ReGATE: Learning Faster and Better with Fewer Tokens in MLLMs

Anonymous ACL submission

Abstract

The computational cost of training multimodal large language models (MLLMs) rapidly increases with the number of tokens involved. Existing efficiency methods primarily target inference and rely on token reduction or merging, offering limited benefit during training. In this paper, we propose ReGATE (**Re**ference-**G**uided **A**daptive **T**oken **E**lision), an adaptive token pruning method for accelerating MLLM training. Specifically, ReGATE adopts a teacher-student framework in which the MLLM being trained serves as the student, and a frozen reference large language model (LLM) acts as the teacher. The teacher computes per-token reference losses, which are combined with an exponential moving average (EMA) of the student's own difficulty scores. This adaptive difficulty-based scoring enables the selective processing of crucial tokens while bypassing less informative ones in the forward pass, significantly reducing computational overhead. Experiments demonstrate that ReGATE, when applied to VideoLLaMA2, matches the peak accuracy of standard training on MVBench up to 2× faster, using only 35% of the tokens. With additional training, it even surpasses the baseline on several multimodal benchmarks, all while reducing the total token count by over 41%. Code and models will be released soon.

1 Introduction

Multimodal large language models (MLLMs) face significant challenges due to the high computational cost of training. A key bottleneck is the self-attention mechanism, whose complexity grows quadratically with input sequence length (Vaswani et al., 2017). This problem is amplified in video tasks, where frames are tokenized into extremely long sequences. Consequently, training MLLMs on large-scale instructional datasets demands substantial computing resources, limiting accessibility and slowing progress in the field.

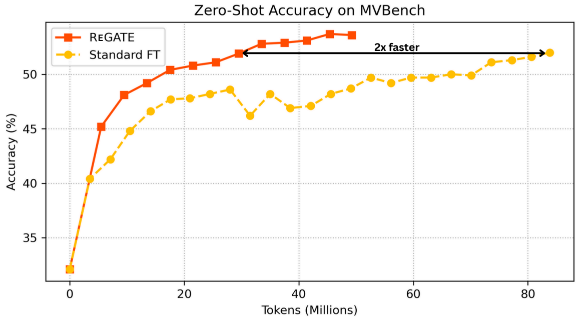

Figure 1: **Zero-shot accuracy on MVBench during fine-tuning of VideoLLaMA2-7B.** ReGATE (red) consistently outperforms standard fine-tuning (orange) at the same token count. It reaches the baseline's peak accuracy twice as fast while using only 35% of the tokens, and surpasses the baseline with just half the tokens.

Several strategies have been proposed to speed up inference in MLLMs, including static token pruning (Arif et al., 2025) and token merging (Chen et al., 2024). However, reducing the high cost of training remains a more complex and less explored challenge. In the unimodal text domain, recent work such as RHO-1 has introduced learnable token pruning techniques that improve training efficiency (Lin et al., 2024c). Yet, these training-time acceleration methods have not been extended to large multimodal models. Earlier attempts to improve visual processing efficiency, typically targeting standard vision transformers (Akbari et al., 2021) or early video-language models (Lei et al., 2021), have relied on heuristic approaches such as random token dropping. These methods fall short in modern MLLMs, as they fail to capture the subtle and often unintuitive cross-modal importance of tokens, particularly in video, where information is both dense and temporally distributed. As a result, such methods risk discarding important visual or semantic content, which can lead to unstable training and weaker multimodal understanding.

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. REGATE adopts a teacher-student architecture, where the student is the multimodal model being trained, and the teacher is a frozen, text-only version of the same LLM backbone. This setup enables REGATE to dynamically identify and retain the most informative tokens during training by combining two complementary signals. First, it assesses whether a token requires visual grounding by checking if the text-only teacher can accurately 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 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 image and video benchmarks demonstrate REGATE's broad applicability and efficiency. Notably, on the challenging MVBench benchmark, REGATE, when applied to VideoLLaMA2, matches the baseline's peak accuracy in just 16.0 hours (compared to 32.4 hours for standard fine-tuning) while processing only 29.3 million tokens, a 65% reduction from the baseline's 83.8 million (Figure 1).

2 Related Work

2.1 Token Compression for Fast Inference

Most existing work in the literature focus on accelerating inference, not training. Inference-time sparsity methods have shown that many tokens can be removed or merged with minimal impact on accuracy. In vision transformers, Dynamic Token Pruning(Tang et al., 2023) halts processing of "easy" tokens layer by layer, reducing FLOPs by 20–35% on semantic segmentation tasks without degrading performance. For video LLMs, DyCoke(Tao et al., 2025) dynamically compresses spatial-temporal tokens during inference, achieving up to 2× speedups while keeping model weights frozen. Moving from pruning to aggregation, Importance-Based Token Merging (Wu et al., 2025) merges highly 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 sequence 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) avoids processing long visual sequences by injecting vision-conditioned deltas—small, token-specific offsets derived from the visual input—into layer norms, eliminating most visual tokens, but this requires a specialized modulation pathway that must be trained from scratch. 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 vision tokens. In contrast, REGATE uniquely combines two complementary difficulty signals: a *static*, cross-modal reference loss from a frozen text-only teacher that identifies tokens requiring visual grounding, and a *dynamic* learning signal based on the student model's own token-wise loss tracked via an exponential moving average (EMA). This fusion of global and local difficulty enables a highly adaptive, parameter-free sparsity mechanism that gates both text and vision tokens, without modifying the underlying model architecture.

2.3 Teacher-Student Distillation for MLLMs

Most distillation approaches for MLLMs mainly focus on parameter compression. A systematic study (Xu et al., 2024) shows that jointly aligning tokens and logits helps a smaller student model in-

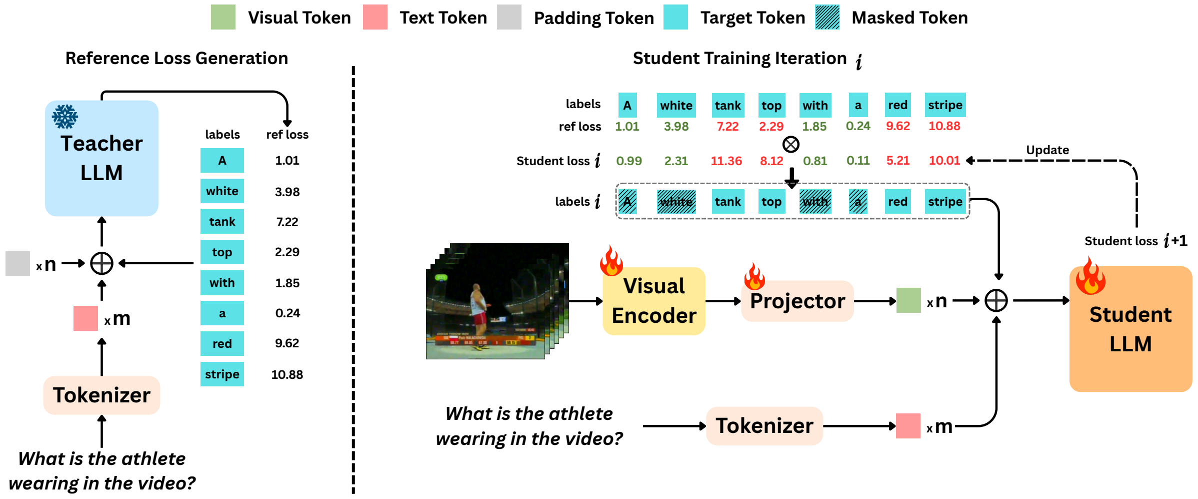
Visual Token
Text Token
Padding Token
Target Token
Masked Token
Reference Loss Generation
Teacher LLM
labels
ref loss
A 1.01
white 3.98
tank 7.22
top 2.29
with 1.85
a 0.24
red 9.62
stripe 10.88
×n
×m
Tokenizer
What is the athlete wearing in the video?
Student Training Iteration i
labels A white tank top with a red stripe
ref loss 1.01 3.98 7.22 2.29 1.85 0.24 9.62 10.88
Student loss i 0.99 2.31 11.36 8.12 0.81 0.11 5.21 10.01
Update
labels i A white tank top with a red stripe
Student loss i+1
Visual Encoder
Projector
×n
Student LLM
What is the athlete wearing in the video?
Tokenizer
×m

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.

herit 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. A more recent approach, MaskedKD (Son et al., 2024), improves efficiency by masking a portion of the image patch tokens fed to the teacher based on the student's attention scores. This strategy saves up to 50% of the teacher's FLOPs without reducing student accuracy. However, MaskedKD only sparsifies the teacher's computation and still requires backpropagation through all student tokens. In contrast, ReGATE introduces a fundamentally different approach by redefining the teacher's role in distillation. It uses the teacher's per-token loss to decide which tokens the student should process during each forward and backward pass. Instead of focusing on compressing the model itself, ReGATE targets compressing the computation path. This novel paradigm provides on-the-fly, modality-agnostic sparsity that optimizes the training process without changing the student's backbone architecture.

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 token positions 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 $\mathbf{x}_b = (x_{b,1}, \dots, x_{b,T})$ 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:

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

A low value of $\ell^{\text{ref}}_{b,i}$ 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:

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

where $\ell^{\text{stu}}_{b,i}$ 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 difficulty score for each token:

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

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 of the backbone LLM. We implement sparse attention by passing the mask directly as the attention mask to flash attention routines and by zeroing out the hidden states of pruned tokens. For the feed-forward network, we gather only the active positions, apply the MLP to them, and scatter the outputs back to their positions. The residual connections ensure that skipped tokens retain their previous representations. Algorithm 1 presents the pseudocode for a single forward decoder layer. This implementation requires no additional parameters and integrates seamlessly into popular libraries, such as HuggingFace Transformers. Importantly, our modifications do not affect the model architecture and thus remain compatible with pre-trained weights.

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 $\mathbf{H}$
```

4 Experiments

4.1 Implementation Details

To demonstrate the effectiveness of the proposed framework, we apply ReGATE to two different models (i.e., VideoLLaMA2 and VideoChat2) and training strategies. We select VideoChat2 and VideoLLaMA2 over newer models like Qwen-2.5-VL and VideoLLaMA3 because ReGATE assumes access to pretrained model weights for fine-tuning. However, in many cases, these weights are not publicly available, making it infeasible to apply methods like ReGATE directly. Training such models from scratch is also impractical, as many recent MLLMs rely on proprietary pretraining pipelines that require hundreds of GPUs, web-scale datasets, and access to private data. Nonetheless, with sufficient resources and access to pretrained weights and training data, ReGATE can be seamlessly integrated into the training pipeline of any modern MLLM.

VideoLLaMA2. We apply ReGATE to VideoLLaMA2-7B (Cheng et al., 2024), whose language backbone is Qwen2-7B (Yang et al., 2024). The model is initially pretrained with a frozen language backbone and subsequently fine-tuned on multimodal data. We introduce token gating during this fine-tuning stage, as the language backbone becomes trainable and can thus benefit from selective token updates. Specifically, the reference teacher model is obtained by removing the visual encoder and adapter from the VideoLLaMA2 backbone, resulting in a pure text-based LLM incapable of processing visual inputs. This teacher then computes token-wise losses, where all visual tokens have been replaced by padding.

VideoChat2. To assess ReGATE's effectiveness in parameter-efficient fine-tuning (PEFT) scenarios, we integrate our method into the LoRA-based Stage 3 training of VideoChat2-7B (Li et al., 2024c), which uses a Mistral-7B backbone. Our key adaptation here is to make the LoRA update process itself token-selective. In a conventional setup, the loss used to update the LoRA adapters is aggregated over all tokens. In our approach, the gradients for the LoRA parameters are computed exclusively from the subset of high-importance tokens identified by ReGATE. This ensures that the parameter-efficient updates are concentrated on the most informative signals, while the original language backbone weights remain frozen. The reference teacher is derived from the text-only Mistral-7B (Jiang et al., 2023) backbone, following the same procedure as described previously.

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. 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. 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 are run on 4 H100 GPUs using mixed-precision training.

4.2 Evaluation Benchmarks

To evaluate ReGATE, we use a diverse suite of benchmarks across image, long-video, and short-video domains. All evaluations are conducted under LMMs-Eval's[1] settings. All benchmarks used in our evaluation follow their respective licenses and are consistent with their intended use. Below, we briefly summarize the key characteristics of each benchmark.

[1] https://github.com/EvolvingLMMs-Lab/lmms-eval

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$_{-0.3}$

Image understanding. **ScienceQA** (Lu et al., 2022) is a multimodal science exam with 21,208 multiple-choice questions and accompanying lectures and explanations; **MME** (Fu et al., 2024) measures perception and cognition across 14 subtasks using manually created question–answer pairs; **VizWiz** (Gurari et al., 2018) collects real photos taken by blind users and asks questions about them and whether they are answerable; **POPE** (Li et al., 2023) is an object hallucination benchmark formulated as a binary-choice task; and **SEED-Bench** (Li et al., 2024b) includes 19 thousands multiple-choice questions covering both image and video modalities across 12 dimensions.

Long-video understanding. **Video-MME** (Fu et al., 2025) spans six primary domains and 30 subfields with videos ranging from 11 seconds to 1 hour; it integrates frames, subtitles and audio and provides 2,700 expert-annotated question–answer pairs for holistic evaluation. **LongVideoBench** (Wu et al., 2024) contains 3,763 videos (up to an hour) and 6,678 multiple-choice questions, many of which require referring to specific temporal segments before reasoning. **MLVU** (Zhou et al., 2025) collects long videos from diverse genres, including movies, surveillance, and egocentric recordings, and offers multiple tasks. Studies show that existing models degrade with longer context. **EgoSchema** (Mangalam et al., 2023) comprises more than 5,000 three-minute clips from 250 hours of egocentric data, with questions requiring reasoning over much longer temporal windows than previous datasets, and current models perform far below human level.

Short-video understanding. **MVBench** (Li et al., 2024c) converts 20 static image tasks into dynamic video tasks, producing multiple-choice questions that probe temporal understanding. **Perception Test** (Pătrăucean et al., 2023) consists of 11,600 real-world videos averaging 23 seconds; it evaluates perception and reasoning across six annotation types and emphasises skills such as memory, abstraction, and physics. **Vinoground** (Zhang et al., 2024a) comprises 1,000 short video–caption pairs designed for counterfactual temporal reasoning, where even large proprietary models struggle to distinguish subtle action differences. **NExT-QA** (Xiao et al., 2021) offers 5,440 videos and about 52,000 questions targeting causal and temporal action reasoning.

4.3 Baseline Models

We evaluate REGATE against a comprehensive set of baselines, including the adapted VideoLLaMA2 (Cheng et al., 2024) and VideoChat2 (Li et al., 2024c) models. Our comparison covers a broad range of state-of-the-art open-source models, primarily drawn from high-performing families such as LLaVA and Qwen. We also report results from proprietary models in the Google Gemini,

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 VideoMME are reported without subtitles. For baseline models, scores are taken from their official publications when available.

Model	**LLM**	**Frames**	**Tokens**	**VideoMME**†	**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., 2024b)	Vicuna-7B	32	–	–	43.5	–	43.9
LLaVA-NeXT-Video (Zhang et al., 2024b)	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

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., 2024b)	Vicuna-7B	32	–	46.5	48.8	25.6	–
LLaVA-NeXT-Video (Zhang et al., 2024b)	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

OpenAI GPT, and Anthropic Claude series. This diverse set of baselines spans multiple LLM backbones and model sizes, ensuring a robust and meaningful comparison. The specific models evaluated across image and video tasks are listed in Tables 1, 2, and 3.

4.4 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 and VideoChat2 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. 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.

On image understanding tasks that require multimodal reasoning, both 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. For long

Table 4: **Efficiency comparison of different models with ReGATE.** All models are trained using 4 H100 GPUs. Performance is measured as zero-shot accuracy (%) on MVBench.

Model	Tokens ↓	Train Time ↓	Acc. (%) ↑
VideoLLaMA2	83.82M	32.4h	52.0
VideoLLaMA2-ReGATE	49.27M	26.9h	**53.6**
VideoLLaMA2-ReGATE	**29.32M**	**16.0h**	51.9
VideoChat2	3.93B	37.2h	55.7
VideoChat2-ReGATE	2.22B	32.5h	**56.6**
VideoChat2-ReGATE	**1.51B**	**21.6h**	55.5

video understanding, VideoChat2-ReGATE shows strong improvements of 6.7% on VideoMME and 4.5% on MLVU. VideoLLaMA2-ReGATE also improves, though more modestly, with gains of 0.8% and 1.3% on the same tasks. Short video tasks benefit as well. VideoLLaMA2-ReGATE improves by 1.6% on MVBench and 1.1% on Perception, while VideoChat2-ReGATE gains 0.9% and 1.6%, respectively.

Overall, these results demonstrate ReGATE's ability to adapt across diverse tasks by efficiently directing computational resources to the most important visual and semantic content.

Learning faster: ReGATE's efficiency gains. Table 4 presents detailed efficiency gains in token usage, training time, and accuracy on the MVBench benchmark.

For VideoLLaMA2, ReGATE closely matches the baseline accuracy (51.9% vs. 52%) in just 16.0 hours, which is less than half the time required for standard fine-tuning (32.4 hours). It does so using only 29.32 million tokens, approximately 35% of the 83.82 million tokens used by the baseline. When training is extended to 26.9 hours (still 5.5 hours less than the baseline), ReGATE processes 41.51% fewer tokens and achieves a higher accuracy of 53.6%.

For VideoChat2, which uses parameter-efficient LoRA fine-tuning, the improvements in training time are more modest. Specifically, ReGATE closely matches the baseline accuracy (55.5% vs. 55.7%) in 21.6 hours, compared to 37.2 hours for the baseline. Furthermore, when training time increases to 32.5 hours (still 4.7 hours less than the baseline), ReGATE processes 43.51% fewer tokens (2.22 billion vs. 3.93 billion) and achieves an improved accuracy of 56.6%.

This speed-up difference between VideoLLaMA2 and VideoChat2 arises from the contrast between full and LoRA fine-tuning strategies. In full fine-tuning, as used in VideoLLaMA2, both forward and backward passes through the model are computationally expensive. By pruning tokens, ReGATE speeds up both passes, especially the backward pass where gradients are computed for all model parameters. In LoRA fine-tuning, as used in VideoChat2, most parameters are frozen, and the backward pass is already efficient since gradients are only computed for a small number of adapter parameters. While ReGATE still accelerates the forward pass through the frozen backbone, the total time savings are smaller because the backward pass is not a bottleneck. Overall, ReGATE delivers significant gains in both token efficiency and training time across different training strategies, making it a flexible and effective solution for reducing computation without compromising performance.

5 Conclusion and Future Work

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 focuses computation on 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. These gains come without compromising model quality. In fact, ReGATE consistently outperforms baselines across a wide range of image and video benchmarks, demonstrating strong data efficiency and generalization. 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 (e.g., starting with higher sparsity early and relaxing it as fine-tuning progresses). We will also investigate fine-grained sparsity control across layers or attention heads for more efficient resource allocation. Moreover, we aim to generalize the notion of "reference" beyond frozen text-only teachers. Using stronger or multimodal teacher models (e.g., vision/video-language) could provide richer supervision for gating, improving cross-modal alignment and enhancing performance on complex spatial and temporal tasks.

Limitations

Due to computational resource constraints, we validate REGATE on 7B-parameter models, VideoLLaMA2 and VideoChat2, demonstrating clear effectiveness and efficiency gains at this scale. However, the full potential of REGATE likely emerges with larger models (e.g., 30B or 70B+ parameters) and massive, web-scale datasets, where time and cost savings become more significant. Future work should focus on evaluating REGATE's performance and scalability in such high-resource settings.

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)*.

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*.

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)*.

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*.

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*.

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)*.

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. 2023. 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)*.

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)*.

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)*.

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, 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*.

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*.

Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. 2024b. 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 Ablation Study

How does each signal in our scoring mechanism affect performance? 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: $d_{b,i} = m_{s,i} + \lambda \, \ell_{b,i}^{\mathrm{ref}}$, where $m_{s,i}$ is the student's dynamic EMA difficulty and $\ell_{b,i}^{\mathrm{ref}}$ is the static reference loss from the teacher model. By varying λ, we can isolate the impact of each signal.

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 5, $\lambda = 0.5$, which balances the reference loss and the student's EMA-based difficulty, results in the best performance.

Table 5: **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 zero-shot accuracy (%) on MVBench.

λ	**Description**	**Acc. (%)**
$\lambda = 0.0$	Student EMA Only	51.3
$\lambda = 1.0$	Reference Loss Only	51.1
$\lambda = 0.5$	**Combined Signals**	**53.6**

B 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 3 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.

C Additional Benchmarks Details

Table 6 lists the evaluation prompts corresponding to each benchmark used in the experiments.

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"
3.20	0.03	0.04	8.50	4.18	8.31	10.13	0.56

"counter"	"in"	"a"	"bar"	"setting"	"."
10.87	6.75	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"
1.96	5.34	1.79	0.08	5.03	1.15	2.70	0.33	4.78

"a"	"reflective"	"and"	"glossy"	"material"	"."
0.34	12.37	2.01	7.40	6.41	1.45

Figure 3: **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.

Table 6: **Summary of the evaluation benchmarks.** Prompts are mostly borrowed from LMMs-Eval.

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-Bench	Answer with the option's letter from the given choices directly.
MLVU	–
MVBench	Only give the best option.
VideoMME	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.