannot adapt token importance dynamically during training. As a result, reducing training costs remains a more complex and less explored problem. In unimodal text models, learnable token pruning methods such as RHO-1 (Lin et al., 2024c) have improved training efficiency, but these approaches have not been extended to large multimodal models. Early efforts to speed up visual processing, often focusing on standard vision transformers (Akbari et al., 2021) or early video-language models (Lei et al., 2021), rely on heuristics like random token dropping. In multimodal settings, however, determining which tokens are important often depends on subtle visual evidence, temporal cues, or interactions between multiple modalities. Ignoring these dependencies can lead to the removal of crucial information, resulting in inefficient training and reduced multimodal understanding. This gap highlights the need for a principled method to estimate token importance

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 46430–46446

during training, one that captures cross-modal dependencies without relying on heuristics.

To address this challenge, we introduce **REGATE** (**Re**ference-**G**uided **A**daptive **T**oken **E**lision), a framework designed to accelerate the training of MLLMs (Figure 1). REGATE adopts a teacher-student architecture, in which the student is the multimodal model being trained, and the teacher is a frozen, text-only version of the same LLM backbone. REGATE combines two complementary signals to identify and retain the most informative tokens during training dynamically. First, it checks whether a token requires visual grounding by seeing if the text-only teacher can predict it from the prompt alone. Second, it evaluates the student model's learning progress using an exponential moving average (EMA) of token-wise historical losses. By integrating these signals, REGATE allocates computation to the subset of tokens that are both critical for multimodal understanding and remain challenging for the model to learn. To validate the effectiveness and generality of REGATE, we evaluate it on three representative MLLMs: VideoChat2 (Li et al., 2024c), VideoLLaMA2 (Cheng et al., 2024), and InternVL3.5 (Wang et al., 2025), covering a wide range of architectures, training paradigms, and scales, and assess performance across diverse image and video benchmarks.

To summarize, our contributions are threefold:

- We introduce REGATE, an adaptive token pruning method for accelerating MLLM training. REGATE leverages a text-only reference teacher model and the student's historical token difficulty to dynamically identify and retain visually essential tokens, without introducing any additional trainable parameters.
- We show that the model-agnostic REGATE integrates seamlessly into existing MLLMs, requiring no architectural changes, making it easy to adopt.
- Extensive experiments on five image benchmarks and eight video benchmarks demonstrate REGATE's broad applicability and efficiency. On the challenging MVBench benchmark, REGATE reaches the baseline's peak accuracy up to **2× faster** while using only **38%** of the tokens on average.

2 Related Work

2.1 Token Compression for Fast Inference

Most existing work focuses on accelerating inference, not training. Inference-time sparsity methods show that many tokens can be removed or merged with minimal impact on accuracy (Shi et al., 2025; Jiang et al., 2025; Yang et al., 2025b; Hyun et al., 2025). In vision transformers, Dynamic Token Pruning (Tang et al., 2023) halts processing of easy tokens layer by layer, reducing FLOPs by 20–35% without degrading performance. For video LLMs, DyCoke (Tao et al., 2025) compresses spatial-temporal tokens during inference, achieving up to 2× speed-ups while keeping model weights frozen. Moving beyond pruning, Importance-Based Token Merging (Wu et al., 2025) merges similar tokens rather than dropping them, maintaining performance on long-video benchmarks while delivering 1.5× faster inference. However, all these methods operate after training is complete. During training, the full token is still processed in every forward and backward pass, leaving the computational cost of training mainly unaddressed.

2.2 Token Compression for Fast Training

Only a few studies have explored token compression during training, rather than just at inference. In text-only language models, RHO-1 (Lin et al., 2024c) ranks tokens with a reference model and backpropagates only through the most difficult subset, reducing pre-training tokens by 50% while improving accuracy. For MLLMs, LaVi (Yue et al., 2025) injects vision-conditioned deltas into layer norms to skip visual tokens but needs extra modulation layers. LLaVA-Meteor (li et al., 2025) introduces a flash-fusion module and a dual-expert scorer that prunes 75–95% visual tokens during instruction tuning but adds extra parameters and targets only visual tokens. In contrast, REGATE combines a *static* cross-modal reference loss from a text-only teacher with a *dynamic* EMA-based student signal. Together, these signals produce an adaptive, parameter-free sparsity mechanism that selectively gates textual tokens while preserving visual ones, without modifying the model architecture.

2.3 Teacher-Student Distillation for MLLMs

Most distillation approaches for MLLMs mainly focus on parameter compression (Li et al., 2023a; Cai et al., 2025; Udandarao et al., 2025). A systematic

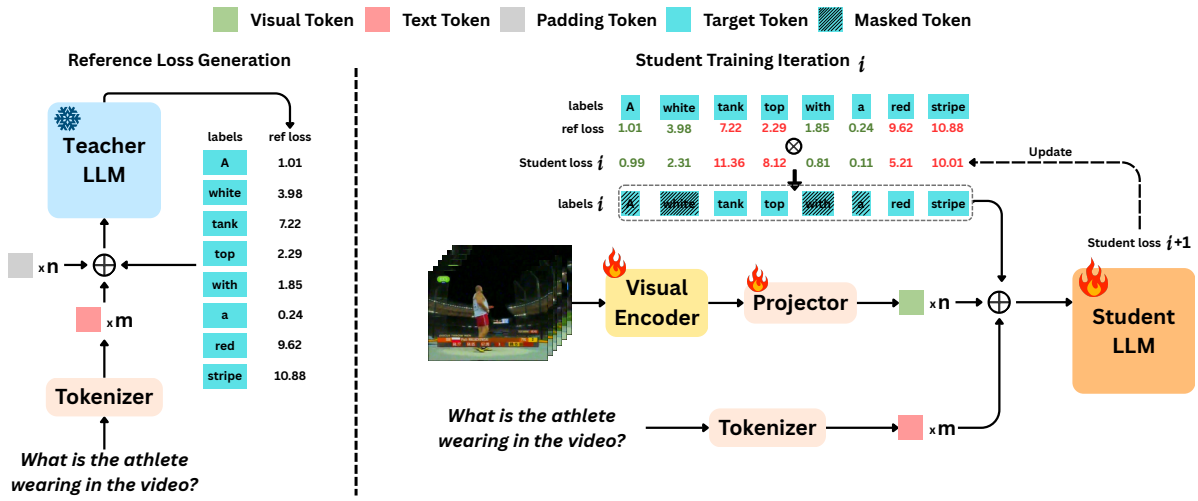


Figure 2: **Overview of REGATE.** The framework operates in two interconnected stages. **(1) Reference Loss Generation (Left):** A frozen, text-only teacher LLM processes the input text (with padding tokens) and computes a per-token reference loss (`ref_loss`), which measures how difficult each token is to predict from text alone. Higher loss values suggest the token likely requires visual grounding (e.g., "white", "red stripe"). **(2) Student Training (Right):** The `ref_loss` is combined with the student model's historical learning difficulty to produce a unified importance score. This score is used to create a binary mask that selects the most informative tokens. During training, the student LLM receives the full multimodal input but only performs computation (e.g., self-attention and feed-forward operations) on the selected tokens, while skipping the rest.

study (Xu et al., 2024) shows that jointly aligning tokens and logits helps a smaller student model inherit visual grounding from a larger teacher model. Similarly, methods like DIME-FM (Sun et al., 2023) show how cross-modal features can be transferred even from unpaired data. More recently, MaskedKD (Son et al., 2024) reduces teacher FLOPs by up to 50% by masking patch tokens based on student attention, but it still back-propagates through all student tokens. In contrast, REGATE leverages the teacher's per-token loss to decide which tokens the student should process during each forward and backward pass. This approach significantly reduces computation without altering parameters and enables model-agnostic sparsity for more efficient training.

3 ReGATE

We introduce REGATE, a method that speeds up the training of MLLMs by selectively allocating computational resources only to tokens that truly require visual information. The key insight is that not all tokens in a multimodal sequence depend equally on visual context: some can be accurately predicted from text alone, while others need cross-modal grounding. To capture this, REGATE uses a teacher-student framework. The student is the main MLLM being trained. The teacher is a reference model created by taking the student's LLM backbone, removing its visual components (the visual encoder and projector), and freezing its weights. This results in a pure text-only LLM that acts as a fixed expert to estimate the degree to which each token depends on visual input. Given a batch of input sequences containing both text and visual tokens, we generate a binary mask that determines which tokens should be actively computed and which can be skipped. This section explains how we calculate per-token difficulty scores using the frozen text-only teacher combined with the student's own training history, how we dynamically adjust the fraction of tokens retained during training, and how we apply the resulting mask within the transformer decoder.

3.1 Difficulty Score Formulation

Let $\mathbf{x}_b = (x_{b,1}, \ldots, x_{b,T})$ denote the token sequence in sample b, including both text tokens and special visual tokens (e.g., <image> or <video> tokens representing visual content). To compute the reference loss, we construct a modified sequence $\hat{\mathbf{x}}_b$ by replacing the actual visual tokens with placeholder tokens (typically the padding token <pad>), ensuring the sequence length remains identical to the original multimodal input fed to the MLLM's backbone LLM. Our reference model is a pure text-only LLM obtained by removing the visual encoder

and projector from the MLLM backbone, thus incapable of processing any visual content. By feeding the constructed placeholder sequence $\hat{\mathbf{x}}_b$ to the reference model in evaluation mode, we compute the per-token negative log-likelihood:

$$\ell_{b,i}^{\text{ref}} = -\log p_{\text{teacher}}\big(x_{b,i} \mid \hat{\mathbf{x}}_{b,<i}\big). \tag{1}$$

A low value of $\ell_{b,i}^{\text{ref}}$ indicates that the teacher can predict $x_{b,i}$ based on the textual context alone, whereas a high value signals that multimodal information is needed to predict the token. In parallel, we monitor how difficult each token has been for the student across training updates. For every training sample s and token position i, we maintain a running difficulty buffer $m_{s,i}$ updated as an exponential moving average (EMA) of the student's cross-entropy loss:

$$m_{s,i} \leftarrow \beta\, m_{s,i} + (1-\beta)\, \ell_{b,i}^{\text{stu}}, \quad \beta \in (0,1), \tag{2}$$

where $\ell_{b,i}^{\text{stu}}$ is the current cross-entropy loss of the student model at token position i, and β controls the smoothing of the EMA. A higher value of $m_{s,i}$ indicates that token i in sample s has consistently posed difficulties during training. We then combine the reference loss and the student's historical difficulty into a unified score for each token:

$$d_{b,i} = m_{s,i} + \lambda\, \ell_{b,i}^{\text{ref}}, \tag{3}$$

where λ balances these two signals. Tokens with a higher combined difficulty, $d_{b,i}$, either consistently challenge the student model or genuinely require visual context, and thus are prioritized during the training updates. Note that this combined difficulty evaluation is performed exclusively on output tokens (labels), as these tokens directly influence the training process through backpropagation.

3.2 Dual-cycle Sparsity Schedule

We employ a deterministic schedule to determine the fraction of tokens kept at each training step. Our schedule repeats every C steps. In the first F steps of each cycle, we keep all tokens (i.e., $p = 1$) to allow the model to stabilize. In the remaining $C - F$ steps, we retain only a fixed proportion p_{sparse} of the tokens. Formally, if t denotes the global training step, we have:

$$p(t) = \begin{cases} 1, & \text{if } t \bmod C < F, \\ p_{\text{sparse}}, & \text{otherwise.} \end{cases} \tag{4}$$

3.3 Dynamic Token Gating

For each sample b, we identify the indices of valid tokens excluding padding and special markers. Let $\mathcal{I}_b$ denote those indices and $N_b = |\mathcal{I}_b|$. We compute the combined difficulty $d_{b,i}$ for $i \in \mathcal{I}_b$ using Equation (3) and select the top $k_b = \max\big(1, \lfloor p(t) \cdot N_b \rfloor\big)$ tokens. The resulting binary mask $\boldsymbol{m}_b \in \{0,1\}^T$ is set to one for retained tokens and zero otherwise. We always retain all special visual tokens (e.g., those corresponding to a frame or image) regardless of their difficulty to preserve multimodal information. Because the difficulty buffer $m_{s,i}$ is updated after every epoch, the set of selected positions adapts throughout training: tokens that become easy for the student are gradually deprioritised, while persistently challenging tokens or those requiring visual grounding remain active. This dynamic gating enables the model to allocate its computational budget to the most informative parts of the sequence at each epoch, rather than committing to a fixed sparsity pattern. Finally, the per-sample binary masks are concatenated and padded to form a batch mask $\mathbf{M} \in \{0,1\}^{B \times T'}$ where T' is the expanded sequence length accounting for visual tokens.

3.4 Adaptive Decoder Sparsity

To exploit the binary mask during forward propagation, we modify the transformer decoder layer in the backbone LLM to support token-level sparse computation. Algorithm 1 outlines how the binary mask M governs computation within each decoder layer. Given the input hidden states, we first apply layer normalization, after which self-attention is computed only over tokens marked as active by M. In practice, sparse attention is implemented by directly passing M as the attention mask to flash attention routines, zeroing out the hidden states of pruned tokens, and restricting query, key, and value computations to active positions only. Specifically, queries are formed only for retained tokens, while keys and values are gathered from the same active subset, excluding pruned tokens entirely from attention computation. The resulting attention outputs are then incorporated via residual connections.

We apply an analogous strategy to the feed-forward network. Hidden states corresponding to active tokens are gathered, processed by the MLP, and scattered back to their original positions. Tokens not selected by M bypass the MLP and remain unchanged, with residual connections ensuring that

Algorithm 1 Sparse Decoder Layer Forward

Require: $\mathbf{H} \in \mathbb{R}^{B\times S\times D}$ ▷ hidden states
Require: $\mathbf{M} \in \{0,1\}^{B\times S}$ ▷ token mask
for $b = 1$ **to** B **do** ▷ B = batch size
  $\mathbf{x} \leftarrow \mathrm{LN}_{\text{in}}(\mathbf{H}[b])$
  $\mathbf{mask} \leftarrow \mathbf{M}[b]$ ▷ 1=keep, 0=skip
  $\mathbf{a} \leftarrow \mathrm{SelfAttn}(\mathbf{x}, \mathbf{mask})$
  $\mathbf{H}[b] \leftarrow \mathbf{H}[b] + \mathbf{a}$
  $\mathrm{active} \leftarrow \mathrm{nonzero}(\mathbf{mask})$
  $\mathbf{h} \leftarrow \mathrm{MLP}(\mathrm{LN}_{\text{post}}(\mathbf{H}[b])[\mathrm{active}])$
  $\mathbf{H}[b][\mathrm{active}] \leftarrow \mathbf{H}[b][\mathrm{active}] + \mathbf{h}$
end for
return H

their previous representations are propagated forward across layers. As a result, each decoder layer performs both self-attention and MLP only on the active subset, while inactive tokens are efficiently skipped without altering the model's functional behavior. This design introduces no additional parameters, integrates seamlessly with existing frameworks such as HuggingFace Transformers, and remains fully compatible with pre-trained weights.

During back propagation, gradients are computed only for parameters associated with the active tokens. Inactive tokens are treated as constants, and their activations and losses receive no gradient updates. Residual connections preserve continuity of gradient flow across layers, aligning the backward computation with the sparse forward path and maintaining training stability in practice.

4 Experiments

4.1 Implementation Details

To demonstrate the effectiveness and generality of REGATE, we apply it to three representative MLLMs, VideoChat2 (Li et al., 2024c), VideoLLaMA2 (Cheng et al., 2024), and InternVL3.5 (Wang et al., 2025), spanning diverse architectures, training paradigms, and scales. We exclude models such as Qwen2.5-VL (Bai et al., 2025) and VideoLLaMA3 (Zhang et al., 2025) due to the lack of publicly available pretrained weights. Training such models from scratch is impractical, as they rely on web-scale data and hundreds of GPUs. However, with sufficient resources and pretrained weights, REGATE can be seamlessly integrated into any MLLM training pipeline. For all experiments, the per-token reference losses from the text-only teacher are computed once over the entire fine-tuning dataset before student training begins and then cached and reused throughout training.

VideoLLaMA2. We first apply REGATE to VideoLLaMA2-7B (Cheng et al., 2024), whose Qwen2-7B (Yang et al., 2024) backbone is unfrozen during multimodal fine-tuning. REGATE is introduced at this stage, using a text-only reference teacher (the same backbone without the visual encoder or adapter) that computes per-token losses with visual tokens masked out. This configuration allows us to examine how REGATE interacts with full fine-tuning on a standard multimodal backbone.

VideoChat2. To evaluate REGATE under parameter-efficient fine-tuning (PEFT), we integrate it into the LoRA-based Stage 3 training of VideoChat2-7B (Li et al., 2024c), built on Mistral-7B (Jiang et al., 2023). Gradients for LoRA parameters are computed only from the high-importance tokens selected by REGATE, while the language backbone weights remain frozen. This setup enables us to assess whether REGATE can complement PEFT methods without architectural modifications.

InternVL3.5. Finally, we evaluate scalability of REGATE on InternVL3.5-14B (Wang et al., 2025), based on Qwen3-14B (Yang et al., 2025a). The reference teacher is derived from the same backbone by removing its visual encoder and projector, forming a text-only LLM. During fine-tuning, REGATE dynamically gates tokens to focus computation on the most informative positions, reducing activation memory and training FLOPs. This setup allows us to assess REGATE's scalability and stability at large model scales.

Datasets and sparsity schedule. We fine-tune VideoLLaMA2 with and without REGATE on the VideoChatGPT dataset (Maaz et al., 2024), which is a subset of VideoLLaMA2's official fine-tuning dataset containing approximately 300,000 instruction-response pairs. For VideoChat2, we similarly use a subset of its official fine-tuning data comprising around 2.6 million instruction pairs. For InternVL3.5, we adopt the same 2.6M-sample dataset used for VideoChat2, since the full InternVL3.5 training corpus is extremely large and not fully released. Training follows the dual-cycle sparsity schedule described in Section 3.2, with parameters set to $C = 128$, $F = 16$, and $p_{\text{sparse}} = 0.5$. To ensure stable training at the start, we prepend a global warm-up phase of 100 iterations, during which all tokens are retained.

Table 1: **Zero-shot evaluation results on image understanding benchmarks.** Previous best results are highlighted in **bold**, while REGATE's best results are underlined. I: SEED benchmark results are reported only for the image subset. For baseline models, scores are taken from their official publications where available.

Model	LLM	Tokens	ScienceQA	MME	VizWiz	POPE	SEED^I
Open-source Models							
InstructBLIP (Dai et al., 2023)	Vicuna-7B	–	60.5	254.3/1137.1	34.5	86.1	46.4
LLaVA-1.5 (Liu et al., 2024a)	Vicuna-7B	–	66.8	302.1/1506.2	50.0	85.9	66.1
Qwen-VL-Chat (Bai et al., 2023)	Qwen-7B	–	68.2	392.1/1467.8	38.9	74.9	58.2
LLaVA-1.6 (Liu et al., 2024a)	Vicuna-7B	–	70.1	–	57.6	86.5	70.2
VILA1.5 (Lin et al., 2024b)	Llama-2-13B	–	79.1	288.9/1429.3	**60.6**	84.2	62.8
LLaVA-Next (Liu et al., 2024b)	Mistral-7B	–	73.0	308.9/1512.3	–	87.3	72.4
LLaVA-OneVision (Li et al., 2024a)	Qwen2-7B	–	**95.4**	415.7/1577.8	53.0	87.4	75.4
Qwen2.5-VL (Bai et al., 2025)	Qwen2.5-7B	–	89.0	613.9/**1698.1**	–	85.9	77.0
Proprietary Models							
Claude3.7-Sonnet (Anthropic, 2025)	–	–	90.9	649.6/1189.7	–	82.4	74.3
Gemini-1.5-Flash (Gemini et al., 2024)	–	–	83.3	488.6/1589.3	–	**88.5**	75.0
Gemini-1.5-Pro (Gemini et al., 2024)	–	–	85.7	548.2/1562.4	–	88.2	76.0
GPT-4o (Hurst et al., 2024)	–	–	90.1	**719.3**/1609.4	–	85.0	76.4
GPT-4.1 (Hurst et al., 2024)	–	–	92.8	673.9/1663.6	–	86.4	**78.0**
Models w/wo REGATE							
VideoChat2	Mistral-7B	3.93B	40.8	314.6/1244.0	28.5	86.2	45.9
VideoChat2-REGATE	Mistral-7B	2.22B (↓ 43.51%)	46.6$_{+5.8}$	360.7/1287.8$_{+46.1/+43.8}$	32.5$_{+4.0}$	85.1$_{-1.1}$	47.2$_{+1.3}$
VideoLLaMA2	Qwen2-7B	83.82M	61.4	376.4/1474.0	46.8	86.7	70.4
VideoLLaMA2-REGATE	Qwen2-7B	49.27M (↓ 41.22%)	80.5$_{+19.1}$	391.1/1507.1$_{+14.7/+33.1}$	48.0$_{+1.2}$	87.5$_{+0.8}$	70.0$_{-0.3}$
InternVL3.5	Qwen3-14B	3.96B	93.3	681.6/1694.3	60.6	91.6	76.8
InternVL3.5-REGATE	Qwen3-14B	2.32B (↓ 41.41%)	94.4$_{+1.1}$	689.3/1698.8$_{+7.7/+4.5}$	61.5$_{+0.9}$	93.1$_{+1.5}$	76.6$_{-0.2}$

The main hyperparameters for REGATE include an exponential moving average (EMA) decay of $\beta = 0.9$ and a teacher loss weighting coefficient of $\lambda = 0.5$. All experiments on VideoLLaMA2 and VideoChat2 are run on 4 H100 GPUs, while experiments on InternVL3.5 use 16 H100 GPUs, all under mixed-precision training.

4.2 Benchmarks and Baselines

We evaluate REGATE across image, long-video, and short-video understanding benchmarks under the LMMs-Eval (Zhang et al., 2024b) framework. For **image understanding**, we include ScienceQA (Lu et al., 2022), MME (Fu et al., 2024), VizWiz (Gurari et al., 2018), POPE (Li et al., 2023b), and SEED (Li et al., 2024b), which together probe scientific reasoning, perception, answerability, hallucination, and broad multimodal comprehension. For **long-video understanding**, we evaluate on Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), MLVU (Zhou et al., 2025), and EgoSchema (Mangalam et al., 2023), all of which emphasize extended temporal reasoning, consistency, and grounding across minutes- to hour-long sequences. For **short-video understanding**, we adopt MVBench (Li et al., 2024c), Perception Test (Pătrăucean et al., 2023), Vinoground (Zhang et al., 2024a), and NExT-QA (Xiao et al., 2021), which target fine-grained event recognition, local temporal relations, counterfactual reasoning, and causal/temporal action reasoning in clips of tens of seconds.

We evaluate REGATE against a diverse set of state-of-the-art models, including high-performing open-source families (LLaVA, Qwen) and proprietary models (Google Gemini, OpenAI GPT, Anthropic Claude). This selection spans multiple LLM backbones and sizes, ensuring robust comparisons across different tasks (Tables 1, 2, 3).

4.3 Results

Learning better: ReGATE's accuracy gains across image and video benchmarks. The comprehensive results presented in Tables 1, 2, and 3 show how VideoLLaMA2, VideoChat2, and InternVL3.5 perform, with and without REGATE, across a range of image, short video, and long video understanding benchmarks. REGATE improves performance consistently by focusing computation on the most informative tokens. Figure 4 visualizes the learned attention maps, showing that ReGATE concentrates attention on visually and semantically important tokens compared to standard fine-tuning. For example, VideoLLaMA2-REGATE outperforms the baseline VideoLLaMA2 on most tasks while using 41.22% fewer tokens. Similarly, VideoChat2-REGATE achieves better results than the baseline VideoChat2 while using 43.51% fewer tokens. At the larger 14B scale, InternVL3.5-REGATE also shows gains across both image and

Table 2: **Zero-shot evaluation results on long video understanding benchmarks.** Previous best results are highlighted in **bold**, while REGATE's best results are underlined. † Results on Video-MME are reported without subtitles. For baseline models, scores are taken from their official publications when available.

Model	LLM	Frames	Tokens	Video-MME†	LongVideoBench	MLVU	EgoSchema
Open-source Models							
Video-LLaVA (Lin et al., 2024a)	Vicuna-7B	8	–	39.9	39.1	47.3	38.4
LLaMA-VID (Li et al., 2024d)	Llama-2-7B	1fps	–	25.9	–	33.2	38.5
LLaVA-NeXT-Video (Zhang et al., 2024c)	Vicuna-7B	32	–	–	43.5	–	43.9
LLaVA-NeXT-Video (Zhang et al., 2024c)	Qwen2-32B	32	–	60.2	–	65.5	60.9
VILA1.5 (Lin et al., 2024b)	Llama-2-40B	8	–	60.1	–	56.7	58.0
LLaVA-OneVision (Li et al., 2024a)	Qwen2-7B	32	–	58.2	56.4	64.7	60.1
Qwen2.5-VL (Bai et al., 2025)	Qwen2.5-7B	–	–	65.1	56.0	70.2	65.0
VideoLLaMA3 (Zhang et al., 2025)	Qwen2.5-7B	1fps	–	66.2	59.8	**73.0**	63.3
Proprietary Models							
Gemini-1.5-Flash (Gemini et al., 2024)	–	–	–	70.3	61.6	–	65.7
Gemini-1.5-Pro (Gemini et al., 2024)	–	–	–	**75.0**	64.0	–	71.2
GPT-4o (Hurst et al., 2024)	–	–	–	71.9	**66.7**	64.6	**72.2**
Models w/wo REGATE							
VideoChat2	Mistral-7B	16	3.93B	26.0	21.8	36.0	55.6
VideoChat2-REGATE	Mistral-7B	16	2.22B (↓ 43.51%)	32.7+6.7	24.3+2.5	40.5+4.5	54.8−0.8
VideoLLaMA2	Qwen2-7B	16	83.82M	53.7	47.7	53.2	58.2
VideoLLaMA2-REGATE	Qwen2-7B	16	49.27M (↓ 41.22%)	54.5+0.8	47.6−0.1	54.5+1.3	56.4−1.8
InternVL3.5	Qwen3-14B	16	3.96B	62.4	57.9	63.7	64.7
InternVL3.5-REGATE	Qwen3-14B	16	2.32B (↓ 41.41%)	63.0+0.6	58.0+0.1	64.2+0.5	63.9−0.8

video benchmarks, confirming that REGATE remains effective for high-capacity backbones.

On image understanding tasks requiring multimodal reasoning, all three models show significant gains. VideoLLaMA2-REGATE improves by 19.1% on ScienceQA and by up to 33.1 points on MME. VideoChat2-REGATE improves by 5.8% and 46.1 points on the same benchmarks. InternVL3.5-REGATE also shows clear improvements, including +7.7/+4.5 points on MME, 0.9% on VizWiz, and 1.6% on POPE. For long-video understanding, VideoChat2-REGATE shows strong improvements of 6.7% on Video-MME and 4.5% on MLVU. VideoLLaMA2-REGATE also improves, though more modestly, with gains of 0.8% and 1.3% on the same tasks. At the larger scale, InternVL3.5-REGATE further improves performance by 0.6% on Video-MME, and 0.5% on MLVU. Short video tasks benefit as well. VideoLLaMA2-REGATE improves by 1.6% on MVBench and 1.1% on Perception, and VideoChat2-REGATE gains 0.9% and 1.6%. InternVL3.5-REGATE also shows improvements, with +1.3% on MVBench and +1.4% on Perception. Beyond performance metrics, REGATE generalizes well across architectures and training strategies. It works effectively with transformer backbones such as Mistral, Qwen2, and Qwen3, and retains its benefits under both full-parameter and LoRA fine-tuning. These results confirm that REGATE's efficiency and accuracy gains are architecture- and training-strategy–agnostic, enabling seamless integration into diverse multimodal models.

Learning faster: ReGATE's efficiency gains. Table 4 summarizes efficiency gains in token usage, training time, and average accuracy on video benchmarks. Since REGATE introduces no architectural changes, per-layer FLOPs remain identical to the baseline.

For VideoLLaMA2-7B, REGATE matches baseline accuracy (48.0% vs. 48.2%) in only 64.0 GPU-hours, which is less than half the standard fine-tuning time (129.6 GPU-hours), while using 29.32M tokens (35% of the baseline's 83.82M). Extending training to 107.6 GPU-hours, which is still 22 hours fewer than the baseline, it processes 41.5% fewer tokens and surpasses the baseline with 48.9% accuracy. The one-time teacher pass adds only 2.1 GPU-hours, less than 2% of the baseline cost.

For VideoChat2-7B, which employs LoRA fine-tuning, time savings are smaller due to its already efficient backward pass. REGATE matches baseline accuracy (46.0% vs. 46.1%) in 86.4 GPU-hours, compared to 148.8 for the baseline, using 38% of the baseline tokens (1.51B vs. 3.93B). When extended to 130.0 GPU-hours, still 18.8 fewer than baseline, it processes 43.5% fewer tokens (2.22B vs. 3.93B) and improves accuracy to 47.8%. The teacher pass costs only 10.0 GPU-hours, about 7% of baseline.

Table 3: **Zero-shot evaluation results on short video understanding benchmarks.** Previous best results are highlighted in **bold**, while REGATE's best results are underlined. ‡ Results reported for Vinoground only for its video sub-task. For baseline models, scores are taken from their official publications when available.

Model	LLM	Frames	Tokens	MVBench	Perception	Vinoground‡	NExT-QA
Open-source Models							
Video-LLaVA (Lin et al., 2024a)	Vicuna-7B	8	–	41.0	44.3	25.8	–
LLaMA-VID (Li et al., 2024d)	Llama-2-7B	1fps	–	41.9	44.6	–	–
LLaVA-NeXT-Video (Zhang et al., 2024c)	Vicuna-7B	32	–	46.5	48.8	25.6	–
LLaVA-NeXT-Video (Zhang et al., 2024c)	Qwen2-32B	32	–	–	59.4	–	77.3
VILA1.5 (Lin et al., 2024b)	Llama-2-40B	8	–	–	54.0	–	67.9
LLaVA-OneVision (Li et al., 2024a)	Qwen2-7B	32	–	56.7	57.1	29.4	79.4
Qwen2.5-VL (Bai et al., 2025)	Qwen2.5-7B	–	–	69.6	70.5	–	–
VideoLLaMA3 (Zhang et al., 2025)	Qwen2.5-7B	1fps	–	**69.7**	**72.8**	–	**84.5**
Proprietary Models							
Gemini-1.5-Pro (Gemini et al., 2024)	–	–	–	60.5	–	22.6	–
GPT-4o (Hurst et al., 2024)	–	–	–	64.6	–	**38.2**	–
Models w/wo REGATE							
VideoChat2	Mistral-7B	16	3.93B	55.7	48.4	22.0	75.2
VideoChat2-REGATE	Mistral-7B	16	2.22B (↓ 43.51%)	56.6+0.9	50.0+1.6	22.8+0.8	75.5+0.3
VideoLLaMA2	Qwen2-7B	16	83.82M	52.0	53.0	24.6	70.8
VideoLLaMA2-REGATE	Qwen2-7B	16	49.27M (↓ 41.22%)	53.6+1.6	54.1+1.1	25.2+0.6	70.0−0.8
InternVL3.5	Qwen3-14B	16	3.96B	68.3	65.3	31.2	80.8
InternVL3.5-REGATE	Qwen3-14B	16	2.32B (↓ 41.41%)	69.6+1.3	66.7+1.4	31.2+0.0	81.2+0.4

Table 4: **Efficiency comparison of different models with REGATE.** Performance is measured as the average zero-shot accuracy (%) on video benchmarks. We report one-time teacher overhead and training time as GPU-hours (wall-clock hours × #GPUs).

Model	Tokens ↓	Teacher Cost (GPU-h)↓	Train Time (GPU-h)↓	Avg. Mem/GPU (GB)↓	Avg. Acc. (%) ↑
VideoLLaMA2	83.82M	-	129.6	69.1	48.2
VideoLLaMA2-REGATE	49.27M	2.1	107.6	**61.3**	**48.9**
VideoLLaMA2-REGATE	**29.32M**	2.1	**64.0**	-	48.0
VideoChat2	3.93B	-	148.8	70.8	46.1
VideoChat2-REGATE	2.22B	10.0	130.0	**63.7**	**47.8**
VideoChat2-REGATE	**1.51B**	10.0	**86.4**	-	46.0
InternVL3.5	3.96B	-	435.2	58.3	61.8
InternVL3.5-REGATE	2.32B	11.3	374.4	**51.9**	**62.2**
InternVL3.5-REGATE	**1.63B**	11.3	**262.4**	-	61.6

For the larger InternVL3.5-14B, REGATE maintains consistent gains. With aggressive pruning, it achieves near-baseline accuracy (61.6% vs. 61.8%) in 262.4 GPU-hours, a 40% speed-up over the baseline, using just 41% of baseline tokens (1.63B vs. 3.96B). Extending to 374.4 GPU-hours, it reduces token usage by 41.4% (2.32B vs. 3.96B) and surpasses the baseline at 62.2% accuracy. The teacher pass requires only 11.3 GPU-hours, less than 3% of baseline.

This speed-up difference across models stems from variations in training strategies and model scales. In full fine-tuning, as used in VideoLLaMA2, both forward and backward passes are computationally expensive. By pruning tokens, REGATE accelerates both passes, particularly the backward pass where gradients are computed for all parameters. In LoRA fine-tuning, as in VideoChat2, most parameters are frozen, so the backward pass is already efficient; REGATE primarily speeds up the forward pass, resulting in smaller total time savings. At the 14B scale, as in InternVL3.5, additional bottlenecks such as GPU memory and inter-GPU communication arise. REGATE reduces both the number of processed tokens and the size of intermediate activations, alleviating memory and communication costs. Consistent with these reductions, it lowers average per-GPU memory by 7–8 GB (about 10–11%) across all backbones: 69.1 → 61.3 GB on VideoLLaMA2, 70.8 → 63.7 GB on VideoChat2, and 58.3 → 51.9 GB on InternVL3.5. Overall, REGATE delivers significant efficiency gains across fine-tuning regimes and model scales, providing a flexible and effective approach for accelerating multimodal training without compromising performance.

How does REGATE compare with state-of-the-art training-time efficiency methods? We com-

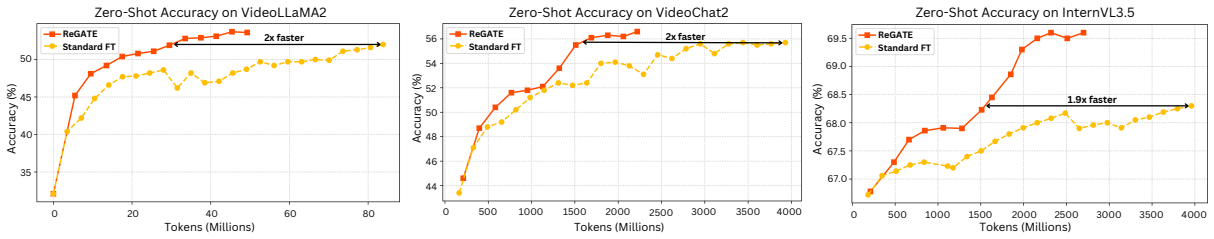


Figure 3: **Zero-shot accuracy on MVBench during fine-tuning.** REGATE (red) consistently outperforms standard fine-tuning (orange) at the same token count. It reaches the baseline's peak accuracy roughly twice as fast while using only 38% of the tokens on average, and surpasses the baseline with 41% fewer tokens.

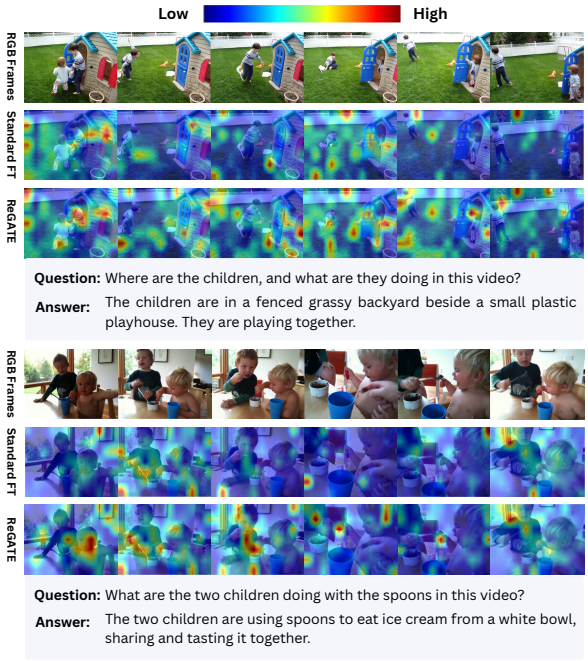


Figure 4: Attention maps from standard fine-tuning and REGATE on video QA tasks. REGATE focuses on contextually relevant regions (e.g., hands and manipulated objects), whereas standard fine-tuning spreads attention across the background.

pare REGATE with two representative approaches: LaVi (Yue et al., 2025) for video understanding and LLaVA-Meteor (li et al., 2025) for image understanding. LaVi introduces additional parameters and architectural changes and must be trained from scratch on the backbone, whereas REGATE is parameter-free and architecture-agnostic. For images, LLaVA-Meteor prunes tokens using heuristics and a tuned hyperparameter, while REGATE is fully self-adaptive, guided by token-level learning difficulty. All comparisons use comparable backbone sizes to ensure fairness. On video benchmarks (Table 5), despite using the weaker VideoLLaMA2-7B backbone, REGATE achieves competitive or superior results on two of four datasets compared to LaVi. On image benchmarks (Table 6), when applied to InternVL3.5-14B, REGATE consistently outperforms Meteor. These results highlight that REGATE delivers strong training-time efficiency without architectural changes or parameter overhead, while maintaining or improving accuracy across both video and image understanding tasks.

Table 5: **Comparison with LaVi.** LaVi results are reported from the original paper.

Method	LLM	VideoMME	MLVU	EgoSchema	MVBench
LLaVA-OneVision + LaVi	Qwen2-7B	54.0	**58.5**	55.5	**54.3**
VideoLLaMA2 + REGATE	Qwen2-7B	**54.5**	54.5	**56.4**	53.6

Table 6: **Comparison with LLaVA-Meteor.** LLaVA-Meteor results are from the original paper.

Method	LLM	VizWiz	POPE	SEED
LLaVA-Meteor	Vicuna-13B	55.3	87.2	64.8
InternVL3.5 + REGATE	Qwen3-14B	**60.5**	**93.1**	**76.6**

5 Conclusion

We introduced REGATE, a reference-guided token gating framework that accelerates the training of multimodal large language models. By combining a student model's learning difficulty with reference losses from a frozen text-only teacher, REGATE dynamically allocates computation to the most informative tokens while skipping those less relevant for multimodal understanding. The method is simple to implement, requires no architectural changes, and substantially improves training efficiency. Experiments show that REGATE achieves comparable or better accuracy than standard full fine-tuning, using only a fraction of the tokens and significantly less training time. Future work will explore adaptive scheduling for token sparsity by dynamically adjusting the retained token ratio based on task complexity, model stability, and training progress, starting with higher sparsity early and gradually relaxing it during fine-tuning.

Limitations

A limitation of the current REGATE is that it uses a fixed design for both sparsity scheduling and reference supervision. While this choice keeps the training pipeline simple and stable, it may leave room for improvement in more challenging settings. For example, dynamically adjusting the retained token ratio across training or according to input complexity could yield a better efficiency–performance trade-off. Similarly, extending the reference from a frozen text-only teacher to stronger or multimodal teachers may provide richer guidance, particularly for tasks requiring fine-grained spatial or temporal reasoning.

Acknowledgments

This research was supported by the National Eye Institute (NEI) of the National Institutes of Health (NIH) under award number R01EY034562. The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. 2021. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. In *Proceedings of the Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS)*.

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

Anthropic. 2025. The Claude 3.7 Sonnet system card.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. 2025. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models in resource-constrained environments. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*.

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, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Yuxuan Cai, Jiangning Zhang, Haoyang He, Xinwei He, Ao Tong, Zhenye Gan, Chengjie Wang, Zhucun Xue, Yong Liu, and Xiang Bai. 2025. Llava-kd: A framework of distilling multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024. 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 (ECCV)*.

Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. 2024. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. 2023. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems (NeurIPS)*.

Mohamed Dhouib, Davide Buscaldi, Sonia Vanier, and Aymen Shabou. 2025. Pact: Pruning and clustering-based token reduction for faster visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, and 2 others. 2025. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gemini, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, and 1 others. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. *arXiv preprint arXiv:1802.08218*.

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. 2025. Multi-granular spatio-temporal token merging for training-free acceleration of video llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Pengfei Jiang, Hanjun Li, Linglan Zhao, Fei Chao, Ke Yan, Shouhong Ding, and Rongrong Ji. 2025. Visa: Group-wise visual token selection and aggregation via graph summarization for efficient mllms inference. In *Proceedings of the 33rd ACM International Conference on Multimedia (ACM MM)*.

Yizhang Jin, Jian Li, Yexin Liu, Tianjun Gu, Kai Wu, Zhengkai Jiang, Muyang He, Bo Zhao, Xin Tan, Zhenye Gan, Yabiao Wang, Chengjie Wang, and Lizhuang Ma. 2024. Efficient multimodal large language models: A survey. *arXiv preprint arXiv:2405.10739*.

Yogesh Kulkarni and Pooyan Fazli. 2024. Videosavi: Self-aligned video language models without human supervision. In *Proceedings of the Conference on Language Modeling (COLM)*.

Yogesh Kulkarni and Pooyan Fazli. 2025. Avatar: Reinforcement learning to see, hear, and reason over video. *arXiv preprint arXiv:2508.03100*.

Jaewoo Lee, Keyang Xuan, Chanakya Ekbote, Sandeep Polisetty, Yi R. Fung, and Paul Pu Liang. 2025. Tamp: Token-adaptive layerwise pruning in multimodal large language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. 2021. Less is more: Clipbert for video-and-language learning via sparse sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024b. Seed-bench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Bonan li, Zicheng Zhang, Songhua Liu, Weihao Yu, and Xinchao Wang. 2025. Top-down compression: Revisit efficient vision token projection for visual instruction tuning. *arXiv preprint arXiv:2505.11945*.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. 2024c. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xinwei Li, Li Lin, Shuai Wang, and Chen Qian. 2023a. Unlock the power: Competitive distillation for multi-modal large language models. *arXiv preprint arXiv:2311.08213*.

Yanwei Li, Chengyao Wang, and Jiaya Jia. 2024d. Llama-vid: An image is worth 2 tokens in large language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. 2024a. Video-LLaVA: Learning united visual representation by alignment before projection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. 2024b. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Yelong Shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024c. Rho-1: Not all tokens are what you need. In *Proceedings of the Thirty-Eighth Conference on Neural Information Processing Systems (NeurIPS)*.

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

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024b. Llava-next: Improved reasoning, ocr, and world knowledge.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Proceedings of the Thirty-Sixth Conference on Neural Information Processing Systems (NeurIPS)*.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Khan. 2024. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Karttikeya Mangalam, Raiymbek Akshulakov, and Jitendra Malik. 2023. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, and 5 others. 2023. Perception test: A diagnostic benchmark for multimodal video models. In *Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Yumeng Shi, Quanyu Long, and Wenya Wang. 2025. Static or dynamic: Towards query-adaptive token selection for video question answering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Seungwoo Son, Jegwang Ryu, Namhoon Lee, and Jaeho Lee. 2024. The role of masking for efficient supervised knowledge distillation of vision transformers. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Ximeng Sun, Pengchuan Zhang, Peizhao Zhang, Hardik Shah, Kate Saenko, and Xide Xia. 2023. Dime-fm: Distilling multimodal and efficient foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Quan Tang, Bowen Zhang, Jiajun Liu, Fagui Liu, and Yifan Liu. 2023. Dynamic token pruning in plain vision transformers for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Keda Tao, Can Qin, Haoxuan You, Yang Sui, and Huan Wang. 2025. Dycoke: Dynamic compression of tokens for fast video large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vishaal Udandarao, Nikhil Parthasarathy, Muhammad Ferjad Naeem, Talfan Evans, Samuel Albanie, Federico Tombari, Yongqin Xian, Alessio Tonioni, and Olivier J. Hénaff. 2025. Active data curation effectively distills large-scale multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the Thirty-First Conference on Neural Information Processing Systems (NeurIPS)*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, Zhaokai Wang, Zhe Chen, Hongjie Zhang, Ganlin Yang, Haomin Wang, Qi Wei, Jinhui Yin, Wenhao Li, Erfei Cui, and 56 others. 2025. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *Proceedings of the Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track (NeurIPS)*.

Haoyu Wu, Jingyi Xu, Hieu Le, and Dimitris Samaras. 2025. Importance-based token merging for efficient image and video generation. *arXiv preprint arXiv:2411.16720*.

Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. 2021. Next-qa:next phase of question-answering to explaining temporal actions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shilin Xu, Xiangtai Li, Haobo Yuan, Lu Qi, Yunhai Tong, and Ming-Hsuan Yang. 2024. Llavadi: What matters for multimodal large language models distillation. *arXiv preprint arXiv:2407.19409*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, Ponnuswamy Sadayappan, Xia Hu, and Bo Yuan. 2025b. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Longrong Yang, Dong Shen, Chaoxiang Cai, Kaibing Chen, Fan Yang, Tingting Gao, Di Zhang, and Xi Li. 2025c. Libra-merging: Importance-redundancy and pruning-merging trade-off for acceleration plug-in in large vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. 2025. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tongtian Yue, Longteng Guo, Yepeng Tang, Zijia Zhao, Xinxin Zhu, Hua Huang, and Jing Liu. 2025. Lavi: Efficient large vision-language models via internal feature modulation. *arXiv preprint arXiv:2506.16691*.

Boqiang Zhang, Kehan Li, Zesen Cheng, Zhiqiang Hu, Yuqian Yuan, Guanzheng Chen, Sicong Leng, Yuming Jiang, Hang Zhang, Xin Li, Peng Jin, Wenqi Zhang, Fan Wang, Lidong Bing, and Deli Zhao. 2025. Videollama 3: Frontier multimodal foundation models for image and video understanding. *arXiv preprint arXiv:2501.13106*.

Jianrui Zhang, Mu Cai, and Yong Jae Lee. 2024a. Vinoground: Scrutinizing lmms over dense temporal reasoning with short videos. *arXiv preprint arXiv:2410.02763*.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2024b. Lmms-eval: Reality check on the evaluation of large multimodal models. *arXiv preprint arXiv:2407.12772*.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024c. Llava-next: A strong zero-shot video understanding model.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2025. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Appendix

A Additional Ablation Studies

In this section, we delve deeper into the design choices of REGATE, specifically examining the sensitivity of the weighting hyperparameter and the impact of the reference teacher's capacity.

A.1 Impact of the Weighting Factor λ

Table 7: **Ablation study on the weighting factor λ.** This parameter balances the student's EMA-based difficulty and the teacher's reference loss. Performance is reported as average zero-shot accuracy (%) on eight video benchmarks.

λ	**Description**	**Acc. (%)**
$\lambda = 0.0$	Student EMA Only	47.7
$\lambda = 1.0$	Reference Loss Only	46.4
$\lambda = 0.5$	**Combined Signals**	**48.9**

To validate the contributions of the individual components within our dual-signal token scoring mechanism, we conduct an ablation study on the hyperparameter λ. This coefficient balances the two core signals in our difficulty score formulation: $d_{b,i} = m_{s,i} + \lambda\, \ell^{\text{ref}}_{b,i}$, where $m_{s,i}$ is the student's dynamic EMA difficulty and $\ell^{\text{ref}}_{b,i}$ is the static reference loss from the teacher model. We evaluate three values for λ: 0.0, 0.5, and 1.0. The experiments use our VideoLLaMA2-REGATE setup with all other hyperparameters fixed for a fair comparison.

As shown in Table 7, setting $\lambda = 0.5$ achieves the best balance. Relying solely on the student's historical difficulty ($\lambda = 0.0$) fails to capture zero-shot visual dependencies, while relying exclusively on the teacher ($\lambda = 1.0$) ignores the student's training dynamics. The combined signal effectively isolates tokens that are both visually critical and challenging for the current model state.

A.2 Impact of Reference Teacher Capacity

Table 8: **Impact of Reference Teacher Capacity.** We compare teachers of varying sizes for a VideoLLaMA2-7B student. Performance is reported as average zero-shot accuracy (%) on eight video benchmarks.

Teacher Model	**Avg. Acc. (%)**
Qwen2-1.5B	45.4
Qwen2-57B	46.8
Qwen2-7B (**Ours**)	**48.9**

A critical design choice in REGATE is the selection of the reference teacher. We investigate how the capacity of the teacher model affects the student's performance. Specifically, for the VideoLLaMA2-7B (Cheng et al., 2024) student (based on Qwen2-7B (Yang et al., 2024)), we evaluate reference signals from a smaller model (Qwen2-1.5B) and a substantially larger model (Qwen2-57B-A14B), comparing them against the matched Qwen2-7B baseline. To ensure fair comparison, we restrict the study to a single model family, which guarantees consistent tokenization and avoids misalignment between teacher and student. Cross-architecture teachers (e.g., Mistral guiding Qwen) are excluded because distinct tokenizers yield inconsistent token boundaries for the same text, violating the one-to-one mapping required for per-token reference loss computation in REGATE. Such mismatch renders the reference signal mathematically invalid.

The results in Table 8 show that the matched Qwen2-7B teacher outperforms both the smaller 1.5B and the much larger 57B models. This pattern reveals a capacity mismatch effect: performance degrades when the teacher's capability diverges too far from the student's.

When using the large Qwen2-57B teacher, performance drops because the teacher is too strong relative to the student. With its vast world knowledge, the 57B model can often predict complex tokens purely from textual context, producing very low reference loss. However, the 7B student lacks this prior knowledge and instead depends on visual grounding to infer these tokens. As a result, the large teacher mistakenly signals REGATE to prune these positions, causing the student to skip crucial computations. Conversely, the smaller Qwen2-1.5B teacher underestimates the student's ability. It assigns high loss to linguistic patterns or factual details that the 7B student already handles easily. This floods the top-k selection with linguistically difficult but visually irrelevant tokens, wasting computation on trivial cases instead of focusing on visual reasoning.

Overall, these findings suggest that the optimal reference teacher should be capacity-aligned with the student. The reference loss is most effective when it reflects the student's own uncertainty, directing computation precisely where textual priors fall short.

Table 9: Prompts specifying the response format used for each evaluation benchmark.

Benchmark	Response formatting prompts
POPE	–
MME	Answer the question using a single word or phrase.
VisWiz	Answer the question using a single word or phrase. When the provided information is insufficient, respond with "Unanswerable".
ScienceQA	Answer with the option's letter from the given choices directly.
SEED	Answer with the option's letter from the given choices directly.
MLVU	–
MVBench	Only give the best option.
Video-MME	Answer with the option's letter from the given choices directly.
EgoSchema	Answer with the option's letter from the given choices directly.
NExT-QA	–
Perception	Answer with the option's letter from the given choices directly.
Vinoground	Please only output one English character.
LongVideoBench	Answer with the option's letter from the given choices directly.

B Additional Benchmarks Details

Table 9 lists the evaluation prompts corresponding to each benchmark used in the experiments, most of which are adapted from LMMs-Eval (Zhang et al., 2024b).

C Qualitative Analysis of Reference Loss

To validate the core mechanism of REGATE, we qualitatively analyze the reference loss signal that guides its token selection. We assume that a high loss score from the text-only teacher indicates that a token requires visual information to be understood. Figure 5 shows two video Q&A examples, visualizing the loss for each word in the answer as calculated by a Mistral-7B (Jiang et al., 2023) teacher model.

The results strongly support our assumption. As illustrated in the figure, tokens for visual details that are hard to guess from text alone, like the action "mixing" or the attribute "reflective", get high loss scores. In contrast, simple grammatical words like "The" and "is", or terms repeated from the question like "bartender", get low scores. This difference confirms that reference loss is a reliable indicator of visual importance, enabling REGATE to focus its computation on the most critical tokens for more efficient training.

Question: What is the bartender doing in the video?

Answer: The bartender is mixing a drink behind the counter in a bar setting.

Reference Loss:	"The"	"bartender"	"is"	"mixing"	"a"	"drink"	"behind"	"the"	"counter"	"in"
	3.20	0.03	0.04	8.50	4.18	8.31	10.13	0.56	10.87	6.75
	"a"	"bar"	"setting"	"."						
	2.70	7.94	6.38	2.00						

Question: What is the material of the floor in the bumper car area?

Answer: The floor in the bumper car area is of a reflective and glossy material.

Reference Loss:	"The"	"floor"	"in"	"the"	"bumper"	"car"	"area"	"is"	"of"	"a"	"reflective"
	1.96	5.34	1.79	0.08	5.03	1.15	2.70	0.33	4.78	0.34	12.37
	"and"	"glossy"	"material"	"."							
	2.01	7.40	6.41	1.45							

Question: What are the main highlights of the video?

Answer: The video shows a man melting down an object using a utensil that creates sparks. Another man walks behind him and puts on a helmet for protection before watching the process.

Reference Loss:	"The"	"video"	"shows"	"a"	"man"	"melting"	"down"	"an"	"object"	"using"	"a"
	1.03	2.11	1.53	0.10	2.94	11.51	3.11	0.61	4.93	3.71	0.13
	"utensil"	"that"	"creates"	"sparks"	"."	"Another"	"man"	"walks"	"behind"	"him"	"and"
	8.10	2.33	9.12	5.54	1.63	6.79	2.61	8.16	5.77	1.13	1.78
	"puts"	"on"	"a"	"helmet"	"for"	"protection"	"before"	"watching"	"the"	"process"	"."
	7.17	3.99	1.91	13.54	3.07	10.47	4.24	8.07	1.31	7.94	2.01

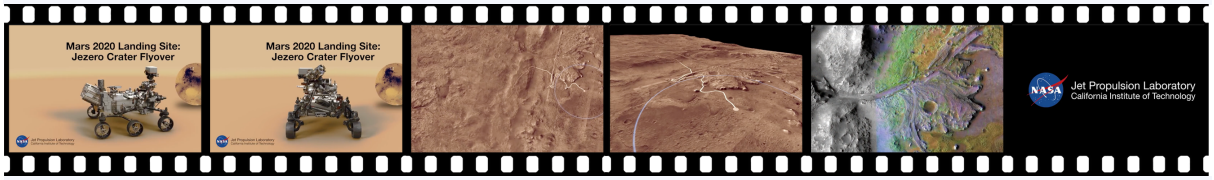
Figure 5: **Qualitative examples illustrating the effectiveness of the reference loss signal.** For two video Q&A pairs, we show the per-token reference loss computed by a text-only teacher model (Mistral-7B). Tokens colored in **red** have the highest losses and represent the top 50% most difficult tokens to predict from text alone. These are precisely the tokens that REGATE prioritizes for computation.

Question: How does the camera move in relation to the tiger at the end of the video?

Answer: The camera remains static as the tiger walks towards it through the dry, grassy landscape.

Token:	"The"	"camera"	"remains"	"static"	"as"	"the"	"tiger"	"walks"	"towards"	"it"
Reference Loss:	1.03	1.63	7.33	7.71	6.94	5.66	5.85	11.89	9.87	5.12
Student EMA loss:	2.01	3.11	8.04	5.35	4.64	2.30	8.26	9.10	9.33	3.09
Total loss:	3.04	4.74	15.37	13.06	11.58	7.96	14.11	20.99	19.20	8.21

Token:	"through"	"the"	"dry"	";"	"grassy"	"landscape"	"."
Reference Loss:	6.82	2.73	9.76	4.14	10.19	7.67	1.93
Student EMA loss:	8.03	1.91	7.53	2.87	11.34	9.12	0.98
Total loss:	14.85	4.64	17.29	7.01	21.53	16.79	2.91



Question: What is the title of the video and which organizations are mentioned in the opening scene?

Answer: The title of the video is Mars 2020 Landing Site: Jezero Crater Flyover and the organizations are NASA and the Jet Propulsion Laboratory California Institute of Technology.

Token:	"The"	"title"	"of"	"the"	"video"	"is"	"Mars"	"2020"	"Landing"	"Site"	":"	"Jezero"
Reference Loss:	0.97	2.88	1.16	0.54	2.14	1.08	9.39	10.63	9.63	6.25	10.33	13.94
Student EMA loss:	2.21	2.07	1.49	0.69	3.32	1.96	10.07	7.67	8.94	8.31	7.74	9.29
Total loss:	3.18	4.95	2.65	1.23	5.46	3.04	19.46	18.30	18.57	14.56	18.07	23.23

Token:	"Crater"	"Flyover"	"and"	"the"	"organizations"	"are"	"NASA"	"and"	"the"	"Jet"
Reference Loss:	12.94	15.98	5.81	3.85	3.00	2.11	15.71	6.52	3.99	11.64
Student EMA loss:	9.31	8.24	3.06	2.79	2.42	1.98	10.30	5.07	4.43	9.80
Total loss:	22.25	24.22	8.87	6.64	5.42	4.09	26.01	11.09	8.42	21.44

Token:	"Propulsion"	"Laboratory"	"California"	"Institute"	"of"	"Technology"	"."
Reference Loss:	10.09	11.77	9.17	8.98	3.03	8.37	2.00
Student EMA loss:	8.63	8.52	7.95	8.18	5.58	7.65	1.31
Total loss:	18.72	20.29	17.12	17.16	8.61	16.02	3.31

Figure 6: **Failure cases illustrating the limitation of fixed sparsity.** Tokens in **green** are retained by REGATE, while those in **red** are clearly helpful for answering but were not selected due to the fixed sparsity ratio (p=0.5). Some visually or semantically important tokens (e.g., static, NASA) are skipped, revealing the need for adaptive sparsity.