DUETTO: DUAL-CHANNEL EFFICIENT TOKEN SELECTION FOR LARGE VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Vision-Language Models introduce significant computational challenges, as encoding images and videos into thousands of vision tokens imposes substantial computational costs. Prior work addresses this challenge by proposing methods to identify and select the most informative vision tokens while preserving model performance. Determining which vision tokens are important requires considering two signals: the information a token contributes relative to the rest of the vision token sequence and its relevance to the instruction. Yet, existing methods fail to jointly optimize both for the downstream task: heuristic approaches rely on fixed ranking rules that cannot adapt to the downstream effects of token selection, while learnable approaches either base selection solely on visual content or defer instruction relevance to a later stage. To address this, we propose DUETTO (**Du**al-channel **E**fficien**t** **To**ken selection), a vision token selector trained end-to-end that fuses a vision channel and an instruction-relevance channel into a single selection score. DUETTO achieves state-of-the-art performance, with the advantage over prior methods increasing as the token budget becomes more restrictive, maintaining 99.1% of full-model accuracy on 33.3% of the vision tokens and, at high resolution, 96.5% using 5.6% of the vision sequence, accelerating prefill by $3.2\times$ while the selection costs only 6% of prefill time. Finally, experimental results show that DUETTO generalizes to diverse backbones with both image and video inputs, achieving state-of-the-art results.

1 INTRODUCTION

Building upon the outstanding performance of Large Language Models (LLMs) (OpenAI, 2023; Llama Team, 2024; Qwen Team, 2025; DeepSeek-AI, 2024), Large Vision-Language Models (LVLMs) have emerged as the natural extension to approach cross-modal tasks (Liu et al., 2023; Qwen Team, 2026; Wang et al., 2025; Wu et al., 2024), including visual question answering and captioning. These models typically encode the visual input through a vision encoder into a sequence of vision tokens, which are then projected into the same feature space as the language model, enabling the interaction between both modalities.

This performance comes at a cost, as the resulting vision token sequence is long, with a single low-resolution image spanning hundreds of tokens (Liu et al., 2024a; Li et al., 2026), and several thousand for high-resolution images (Liu et al., 2024b) or video inputs (Lin et al., 2024), often far outnumbering the number of text instruction tokens. Recent work has shown that vision tokens are highly redundant (Chen et al., 2024a), and since standard self-attention scales quadratically with sequence length (Vaswani et al., 2017), these unnecessary tokens dominate the cost of inference and limit the deployment of LVLMs in resource-constrained scenarios (Jin et al., 2025). As a result, vision token compression has become an important area of research in LVLMs, aiming to increase efficiency while preserving performance (Zhang et al., 2025c; Alvar et al., 2025; Zhang et al., 2025b; Fang et al., 2026; Baek et al., 2026; Yang et al., 2025; Huang et al., 2025; Zhu et al., 2025).

To determine which tokens carry the most important information, we need to take into consideration whether a token is visually salient, containing information that the remaining tokens do not already carry, as well as whether it is relevant to the user instruction. Both signals are illustrated in Figure 1, where instruction relevance localizes the tokens covering the queried object in the first example, whereas in the second example a generic instruction yields little relevance, and it is the

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

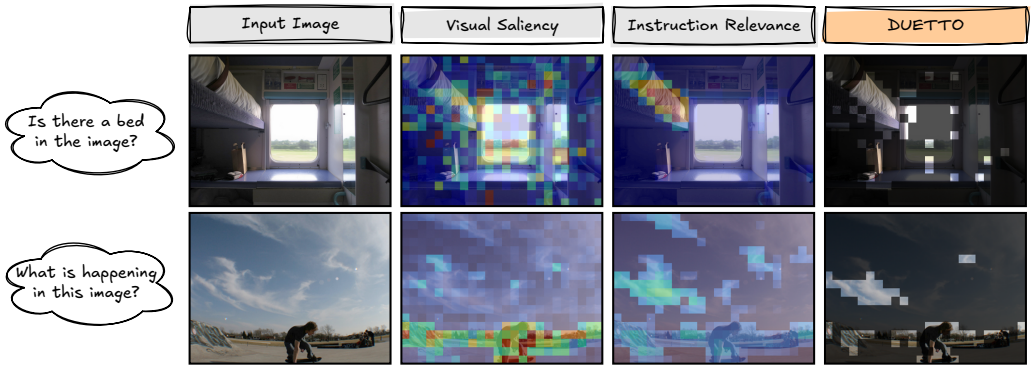


Figure 1: Two selection signals for vision token compression, on a targeted question **(top)** and on a generic instruction **(bottom)**, each failing in one of the two scenarios while DUETTO retains the informative tokens in both.

visual saliency that identifies the tokens worth keeping. Neither signal suffices in every scenario, and few methods target both at once, motivating a selection that incorporates both signals.

Existing vision token compression methods can be broadly categorized as heuristic and learnable methods. Heuristic approaches rank tokens using signals such as visual diversity, saliency, and instruction relevance (Zhang et al., 2025c; Alvar et al., 2025; Zhang et al., 2025b; Fang et al., 2026; Baek et al., 2026; Yang et al., 2025), but rely on fixed ranking rules and do not optimize token selection for its downstream effect on model output. Learnable approaches instead train lightweight modules for token selection (Huang et al., 2025; Takezoe et al., 2026; Ye et al., 2025; Zhu et al., 2025), but either omit the instruction from the selection process (Huang et al., 2025; Zhu et al., 2025), making the decision conditioned exclusively on visual content, or perform selection at multiple decoder stages (Takezoe et al., 2026; Ye et al., 2025), introducing additional computation that sacrifices efficiency gains of compression.

To address these limitations, we propose DUETTO (**Du**al-channel **E**fficien**t** **To**ken selection), a learnable method optimized for vision token selection that combines two complementary channels: a vision channel that projects visual tokens into a learned scoring space, where a self-attention mechanism captures token redundancy and favors distinctive features across the vision token sequence, optimized to the effect selection has on model output, and an instruction-relevance channel that measures the affinity between vision and instruction tokens, retaining both sharp agreement with a single decisive instruction token and broad agreement with the instruction as a whole through a smoothed, local-neighborhood scoring strategy. The two signals are fused into a unified estimate to perform budget-exact differentiable dual-channel selection.

In summary, our contributions are threefold:

- **DUETTO.** A novel learnable vision token selector that combines complementary vision and instruction-relevance signals into a single budget-exact selection. Through a vision channel that models redundancy across vision tokens, and an instruction-relevance channel that conditions selection on the user instruction, the resulting fusion enables context-dependent instruction-relevant token selection in a single pass at an early layer.
- **Dual-Channel Selection.** While the vision channel redundancy modeling provides the basis for vision token selection, the proposed instruction-relevance signal conditions and refines this basis into a finer-grained, instruction-relevant selection. The resulting combined dual-channel selection pushes results beyond the state-of-the-art.
- **Generalization.** Extensive evaluations validate DUETTO on 11 image and video benchmarks over 6 backbones, where it holds the highest relative accuracy at every token budget, with the advantages over other methods increasing as the token budget tightens, retaining **99.1%** of the full-token model on 33.3% of the vision tokens, and, at high resolution, **96.5%** on 5.6%, while accelerating prefill **3.2**$\times$ with a selection costing 6% of it, placing DUETTO on the Pareto front of the accuracy-efficiency trade-off.

2 Related Work

Heuristic Methods. Heuristic methods rely primarily on rules based on signals produced during the model's forward pass, without optimizing token selection for the downstream task. Some approaches leverage encoder `[CLS]` attention (Arif et al., 2025) to rank token importance and merge non-dominant tokens (Shang et al., 2025; Yang et al., 2025). However, high-norm tokens can dominate `[CLS]` attention, as noted by Darcet et al. (2024). Other methods measure redundancy between vision tokens based on feature similarity (Bolya et al., 2023; Alvar et al., 2025; Wen et al., 2025b), but do not account for the relevance of visual tokens to the user instruction. Notably, Chen et al. (2024b) instead observe that vision-token attention becomes increasingly redundant in deeper decoder layers and use the attention received by vision tokens at an early layer to guide pruning. Other approaches similarly use attention scores that vision tokens receive from other tokens (Zhang et al., 2025c; Xing et al., 2024). However, Zhang et al. (2025a); Wen et al. (2025a) note that these methods can suffer from attention shift, which compromises selection accuracy. Moreover, relying on attention scores can make such methods incompatible with efficient attention implementations such as FlashAttention (Dao, 2024). More recent heuristic methods combine multiple criteria, such as diversity with instruction relevance (Zhang et al., 2025b) or encoder saliency (Fang et al., 2026; Baek et al., 2026). However, these methods still rely on fixed selection rules and cannot adapt their scoring to the downstream objective.

Learnable Methods. In contrast, learnable methods train lightweight modules for token selection, allowing the selection decision to be optimized for the task. Dynamic-LLaVA (Huang et al., 2025) trains predictors over the vision and output text tokens under the language-modeling objective, while VisionSelector (Zhu et al., 2025) scores projected vision features with a learned module trained using a budget-exact differentiable top-k. However, both methods keep the selection guided exclusively by visual content. ATP-LLaVA (Ye et al., 2025) learns instance-specific thresholds at three points in the decoder, while LearnPruner (Takezoe et al., 2026) performs selection in two stages, first after the vision encoder and again within the LLM to account for instruction relevance. This two-stage design, however, adds computation. In contrast, the fine-tuned variant of VisionZip‡ (Yang et al., 2025) keeps the original training-free selection rule while adapting the connector to its pruned input.

3 Method

3.1 Preliminaries and Problem Formulation

Standard Large Vision-Language Models consist of a vision encoder f_v, a connector g, and a language model f_ϕ with L decoder layers (Liu et al., 2024a; Li et al., 2026). Given an image X_v, the vision encoder produces N vision tokens, which the connector maps to the D-dimensional embedding space of f_ϕ as $\mathbf{H}_v = g(f_v(X_v)) \in \mathbb{R}^{N\times D}$. The resulting vision tokens are concatenated with the M instruction token embeddings $\mathbf{H}_q \in \mathbb{R}^{M\times D}$ and passed through the language model to generate a response. We partition the decoder at layer ℓ as $f_\phi = f_\phi^{(\ell:L-1)} \circ f_\phi^{(0:\ell-1)}$, where the $f_\phi^{(\ell:L-1)}$ block includes the output head, and our goal is to retain an informative subset of $k \ll N$ vision tokens while preserving the model's full-token performance. Let $\mathbf{H}^{(\ell)} = f_\phi^{(0:\ell-1)}([\mathbf{H}_v; \mathbf{H}_q])$ denote the layer-ℓ hidden states of the full sequence, split into the vision block $\mathbf{H}_v^{(\ell)}$ and the instruction block $\mathbf{H}_q^{(\ell)}$. The optimal set of k vision tokens is defined as

$$\mathcal{S}^\star = \underset{\mathcal{S}\subseteq\mathbf{H}_v^{(\ell)},\ |\mathcal{S}|=k}{\arg\min}\ \mathcal{L}_{\mathrm{LM}}\Big(f_\phi^{(\ell:L-1)}\Big(\big[\mathcal{S};\ \mathbf{H}_q^{(\ell)}\big]\Big),\ Y\Big), \tag{1}$$

where $\mathcal{S}$ retains its tokens in their original order, the instruction states are carried through in full, and $\mathcal{L}_{\mathrm{LM}}$ is the cross-entropy loss over the target response Y.

3.2 DUETTO

The main mechanism is a lightweight selector at decoder layer ℓ. Layers 0 to $\ell-1$ process the full sequence, after which the selector scores and prunes vision tokens. Layers ℓ to $L-1$ then process the reduced sequence. Acknowledging the findings of Chen et al. (2024b), we place the selector early, with $\ell \ll L$, so that only a few layers process the full sequence.

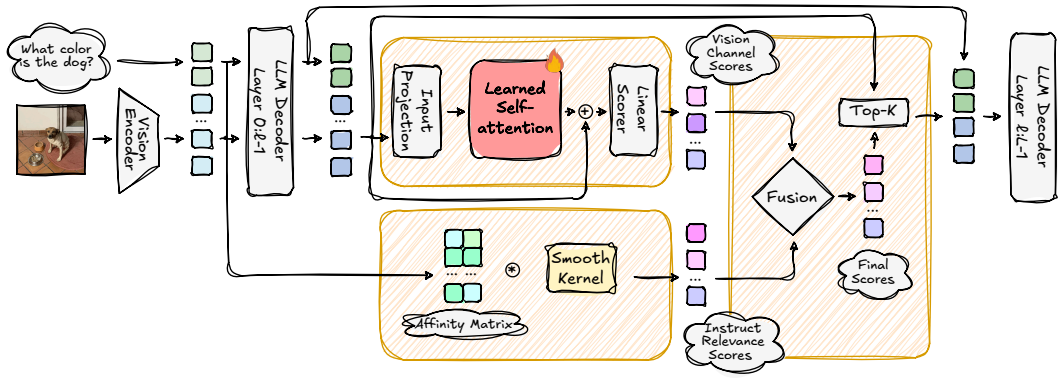


Figure 2: Overview of DUETTO. Decoder layers 0 to $\ell-1$ run on the full sequence. At layer ℓ, the **vision channel** scores each token through an input projection, a learned self-attention block, and a linear scorer, while the **instruction-relevance channel** smooths the cosine similarity between the vision and instruction tokens. The fused score feeds a budget-exact top-k, and layers ℓ to $L-1$ run on the k retained vision tokens.

The selector computes a single decision score per token by fusing two channels:

- **Vision channel** (§3.2.1): projects the vision tokens into a learned scoring space, where a trainable self-attention mechanism identifies redundancy within the vision tokens and assigns each token a score.
- **Instruction-relevance channel** (§3.2.2): measures the relevance of each vision token to the input instruction by computing their similarity in the language model's input-embedding space, resulting in a per-token score.

We train the method end-to-end, combining the two scores (§3.2.3) and passing them to a budget-exact differentiable top-k operator (Zhu et al., 2025).

3.2.1 Vision Channel

The vision channel scores each token from the layer-ℓ hidden states $\mathbf{H}_v^{(\ell)} \in \mathbb{R}^{N\times D}$ of the full vision sequence. Since these representations are optimized for next-token prediction and not for token scoring, the channel first re-embeds each token through a learned linear projection and a GELU nonlinearity (Hendrycks & Gimpel, 2016) into a D_a-dimensional scoring space,

$$\mathbf{Z} = \mathrm{GELU}\Big(\mathrm{LN}(\mathbf{H}_v^{(\ell)})\,\mathbf{W}_{\text{in}}\Big) \in \mathbb{R}^{N\times D_a}, \qquad \mathbf{W}_{\text{in}} \in \mathbb{R}^{D\times D_a}, \tag{2}$$

where LN is layer normalization (Ba et al., 2016) over the feature dimension. A single-head self-attention block then mixes the re-embedded tokens across the span, entering through a pre-norm residual. A linear readout then collapses each mixed token into a scalar score,

$$\mathbf{Z}' = \mathbf{Z} + \mathrm{SelfAttn}(\mathrm{LN}(\mathbf{Z}))\,, \qquad h_i = \mathbf{Z}'_i\,\mathbf{w}_s + b_s, \tag{3}$$

with score-head weight vector $\mathbf{w}_s \in \mathbb{R}^{D_a}$ and bias $b_s \in \mathbb{R}$, whose per-token outputs form the vision score $\mathbf{h} \in \mathbb{R}^N$. The bidirectional self-attention allows each score h_i to capture contextual relationships between vision tokens, enabling the vision channel to favor tokens that provide distinctive visual information.

3.2.2 Instruction-Relevance Channel

The connector g, optimized to map the vision tokens into the D-dimensional space that the first decoder layer consumes (Liu et al., 2023), places the vision tokens alongside the language model's token embeddings $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}|\times D}$. Let $\mathbf{q} = [q_1, \ldots, q_M]$ be the token IDs of the user instruction, which index $\mathbf{E}$ to give the instruction embeddings, normalized row-wise to unit norm as are the projected vision tokens:

$$\mathbf{T} = \mathrm{Normalize}\big(\mathbf{E}[\mathbf{q}]\big) \in \mathbb{R}^{M\times D}, \qquad \mathbf{V} = \mathrm{Normalize}\big(\mathbf{H}_v\big) \in \mathbb{R}^{N\times D}, \tag{4}$$

yielding the cosine-similarity matrix $\mathbf{C} = \mathbf{V}\mathbf{T}^\top \in \mathbb{R}^{N\times M}$ between the vision and instruction tokens. A token can be relevant in two ways: through broad agreement with the instruction as a whole, or through sharp agreement with one decisive token. Thus, we reduce over the instruction axis by mean and by max, normalizing each to $[0, 1]$, and taking their per-token maximum, as illustrated in Figure 8.

$$c_i^{\text{mean}} = \tfrac{1}{M}\textstyle\sum_{j=1}^{M} C_{ij}, \qquad c_i^{\max} = \max_j C_{ij}, \qquad a_i = \max(\hat{c}_i^{\text{mean}}, \hat{c}_i^{\max}), \tag{5}$$

where $\hat{c}_i$ is the min-max normalized score over the N vision tokens of a sample, so that the two reductions reach the maximum on a common scale, and a strong single-token match and a strong average match are both able to prevail.

Relevant image content is spatially contiguous, as neighboring positions on the token grid tend to carry similar content (Liu et al., 2024c), while the per-token cosine of Equation 5 is computed independently and is therefore noisy, producing isolated responses. To exploit this contiguity, we reshape the affinities $\mathbf{a}$ onto the token grid and convolve them with a small smoothing kernel $\boldsymbol{\kappa}$ over each token's neighborhood, then min-max normalize over the sample, resulting in the relevance score $\hat{\mathbf{r}} \in [0, 1]^N$, whose effect is further studied in Appendix A.1.2.

3.2.3 Fusion and Budget-Exact Selection

In order to use the score produced by each channel, we combine the two channels into one score per token, with relevance expressed in units of the learned score's spread $\rho = \max_i h_i - \min_i h_i$, held constant in the backward pass,

$$s_i = h_i + \alpha\, \rho\, \hat{r}_i. \tag{6}$$

Measuring the relevance term against ρ makes α an exchange rate between the two channels, as relevance can displace a token by at most α times the spread of the learned score.

Keeping the k highest scores meets the budget exactly but carries no gradient, so we pair it with the sigmoid-bisection relaxation of Zhu et al. (2025) during training, which offsets the scores until the mask they induce carries the budget's mass,

$$m_i = \sigma(s_i + \tau), \qquad \text{with } \tau \text{ the unique solution of } \textstyle\sum_{i=1}^{N} \sigma(s_i + \tau) = k. \tag{7}$$

The tokens that continue are the k highest-scoring ones, so the selector approximates the optimum $\mathcal{S}^\star$ of Equation 1 by $\mathcal{S} = \text{top-}k(\mathbf{s})$. Unlike the original operator, we train with no auxiliary rate or constraint loss. The mask scales the retained tokens, and the gradient concentrates on the tokens near the selection boundary, further explained in Appendix A.1.3.

3.2.4 Training the Vision Channel under Instruction Relevance

The vision channel is trained end-to-end, coupled with the instruction-relevance signal, as the gradient flows through the fused score s_i, which carries the relevance term $\hat{r}_i$. The vision tokens are pruned at layer ℓ, while the shortened sequence continues through layers ℓ to $L-1$, and the language-modeling loss returns through the top-k backward pass into the channel's parameters. Since this is the only gradient those parameters receive, they are fit to the effect the ranking has on the model output, which is exactly the objective of Equation 1. What the channel learns is therefore conditioned on where the relevance signal has placed its mass, since that signal decides which tokens survive to produce the gradient. We isolate the effect of this coupling on object localization in the ablation study of §4.4.

4 Experiments

4.1 Experimental Setup

Following Yang et al. (2025); Takezoe et al. (2026), we train the selector for one epoch on a 10% subset of the LLaVA-665K dataset (Liu et al., 2024a), placing it at decoder layer ℓ=2 and fusion weight α=0.5. For the video backbone, we use video instruction data (Maaz et al., 2024). Further details are provided in Appendix A.1. To assess DUETTO's generalization, we evaluate it across diverse architectures and modalities: we use LLaVA-1.5 (Liu et al., 2024a) for general image tasks,

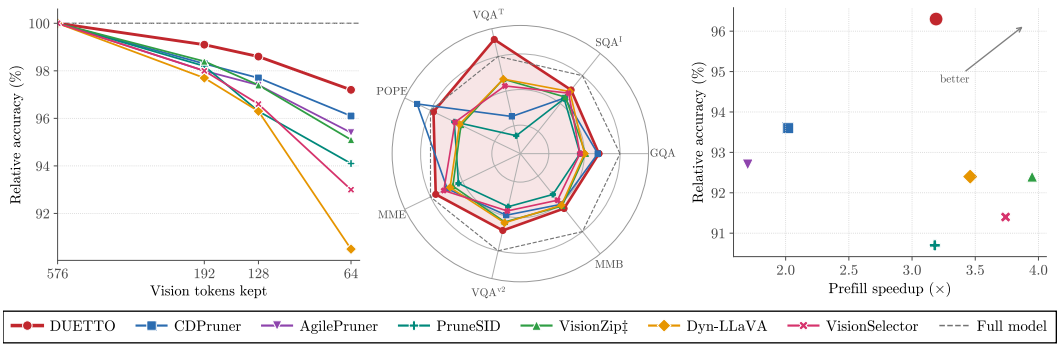


Figure 3: Overview of results. **(left)** Relative accuracy on LLaVA-1.5-7B (mean over the benchmarks of each model's per-benchmark ratio to the full-token model) against the retained token budget. **(middle)** Per-benchmark retention on LLaVA-NeXT-7B at the 320-token budget. **(right)** Accuracy-efficiency trade-off on LLaVA-NeXT-7B at 160 tokens.

LLaVA-NeXT (Liu et al., 2024b) for high-resolution settings, Mini-Gemini (Li et al., 2026) for a distinct vision architecture following Fang et al. (2026) and Yang et al. (2025), and Video-LLaVA (Lin et al., 2024) for video inputs following Yang et al. (2025), Fang et al. (2026) and Takezoe et al. (2026). Extended results including Qwen2.5-VL (Bai et al., 2025) and LLaVA-1.5-13B are provided in Appendix A.2.

Experiments were done on 11 standard multimodal benchmarks (Zhang et al., 2025b; Takezoe et al., 2026; Fang et al., 2026; Yang et al., 2025), of which 7 are image benchmarks, namely GQA (Hudson & Manning, 2019), ScienceQA (Lu et al., 2022), TextVQA (Singh et al., 2019), POPE (Li et al., 2023), MME (Fu et al., 2025), VQAv2 (Goyal et al., 2017), and MMBench (Liu et al., 2024d), and 4 are video understanding benchmarks, comprising MSVD-QA (Xu et al., 2017), MSRVTT-QA (Xu et al., 2017), TGIF-QA (Jang et al., 2017), and ActivityNet-QA (Yu et al., 2019). We compare several recent open-source methods, covering both heuristic and learnable methods (Zhang et al., 2025c; Alvar et al., 2025; Zhang et al., 2025b; Fang et al., 2026; Baek et al., 2026; Yang et al., 2025; Huang et al., 2025).

4.2 Results and Discussion

Figure 3 summarizes the comparison across three views: performance over the token budget, performance across individual benchmarks, and the trade-off between accuracy and inference cost. Across token budgets, DUETTO remains closest to the full-token LLaVA-1.5 performance, while the baselines are close at 192 tokens but spread by 9.5 points at 64 tokens, particularly among the learned methods, highlighting the importance of robust token selection. At 320 tokens on LLaVA-NeXT, the per-benchmark view shows DUETTO enclosing the other methods on almost every axis and tracing the highest overall area contour. The accuracy-prefill-speedup view on the same backbone shows no baseline above and to the right of DUETTO, indicating that none of the compared methods achieves both higher accuracy and higher prefill speedup.

Taken together, these results demonstrate the robust performance of DUETTO across a variety of budgets, backbones, and inputs. DUETTO achieves the highest relative accuracy in every evaluated scenario, while consistently lying on the Pareto front of the accuracy-efficiency trade-off, highlighting its ability to maintain superior accuracy while remaining highly competitive in efficiency.

We first apply DUETTO to LLaVA-1.5, the standard backbone for evaluating vision token compression, retaining 192, 128 and 64 of the 576 vision tokens it encodes. Across all three budgets, DUETTO achieves the strongest relative accuracy (RelAcc), with its lead becoming increasingly pronounced as the budgets tightens. As shown in Table 1, the lead over CDPruner, the closest competing method, grows from 0.8 to 0.9 and ultimately 1.1 points, highlighting the precise selection under a tight budget. At 192 tokens, 33.3% of the sequence, DUETTO maintains **99.1%** full-token accuracy, ahead of previous learnable methods Dynamic-LLaVA and VisionSelector by 1.4 and 1.1 points respectively, and ahead of the diversity-based competitors CDPruner and PruneSID, at 98.3% and 98.2%. Retaining 128 tokens, DUETTO preserves **98.6%** relative accuracy, widening the gap

Table 1: Performance on LLaVA-1.5-7B. RelAcc represents the average percentage of performance maintained. The best results on each budget are **bolded**.

Method	GQA	SQA^I	VQA^T	POPE	MME	VQA^{v2}	MMB	Acc.	RelAcc
Upper Bound, 576 Tokens (100%)									
LLaVA-1.5-7B	61.9	69.3	58.1	85.8	1862	78.5	64.7	69.2	100.0%
Retain 192 Tokens (33.3%, ↓ 66.7%)									
SparseVLM	57.6	**69.1**	56.1	83.6	1721	75.6	62.5	66.6	96.0%
DivPrune	60.0	68.7	56.4	87.0	1752	75.5	62.3	67.5	97.3%
CDPruner	60.2	68.8	57.1	**87.2**	1800	77.3	62.6	68.2	98.3%
PruneSID	60.1	68.5	56.7	86.9	1791	76.8	**63.7**	68.1	98.2%
AgilePruner	59.9	68.5	57.7	85.8	1782	77.1	62.8	67.9	98.0%
VisionZip‡	59.8	68.6	**58.0**	84.6	1813	77.4	63.4	68.1	98.3%
Dyn-LLaVA	60.2	68.2	57.8	81.8	1844	77.2	62.3	67.6	97.7%
VisionSelector	60.0	68.5	57.1	85.9	1791	77.0	63.3	68.0	98.0%
DUETTO	**61.3**	68.6	57.7	85.0	**1859**	**77.8**	63.5	**68.6**	**99.1%**
Retain 128 Tokens (22.2%, ↓ 77.8%)									
SparseVLM	56.0	67.1	54.9	80.5	1696	73.8	60.0	64.7	93.3%
DivPrune	59.3	69.0	56.1	86.9	1752	76.0	62.3	67.5	97.2%
CDPruner	59.9	68.8	56.0	87.3	1781	76.6	62.4	67.8	97.7%
PruneSID	58.8	68.3	54.7	86.5	1749	75.3	62.1	66.9	96.3%
AgilePruner	59.4	68.6	57.0	**87.4**	1748	76.4	61.8	67.6	97.4%
VisionZip‡	59.1	**69.1**	57.0	83.3	1822	76.5	63.0	67.6	97.6%
Dyn-LLaVA	58.2	68.4	57.1	79.1	1818	76.0	62.6	66.6	96.3%
VisionSelector	58.8	68.7	56.0	84.0	1780	75.8	61.8	66.9	96.6%
DUETTO	**60.9**	67.9	**57.6**	84.9	**1844**	**77.3**	**63.4**	**68.3**	**98.6%**
Retain 64 Tokens (11.1%, ↓ 88.9%)									
SparseVLM	52.7	62.2	51.8	75.1	1505	68.2	56.2	60.0	86.6%
DivPrune	57.8	68.3	55.5	85.6	1674	74.1	60.1	65.9	94.9%
CDPruner	**58.8**	68.0	55.2	**87.3**	1730	**75.5**	60.6	66.7	96.1%
PruneSID	57.1	67.8	54.2	83.8	1733	73.7	58.8	65.3	94.1%
AgilePruner	57.4	68.6	56.0	84.1	1703	**75.5**	60.7	66.2	95.4%
VisionZip‡	57.3	69.0	55.9	80.7	1782	74.1	60.9	65.9	95.2%
Dyn-LLaVA	53.6	69.0	54.4	67.7	1698	70.6	60.8	62.4	90.5%
VisionSelector	56.0	**69.4**	53.7	80.3	1664	72.5	60.0	64.5	93.0%
DUETTO	**58.8**	68.8	**56.7**	84.6	**1817**	75.4	**62.2**	**67.3**	**97.2%**

to Dynamic-LLaVA and VisionSelector to 2.3 and 2.0 points, while staying ahead of the fine-tuned VisionZip‡ by 1.0 point. The gap is widest at 64 tokens, where DUETTO preserves **97.2%** while Dynamic-LLaVA and VisionSelector fall to 90.5% and 93.0%, below the heuristic CDPruner and VisionZip‡ at 96.1% and 95.2%, emphasizing the need for a robust learned selection.

4.2.1 GENERALIZATION

High-Resolution Inputs. We next apply DUETTO to LLaVA-NeXT, whose tiling divides the image into sub-images, producing a longer and more redundant token sequence of up to 2,880 tokens. In Table 2, we compress this sequence to 22%, 11% and 5.6% of the total tokens. DUETTO holds the best relative accuracy at every budget, leading the strongest baseline by 1.0, 2.2 and 2.9 points as the sequence shortens, and comes within 0.3 points of the full-token model at 640 tokens with **99.7%** relative accuracy, while still preserving **96.5%** once only 5.6% of the sequence survives.

Different Architectures. To assess the generalizability of DUETTO, we evaluate on Mini-Gemini, which pairs the encoder of LLaVA-1.5 with a high-resolution ConvNeXt (Liu et al., 2022) to inject fine-grained information into the 576 image tokens. As Table 3 shows, it holds the best relative accuracy at every budget, from **99.5%** at 192 tokens to **97.2%** at 64 tokens, and once again, the margin over the strongest previous method is more pronounced as the budget decreases, from 0.5 to 2.4 points. These results indicate that DUETTO consistently identifies informative vision tokens, regardless of the underlying vision encoder used to produce them.

Table 2: Performance on LLaVA-NeXT-7B.

Method	GQA	SQA^I	VQA^T	POPE	MME	VQA^{v2}	MMB	RelAcc.
Upper Bound, 2880 Tokens (100%)								
LLaVA-NeXT-7B	64.1	70.2	61.3	86.3	1834	81.8	67.5	100.0%
Retain 640 Tokens (22.2%, ↓ 77.8%)								
SparseVLM	60.3	67.7	57.8	85.3	1772	77.1	65.7	96.0%
CDPruner	**63.7**	67.9	57.8	**87.7**	1807	79.6	66.2	98.1%
PruneSID	61.6	68.3	54.7	86.3	1795	78.5	64.2	95.9%
AgilePruner	62.7	67.9	58.8	86.6	1820	79.2	66.7	98.0%
VisionZip‡	62.4	67.9	60.8	86.0	1778	79.9	65.9	97.9%
Dyn-LLaVA	62.6	68.6	61.3	85.4	1819	80.3	66.2	98.5%
VisionSelector	63.2	68.2	60.2	85.8	1848	79.6	**67.0**	98.7%
DUETTO	63.6	**69.2**	**63.5**	86.7	**1857**	**80.7**	65.1	**99.7%**
Retain 320 Tokens (11.1%, ↓ 88.9%)								
SparseVLM	57.7	67.3	55.9	76.9	1694	73.4	64.3	91.9%
CDPruner	62.1	67.4	56.0	**88.1**	1786	77.6	64.2	96.2%
PruneSID	60.5	67.3	54.3	83.1	1754	76.6	63.0	94.0%
AgilePruner	61.2	67.5	57.5	86.7	1770	77.8	63.4	95.9%
VisionZip‡	61.0	67.5	59.3	82.2	1770	78.4	64.4	95.9%
Dyn-LLaVA	60.9	68.2	59.3	82.4	1778	78.5	64.3	96.1%
VisionSelector	60.5	68.0	58.7	83.0	1796	77.1	63.7	95.7%
DUETTO	**62.2**	**68.4**	**62.8**	85.9	**1819**	**79.4**	**64.7**	**98.4%**
Retain 160 Tokens (5.6%, ↓ 94%)								
SparseVLM	51.2	67.5	46.4	77.3	1542	66.3	63.1	85.7%
CDPruner	59.8	67.0	53.5	**88.0**	**1713**	75.2	62.2	93.6%
PruneSID	58.9	67.1	52.0	76.9	1704	73.8	60.8	90.7%
AgilePruner	57.3	67.3	56.7	82.7	1662	77.2	61.1	92.7%
VisionZip‡	58.2	67.5	57.3	74.8	1699	75.6	**63.9**	92.4%
Dyn-LLaVA	58.1	68.7	55.3	78.7	1694	75.1	62.6	92.4%
VisionSelector	57.2	67.3	57.1	79.7	1636	73.7	60.8	91.4%
DUETTO	**61.5**	**69.0**	**61.6**	84.9	1707	**78.1**	63.4	**96.5%**

Table 3: Performance on Mini-Gemini-7B.

Method	GQA	SQA^I	VQA^T	POPE	MME	MMB	RelAcc.
Upper Bound, 576 Tokens (100%)							
Mini-Gemini-7B	62.6	71.3	66.6	87.0	1868	77.0	100.0%
Retain 192 Tokens (33.3%, ↓ 66.7%)							
VisionZip‡	61.2	69.9	64.7	86.2	**1922**	76.5	99.0%
Dyn-LLaVA	59.0	**71.0**	65.3	83.0	1823	**76.9**	97.5%
DUETTO	**61.9**	70.8	**65.8**	**87.6**	1855	76.7	**99.5%**
Retain 128 Tokens (22.2%, ↓ 77.8%)							
VisionZip‡	60.0	70.1	63.4	85.2	1839	**76.3**	97.5%
Dyn-LLaVA	56.8	71.0	64.0	80.2	1734	75.7	95.0%
DUETTO	**60.9**	**71.0**	**65.0**	**87.5**	**1859**	76.2	**98.9%**
Retain 64 Tokens (11.1%, ↓ 88.9%)							
VisionZip‡	57.3	**70.9**	61.2	82.0	1747	**75.3**	94.8%
Dyn-LLaVA	52.0	69.5	59.0	70.2	1574	71.7	87.9%
DUETTO	**59.1**	70.5	**64.7**	**85.7**	**1821**	74.5	**97.2%**

Table 4: Performance on Video-LLaVA.

Method	MSVD	MSRVTT	TGIF	ANet	RelAcc.
Upper Bound, 2048 Tokens (100%)					
Video-LLaVA-7B	72.9	63.1	68.6	47.7	100.0%
Retain 452 Tokens (22.0%, ↓ 78.0%)					
VisionZip‡	**72.8**	62.1	68.6	**47.1**	99.3%
Dyn-LLaVA	71.3	63.5	67.2	45.8	98.1%
DUETTO	72.7	**65.0**	**69.6**	45.7	**100.0%**
Retain 226 Tokens (11.0%, ↓ 89.0%)					
VisionZip‡	**72.8**	60.7	65.7	43.6	95.8%
Dyn-LLaVA	72.3	**63.0**	64.6	**45.4**	97.1%
DUETTO	71.8	62.5	**69.0**	45.3	**98.3%**

Video Inputs. To extend experiments to video tasks, we implement DUETTO in Video-LLaVA, which encodes eight sampled frames into 2,048 tokens, compressed to 22% and 11% in Table 4. DUETTO leads at both budgets, remarkably matching the full-token model at 22% with **100.0%** relative accuracy and preserving **98.3%** at 11%, ahead of Dynamic-LLaVA and VisionZip‡.

4.3 Efficiency Analysis

High-resolution image encoding leaves vision-language models with a prefill dominated by the image, as LLaVA-NeXT-7B can turn a single input into 2880 vision tokens and spend 248.7 ms during prefill. We measure inference cost on a single NVIDIA A100-40GB, reducing the sequence to 160 tokens, and we report prefill and total time, selection latency, theoretical FLOPs following Chen et al. (2024b), and peak memory usage in Table 5, whose measurement details are given in Appendix A.1.4. At a compression rate of 5.6%, DUETTO accelerates prefill, the stage where vision

Table 5: Inference efficiency analysis and comparison on LLaVA-NeXT-7B.

Method	#Tok	Prefill ms (×) ↓	Total ms (×) ↓	Select. ms (%) ↓	FLOPs T ↓	Mem GB ↓	Acc. % ↑
LLaVA-NeXT-7B	2880	248.7 (1.0×)	822.9 (1.0×)	0.0 (0%)	17.00	15.7	100.0
CDPruner	160	123.2 (2.0×)	697.1 (1.2×)	56.3 (46%)	1.19	14.0	93.6
PruneSID	160	78.1 (3.2×)	657.7 (1.3×)	16.4 (21%)	**1.07**	14.7	90.7
AgilePruner	160	145.9 (1.7×)	722.9 (1.1×)	57.8 (40%)	1.09	14.7	92.7
VisionZip‡	160	**62.9 (4.0×)**	**629.3 (1.3×)**	4.8 (8%)	**1.07**	14.6	92.4
Dyn-LLaVA	160	71.8 (3.5×)	644.7 (1.3×)	**1.0 (1%)**	2.06	13.6	92.4
VisionSelector	160	66.5 (3.7×)	641.0 (1.3×)	5.7 (9%)	1.23	**13.5**	91.4
DUETTO	160	78.0 (3.2×)	653.8 (1.3×)	4.5 (6%)	2.06	13.8	**96.5**

token compression acts, from 248.7 to 78.0 ms, a **3.2**× reduction, while bringing end-to-end latency from 822.9 to 653.8 ms and peak memory from 15.7 to 13.8 GB, while retaining **96.5**% of the full-token accuracy, the highest of the comparison. The selection costs 4.5 ms, only 6% of prefill, whereas the training-free heuristic baselines spend between 21% and 46% with iterative per-token procedures, and among the learnable methods only Dynamic-LLaVA selects more cheaply, at 1% of prefill, giving up 4.1 points of relative accuracy, while VisionSelector spends more on selection and falls behind by 5.1 points. As Figure 3 shows, no method we compare against is both faster and more accurate, making DUETTO Pareto-optimal in the accuracy-efficiency trade-off.

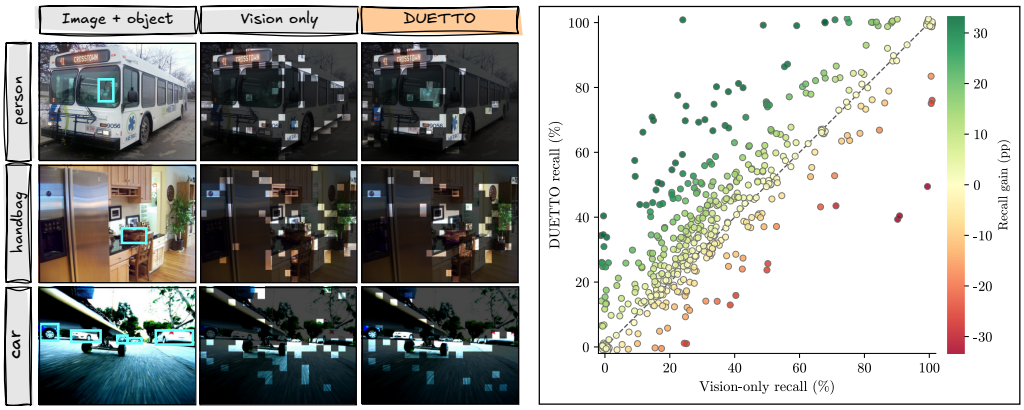


Figure 4: Vision-only selection versus DUETTO at the 64-token budget. **(left)** Tokens retained by each method, with the instruction object's bounding box highlighted in cyan. **(right)** Per-sample object recall, measured as the fraction of vision tokens within the ground-truth bounding box that are retained, on 500 samples. Each point is colored by its recall gain (DUETTO − vision-only).

4.4 Ablation Study

Table 6: Ablation study of the score components.

Selector	128 tokens RelAcc.	128 tokens Δ	64 tokens RelAcc.	64 tokens Δ
DUETTO	**98.6%**	—	**97.2%**	—
(I) instruction-only	70.8%	−27.8	65.6%	−31.6
(II) vision-only	97.9%	−0.7	95.8%	−1.4
(III) w/o self-attention	98.0%	−0.6	94.8%	−2.4
(IV) w/o spatial smoothing	98.2%	−0.4	96.6%	−0.6

To assess the contribution of each channel and their individual components, we evaluate four variants of DUETTO applied to LLaVA-1.5 at the 128- and 64-token budgets: **(I)** an instruction-only selection that scores tokens by instruction relevance alone, **(II)** a vision-only ranking that removes the instruction channel, **(III)** replacing the trained self-attention in the vision channel with a per-token MLP, and **(IV)** removing the spatial smoothing of the instruction-relevance map. The instruction-only ranking **(I)** collapses to 70.8% and 65.6% relative accuracy at the 128 and 64 token budgets, respectively, showing that instruction relevance alone is insufficient to replace the vision channel. Removing the instruction channel **(II)** reduces relative accuracy by 0.7 and 1.4 points, while replacing self-attention with a per-token MLP **(III)** reduces it by 0.6 and 2.4 points, respectively. The larger degradation at the tighter budget indicates that scoring becomes increasingly important as fewer tokens can be retained, where selection errors have a greater impact on model output. Finally, removing the spatial smoothing of the instruction-relevance map **(IV)** costs a comparable 0.4 and 0.6 points. We further examine how the instruction-relevance channel performs when it is learned through a cross-attention mechanism between modalities in Appendix A.3.2. Figure 4 illustrates the difference in selected tokens between a vision-only selector and DUETTO, where the vision-only selector frequently overlooks the object specified by the instruction, leading to the accuracy drop of **(II)**. DUETTO recovers tokens corresponding to the queried object and improves recall on 72.8% of samples.

5 Conclusion

In this paper, we introduce DUETTO, a learnable vision token selector that fuses a trainable vision channel with an instruction-relevance channel into a single score, driven through a budget-exact differentiable top-k operator in one pass at an early decoder layer. The vision channel provides the basis for that selection by modeling redundancy across the vision tokens, and training the channel under the instruction-relevance signal sharpens this basis into a finer-grained selection that also retains the tokens the instruction calls for. Across 11 image and video benchmarks and 6 backbones, DUETTO holds the highest relative accuracy at every evaluated budget, with the advantage over prior methods widening as the budget tightens, placing DUETTO on the Pareto front of the accuracy-efficiency trade-off. Future work could investigate input-adaptive weighting of the two channels, allowing their relative contributions to respond to the visual content and instruction, while also adapting the token budget to the complexity of each input.

6 AI USE STATEMENT

In this work, we used generative AI tools to assist with writing to improve readability by improving grammar and text clarity. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, or interpret results, create or modify scientific figures or images, suggest experimental parameters, create or edit software code, creation of artifacts, draft parts of a research paper, transcribe recordings of research material, summarize or analyse existing literature, discover research topics or identify gaps, brainstorming, sourcing/searching for information, identify relevant literature, format references, suggest a structure for a research paper, propose a title or keywords for a research paper. Usage of AI tools to generate synthetic datasets, assist with translation, clean and reformat datasets, or formulate questions for surveys or interviews is not applicable to our work. All AI-assisted work was thoroughly reviewed, verified, and edited by the authors, who take full responsibility for the final content of this work, including text produced with the aid of generative AI.

7 ETHICS STATEMENT

This work studies vision token compression to improve the efficiency of Large Vision-Language Models. The research process does not violate the ICLR Code of Ethics, which includes responsible research, avoidance of harm, transparency, fairness, privacy, and respect for intellectual contributions. The proposed method does not involve human subjects, personal data collection, or sensitive information, and we are not aware of any specific risks of discrimination, bias, or unfair treatment arising from the research process or its intended use. The methodology and experimental results are reported accurately and transparently.

8 REPRODUCIBILITY STATEMENT

This research exclusively uses publicly available, open-source backbones and datasets, with all sources appropriately referenced. Experimental settings and implementation details are provided in Section 4 and Appendix A.1. To further support reproducibility, we will release the source code and model checkpoints.

REFERENCES

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 9392–9401. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.00877. URL https://openaccess.thecvf.com/content/CVPR2025/html/Alvar_DivPrune_Diversity-based_Visual_Token_Pruning_for_Large_Multimodal_Models_CVPR_2025_paper.html.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In Toby Walsh, Julie Shah, and Zico Kolter (eds.), *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pp. 1773–1781. AAAI Press, 2025. doi: 10.1609/AAAI.V39I2.32171. URL https://doi.org/10.1609/aaai.v39i2.32171.

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. URL http://arxiv.org/abs/1607.06450.

Changwoo Baek, Jou Won Song, Sohyeon Kim, and Kyeongbo Kong. Agilepruner: An empirical study of attention and diversity for adaptive visual token pruning in large vision-language models. *CoRR*, abs/2603.01236, 2026. doi: 10.48550/ARXIV.2603.01236. URL https://doi.org/10.48550/arXiv.2603.01236.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *CoRR*, abs/2502.13923, 2025. doi: 10.48550/ARXIV.2502.13923. URL https://doi.org/10.48550/arXiv.2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=JroZRaRw7Eu.

David L. Chen and William B. Dolan. Collecting highly parallel data for paraphrase evaluation. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, pp. 190–200. The Association for Computer Linguistics, 2011. URL https://aclanthology.org/P11-1020/.

Jieneng Chen, Luoxin Ye, Ju He, Zhaoyang Wang, Daniel Khashabi, and Alan L. Yuille. Efficient large multi-modal models via visual context compression. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024a. URL http://papers.nips.cc/paper_files/paper/2024/hash/871ed095b734818cfba48db6aeb25a62-Abstract-Conference.html.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXI*, volume 15139 of *Lecture Notes in Computer Science*, pp. 19–35. Springer, 2024b. doi: 10.1007/978-3-031-73004-7_2. URL https://doi.org/10.1007/978-3-031-73004-7_2.

Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=mZn2Xyh9Ec.

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=2dnO3LLiJ1.

DeepSeek-AI. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024. doi: 10.48550/ARXIV.2412.19437. URL https://doi.org/10.48550/arXiv.2412.19437.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in vlms via synergistic importance-diversity. *CoRR*, abs/2603.09480, 2026. doi: 10.48550/ARXIV.2603.09480. URL https://doi.org/10.48550/arXiv.2603.09480.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and

Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/d79a27cf2772fe00be7f341efc0eb517-Abstract-Datasets_and_Benchmarks_Track.html.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 6325–6334. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.670. URL https://doi.org/10.1109/CVPR.2017.670.

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pp. 961–970. IEEE Computer Society, 2015. doi: 10.1109/CVPR.2015.7298698. URL https://doi.org/10.1109/CVPR.2015.7298698.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=hzVpZDrW73.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 6700–6709. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00686. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Hudson_GQA_A_New_Dataset_for_Real-World_Visual_Reasoning_and_Compositional_CVPR_2019_paper.html.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: toward spatio-temporal reasoning in visual question answering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 1359–1367. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.149. URL https://doi.org/10.1109/CVPR.2017.149.

Yizhang Jin, Jian Li, Tianjun Gu, Yexin Liu, Bo Zhao, Jinxiang Lai, Zhenye Gan, Yabiao Wang, Chengjie Wang, Xin Tan, and Lizhuang Ma. Efficient multimodal large language models: a survey. *Vis. Intell.*, 3(1), 2025. doi: 10.1007/S44267-025-00099-6. URL https://doi.org/10.1007/s44267-025-00099-6.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vis.*, 123(1):32–73, 2017. doi: 10.1007/S11263-016-0981-7. URL https://doi.org/10.1007/s11263-016-0981-7.

Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(3):3530–3543, 2026. doi: 10.1109/TPAMI.2025.3637265. URL https://doi.org/10.1109/TPAMI.2025.3637265.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural*

Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pp. 292–305. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.20. URL https://doi.org/10.18653/v1/2023.emnlp-main.20.

Yuncheng Li, Yale Song, Liangliang Cao, Joel R. Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A new dataset and benchmark on animated GIF description. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 4641–4650. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.502. URL https://doi.org/10.1109/CVPR.2016.502.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pp. 5971–5984. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.EMNLP-MAIN.342. URL https://doi.org/10.18653/v1/2024.emnlp-main.342.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/6dcf277ea32ce3288914faf369fe6de0-Abstract-Conference.html.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 26286–26296. IEEE, 2024a. doi: 10.1109/CVPR52733.2024.02484. URL https://doi.org/10.1109/CVPR52733.2024.02484.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024b. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model. *CoRR*, abs/2411.10803, 2024c. doi: 10.48550/ARXIV.2411.10803. URL https://doi.org/10.48550/arXiv.2411.10803.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? In Ales Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part VI*, volume 15064 of *Lecture Notes in Computer Science*, pp. 216–233. Springer, 2024d. doi: 10.1007/978-3-031-72658-3_13. URL https://doi.org/10.1007/978-3-031-72658-3_13.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 11966–11976. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01167. URL https://doi.org/10.1109/CVPR52688.2022.01167.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/11332b6b6cf4485b84afadb1352d3a9a-Abstract-Conference.html.

Muhammad Maaz, Hanoona Abdul Rasheed, Salman Khan, and Fahad Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 12585–12602. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.679. URL https://doi.org/10.18653/v1/2024.acl-long.679.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 8024–8035, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 22857–22867. IEEE, 2025. doi: 10.1109/ICCV51701.2025.02122. URL https://doi.org/10.1109/ICCV51701.2025.02122.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 8317–8326. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00851. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Singh_Towards_VQA_Models_That_Can_Read_CVPR_2019_paper.html.

Rinyoichi Takezoe, Yaqian Li, Zi-Hao Bo, Anzhou Hou, Mo Guang, and Kaiwen Long. Learnpruner: Rethinking attention-based token pruning in vision language models. *CoRR*, abs/2604.23950, 2026. doi: 10.48550/ARXIV.2604.23950. URL https://doi.org/10.48550/arXiv.2604.23950.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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, Guanzhou Chen, Zichen Ding, Changyao Tian, Zhenyu Wu, JingJing Xie, Zehao Li, Bowen Yang, Yuchen Duan, Xuehui Wang, Zhi Hou, Haoran Hao, Tianyi Zhang, Songze Li, Xiangyu Zhao, Haodong Duan, Nianchen Deng, Bin Fu, Yinan He, Yi Wang, Conghui He, Botian Shi, Junjun He, Yingtong Xiong, Han Lv, Lijun Wu, Wenqi Shao, Kaipeng Zhang, Huipeng Deng, Biqing Qi, Jiaye Ge, Qipeng Guo, Wenwei Zhang, Songyang Zhang, Maosong Cao, Junyao Lin, Kexian Tang, Jianfei Gao, Haian Huang, Yuzhe Gu, Chengqi Lyu, Huanze Tang, Rui Wang, Haijun Lv, Wanli Ouyang, Limin Wang, Min Dou, Xizhou Zhu, Tong Lu, Dahua Lin, Jifeng Dai, Weijie Su, Bowen Zhou, Kai Chen, Yu Qiao, Wenhai Wang, and Gen Luo. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *CoRR*, abs/2508.18265, 2025. doi: 10.48550/ARXIV.2508.18265. URL https://doi.org/10.48550/arXiv.2508.18265.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pp. 15537–15549. Association for Computational Linguistics, 2025a. doi: 10.18653/V1/2025.FINDINGS-ACL.802. URL https://doi.org/10.18653/v1/2025.findings-acl.802.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 9961–9980. Association for Computational Linguistics, 2025b. doi: 10.18653/V1/2025.EMNLP-MAIN.505. URL https://doi.org/10.18653/v1/2025.emnlp-main.505.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and Chong Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *CoRR*, abs/2412.10302, 2024. doi: 10.48550/ARXIV.2412.10302. URL https://doi.org/10.48550/arXiv.2412.10302.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *CoRR*, abs/2410.17247, 2024. doi: 10.48550/ARXIV.2410.17247. URL https://doi.org/10.48550/arXiv.2410.17247.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In Qiong Liu, Rainer Lienhart, Haohong Wang, Sheng-Wei "Kuan-Ta" Chen, Susanne Boll, Yi-Ping Phoebe Chen, Gerald Friedland, Jia Li, and Shuicheng Yan (eds.), *Proceedings of the 2017 ACM on Multimedia Conference, MM 2017, Mountain View, CA, USA, October 23-27, 2017*,

pp. 1645–1653. ACM, 2017. doi: 10.1145/3123266.3123427. URL https://doi.org/10.1145/3123266.3123427.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. MSR-VTT: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 5288–5296. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.571. URL https://doi.org/10.1109/CVPR.2016.571.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 19792–19802. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.01843. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_VisionZip_Longer_is_Better_but_Not_Necessary_in_Vision_Language_CVPR_2025_paper.html.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 24972–24982. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02325. URL https://openaccess.thecvf.com/content/CVPR2025/html/Ye_ATP-LLaVA_Adaptive_Token_Pruning_for_Large_Vision_Language_Models_CVPR_2025_paper.html.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 9127–9134. AAAI Press, 2019. doi: 10.1609/AAAI.V33I01.33019127. URL https://doi.org/10.1609/aaai.v33i01.33019127.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 20857–20867. IEEE, 2025a. doi: 10.1109/ICCV51701.2025.01939. URL https://doi.org/10.1109/ICCV51701.2025.01939.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *CoRR*, abs/2506.10967, 2025b. doi: 10.48550/ARXIV.2506.10967. URL https://doi.org/10.48550/arXiv.2506.10967.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025c. URL https://proceedings.mlr.press/v267/zhang25s.html.

Jiaying Zhu, Yurui Zhu, Xin Lu, Wenrui Yan, Dong Liu, Kunlin Liu, Xueyang Fu, and Zheng-Jun Zha. Visionselector: End-to-end learnable visual token compression for efficient multimodal llms. *CoRR*, abs/2510.16598, 2025. doi: 10.48550/ARXIV.2510.16598. URL https://doi.org/10.48550/arXiv.2510.16598.

A Appendix

Contents of Appendix

A.1 Implementation Details 18
A.1.1 Training and Model Configuration 18
A.1.2 Spatial Smoothing of the Relevance Map 18
A.1.3 Selection Operator 19
A.1.4 Efficiency Measurement 20
A.2 Extended Results 21
A.2.1 Dynamic-Resolution Backbone 21
A.2.2 Larger Backbone 22
A.3 Extended Ablations 23
A.3.1 Instruction Relevance Channel 23
A.3.2 Learned Instruction-Relevance Channel 24
A.3.3 Score Fusion 25
A.3.4 Pruning Layer 26
A.4 Qualitative Analysis 26
A.5 Limitations 26
A.6 Evaluation Benchmarks 27

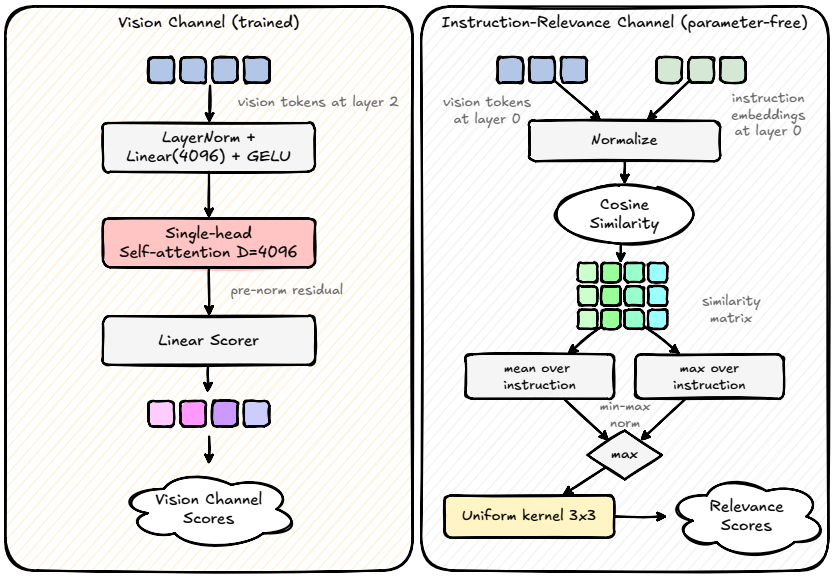


Figure 5: Overview of the vision and instruction-relevance channels.

A.1 Implementation Details

A.1.1 Training and Model Configuration

The selector is inserted at decoder layer ℓ=2 for all backbones and fuses the two channels of Figure 5 with α=0.5. For fixed-length backbones, we smooth the resulting relevance map using a uniform 3×3 kernel. Extended ablations can be found in Appendix A.3.

DUETTO is trained for one epoch on 10% of the LLaVA-1.5 665K instruction-tuning dataset (Liu et al., 2024a) on a single NVIDIA H200 140-GB GPU using PyTorch (Paszke et al., 2019), with a global batch size of 128, and a learning rate of 1×10^{-5}. The vision channel attention dimension is 4096, with a single attention head, amounting to 84M parameters, approximately 1% of the full model parameters. We use a 0.03 learning-rate warm-up followed by cosine decay, with AdamW (Loshchilov & Hutter, 2019) as the optimizer and FlashAttention-2 (Dao, 2024) for attention computation.

A.1.2 Spatial Smoothing of the Relevance Map

As described in §3.2.2, we smooth the instruction relevance map by placing each vision token at its corresponding position in the two-dimensional grid, and convolving the affinity from Equation 5 over the grid. Let $a_{u,v}$ denote the affinity of the token at row u and column v in an $H \times W$ grid. We use a uniform 3×3 kernel giving,

$$r_{u,v} = \frac{1}{9} \sum_{\delta u=-1}^{1} \sum_{\delta v=-1}^{1} a_{\pi_H(u+\delta u),\, \pi_W(v+\delta v)}, \qquad \pi_H(x) = \min\big(\max(x, 1),\, H\big), \tag{8}$$

where π_W clamps the column index in the same way. Thus, when a neighborhood extends beyond a grid boundary, the nearest border token is used, so border tokens are smoothed in the same way as interior tokens. For the Gaussian variant in Table 9, we replace the uniform weights with $\kappa_{\delta u,\delta v} \propto \exp\big(-(\delta u^2 + \delta v^2)/2\big)$ and normalize the weights to sum to one.

Figure 6 shows the operator at work, as the per-token cosine responds independently and spreads its mass over isolated tokens, which the convolution collects into a few coherent regions.

LLaVA-1.5 and Mini-Gemini hand the decoder a single 24×24 grid of 576 tokens, which is convolved once. Video-LLaVA encodes each of its eight frames into a class token followed by a 16×16 patch grid, so every frame is convolved independently, and the class tokens pass through untouched,

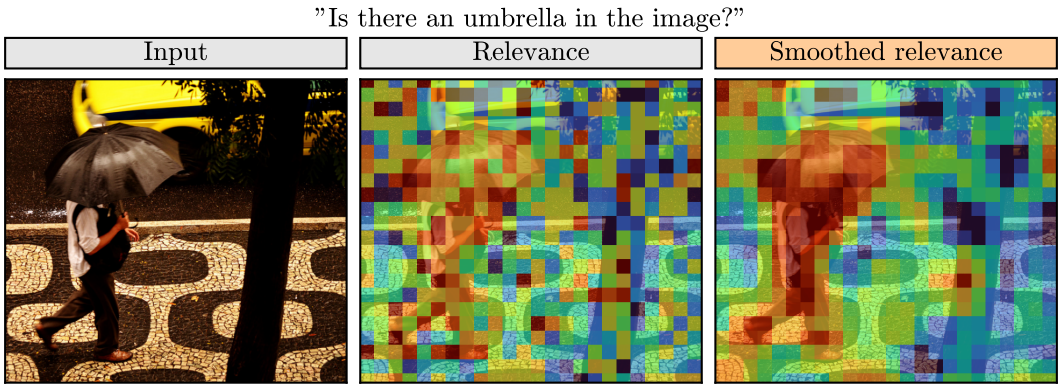


Figure 6: Instruction relevance before and after the smoothing of Equation 8.

which confines the smoothing within a frame and leaves the temporal axis alone. LLaVA-NeXT provides two grids, a thumbnail that encodes the whole image and a higher-resolution grid composed of the sub-images, which we smooth separately: the 24×24 thumbnail as in LLaVA-1.5, and the assembled sub-images together, skipping the newline tokens.

A.1.3 Selection Operator

The operator of §3.2.3 adapts the differentiable top-k operator of VisionSelector (Zhu et al., 2025). The forward threshold τ of Eq. equation 7 is the root of

$$\Phi(\mathbf{s},\tau) \triangleq \sum_{i=1}^{N} \sigma(s_i+\tau) - k. \tag{9}$$

Since σ is continuous and strictly increasing, so is Φ in τ, with $\Phi \to -k$ as $\tau \to -\infty$ and $\Phi \to N-k$ as $\tau \to +\infty$, so for any budget $0 < k < N$ the root exists and is unique. The procedure is summarized in Algorithm 1.

Forward pass. During training, the mask enters as a row scaling of the vision tokens, while at inference the rows outside the retained set are removed,

$$\tilde{\mathbf{H}}_v^{(\ell)} = \mathrm{diag}(\mathbf{m})\,\mathbf{H}_v^{(\ell)} \quad \text{(training)}, \qquad \tilde{\mathbf{H}}_v^{(\ell)} = \mathcal{S} \quad \text{(inference)}, \tag{10}$$

and in both cases the layers from ℓ onward attend only over $\mathcal{S} = \text{top-}k(\mathbf{s})$, so the budget is exact during training as well. The mask multiplies the hidden states, so no straight-through estimator is needed. Saturation is favored by the objective, since attenuating a token the answer depends on raises the loss.

Backward pass. The threshold moves with the scores through the budget constraint, and differentiating that constraint gives the gradient of the mask with respect to the scores (Zhu et al., 2025). With $g_i = \partial\mathcal{L}/\partial m_i$ the gradient arriving at the mask and $v_i = m_i(1-m_i)$,

$$\frac{\partial\mathcal{L}}{\partial s_i} = v_i\big(g_i - \bar{g}\big), \qquad \bar{g} = \frac{\sum_j v_j\, g_j}{\sum_j v_j}. \tag{11}$$

Tokens outside $\mathcal{S}$ are masked out of the attention that follows, so $g_i = 0$ and $-v_i\bar{g}$ is the only signal their scores receive, largest for the tokens nearest the threshold.

We measure the mask values' distribution on 150 samples, evaluated on LLaVA-1.5 at the 64-token budget, reading m_i as returned by Algorithm 1, both at initialization and after training, illustrated in Figure 7. At initialization, both groups sit in an overlapping band in the lower end of the values, where tokens inside $\mathcal{S}$ range from 0.18 to 0.83 and outside from 0.00 to 0.29, giving a largely undecided score with no mass near either extreme. After training, outside $\mathcal{S}$ mass is heavily concentrated near 0, and no observed value exceeds 0.56. Inside $\mathcal{S}$, the mask is not uniformly saturated, as while

Algorithm 1 Budget-exact soft mask via sigmoid bisection

Require: scores $\mathbf{s} \in \mathbb{R}^N$, budget k, iterations T=64, margin c=10

$\tau_{\text{lo}} \leftarrow -\max_i s_i - c$; $\tau_{\text{hi}} \leftarrow -\min_i s_i + c$
for $t = 1$ **to** T **do**
  $\tau_{\text{mid}} \leftarrow \frac{1}{2}(\tau_{\text{lo}} + \tau_{\text{hi}})$
  if $\sum_i \sigma(s_i + \tau_{\text{mid}}) < k$ **then**
    $\tau_{\text{lo}} \leftarrow \tau_{\text{mid}}$
  else
    $\tau_{\text{hi}} \leftarrow \tau_{\text{mid}}$
  end if
end for
$\tau \leftarrow \frac{1}{2}(\tau_{\text{lo}} + \tau_{\text{hi}})$
return $m_i = \sigma(s_i + \tau)$ for all i

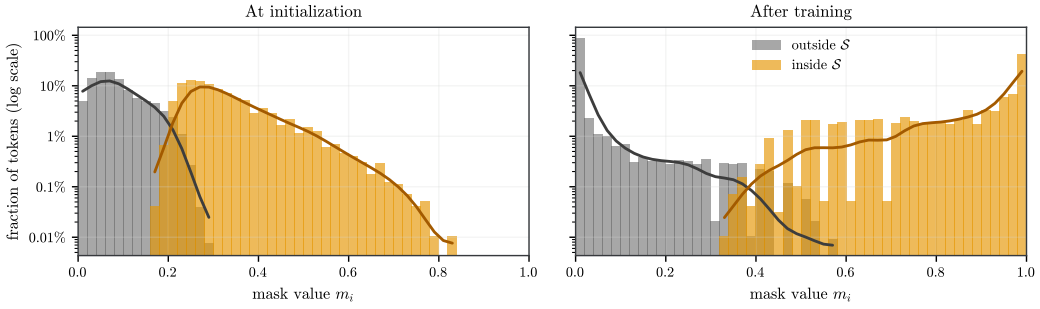


Figure 7: Distribution of the mask values m_i at initialization and after training, over 150 samples on LLaVA-1.5 at 64 tokens. The fraction of tokens is presented in log scale.

the mode sits near 1, 37.7% of kept tokens fall below 0.9 and the minimum observed value is 0.33, showing that the objective saturates the tokens the answer most depends on while leaving a fraction of the remaining kept tokens only partially attenuated during training, since the loss only penalizes attenuating a token whose removal would change the answer.

A.1.4 Efficiency Measurement

All methods are benchmarked using a unified evaluation harness on a single NVIDIA A100-40GB GPU, at a 160-token budget, with a fixed generation length of 24 tokens to capture end-to-end inference latency under a consistent decoding workload. **Prefill** denotes the latency to the first generated token, **Total** denotes end-to-end inference latency, **Select.** denotes the standalone cost of the compression procedure, with its corresponding fraction of prefill latency reported alongside it, **Mem** denotes peak allocated memory, and **Acc.** denotes the relative accuracy reported in Table 2.

FLOPs follow the estimate FastV (Chen et al., 2024b) introduces and later adopted by other methods. For one transformer layer, with n the token count, d the hidden state size, and m the FFN intermediate size, the total is estimated as $4nd^2 + 2n^2d + 2ndm$. For a method that prunes the tokens from n to $\hat{n}$ after layer K of T layers, we report the staged total

$$K \times \left(4nd^2 + 2n^2d + 2ndm\right) + (T - K) \times \left(4\hat{n}d^2 + 2\hat{n}^2d + 2\hat{n}dm\right), \tag{12}$$

with d=4096, m=11008 and T=32. DUETTO and Dynamic-LLaVA prune at K=2 and therefore incur two full-length layers, the methods acting before the language model correspond to K=0, and the full-token model has $\hat{n}$=n. Since only vision tokens are pruned, $\hat{n}$ counts the system and instruction tokens of the benchmarked prompt together with the retained vision tokens.

Table 7: Performance on Qwen2.5-VL-7B. The best result at each budget is **bolded**.

Method	GQA	SQA^I	VQA^T	POPE	MME	MMB	RelAcc.
Upper Bound (100%)							
Qwen2.5-VL-7B	60.9	88.3	79.0	87.8	2323	86.8	100.0%
Token retention ratio 33.3% (↓ 66.7%)							
VisionZip‡	59.1	85.2	77.5	**86.6**	2125	83.6	96.3%
Dyn-LLaVA	57.9	83.6	65.4	84.0	1980	82.1	91.3%
DUETTO	**60.3**	**85.9**	**81.8**	85.9	**2127**	**84.1**	**97.7%**
Token retention ratio 22.2% (↓77.8%)							
VisionZip‡	57.6	**84.0**	74.1	**85.3**	2006	82.7	93.7%
Dyn-LLaVA	56.0	80.6	64.0	84.6	1938	83.4	90.0%
DUETTO	**59.3**	83.8	**78.5**	84.1	**2071**	**84.1**	**95.6%**
Token retention ratio 11.1% (↓ 88.9%)							
VisionZip‡	56.0	83.0	**73.4**	**84.1**	**1931**	79.7	91.6%
Dyn-LLaVA	55.3	78.6	60.2	81.6	1726	79.0	85.7%
DUETTO	**56.9**	**83.4**	72.9	83.4	1903	**81.8**	**91.9%**

Table 8: Performance on LLaVA-1.5-13B. The best result at each budget is **bolded**.

Method	GQA	SQA^I	VQA^T	POPE	MME	MMB	RelAcc.
Upper Bound, 576 Tokens (100%)							
LLaVA-1.5-13B	63.3	72.5	61.2	86.0	1827	67.7	100.0%
Retain 192 Tokens (∼33%, ↓ 67%)							
VisionZip‡	61.5	**73.3**	59.7	85.4	1825	**67.3**	99.1%
Dyn-LLaVA	61.8	72.6	**60.5**	82.8	1821	66.8	98.6%
DUETTO	**62.4**	72.4	60.2	**86.0**	**1827**	**67.3**	**99.4%**
Retain 128 Tokens (22.2%, ↓ 77.8%)							
VisionZip‡	60.0	**73.7**	59.0	84.2	1788	66.5	97.8%
Dyn-LLaVA	59.9	72.5	59.6	79.7	1791	66.6	96.9%
DUETTO	**62.0**	72.6	**60.1**	**85.7**	**1798**	**66.7**	**98.8%**
Retain 64 Tokens (11.1%, ↓ 88.9%)							
VisionZip‡	58.2	73.0	57.8	81.4	1722	64.9	95.3%
Dyn-LLaVA	55.3	71.8	56.8	68.8	1652	64.2	90.7%
DUETTO	**59.5**	**73.2**	**59.5**	**84.6**	**1774**	**65.1**	**97.3%**

A.2 Extended Results

Beyond the results of the main text, we evaluate the selector on a backbone whose visual sequence length varies with the input and on a larger decoder, holding the recipe of §A.1.1 fixed in both.

A.2.1 Dynamic-Resolution Backbone

To test whether the selector transfers to a backbone that derives the vision sequence length from input resolution, we apply it to Qwen2.5-VL-7B (Bai et al., 2025), which then merges the vision tokens fourfold inside the vision tower, so that every token the downstream LLM receives already

Table 9: Instruction relevance channel at the 128- and 64-token budgets, Δ against the **DUETTO** on each budget, which uses the max of the mean and max reductions with a uniform kernel, and adopts an uniform 3×3 kernel for smoothing.

Variant	GQA	SQA^{I}	VQA^{T}	POPE	MME	MMB	Acc.	RelAcc.	Δ
Retain 128 Tokens (22.2%, ↓ 77.8%)									
DUETTO	60.9	67.9	57.6	84.9	1844	63.4	66.8	**98.6%**	—
reduction = mean	60.0	68.8	57.9	83.6	1825	62.6	66.3	98.0%	−0.6
reduction = max	60.7	67.9	57.4	85.1	1836	62.8	66.6	98.3%	−0.3
gaussian 3×3 kernel	60.2	68.7	57.6	84.8	1852	62.8	66.7	98.5%	−0.1
Retain 64 Tokens (11.1%, ↓ 88.9%)									
DUETTO	58.8	68.8	56.7	84.6	1817	62.2	66.0	**97.4%**	—
reduction = mean	58.6	68.3	57.1	82.9	1762	62.7	65.4	96.6%	−0.8
reduction = max	59.3	68.4	57.0	84.6	1743	62.5	65.7	96.9%	−0.5
gaussian 3×3 kernel	59.1	68.2	56.6	84.9	1757	62.2	65.6	96.8%	−0.6

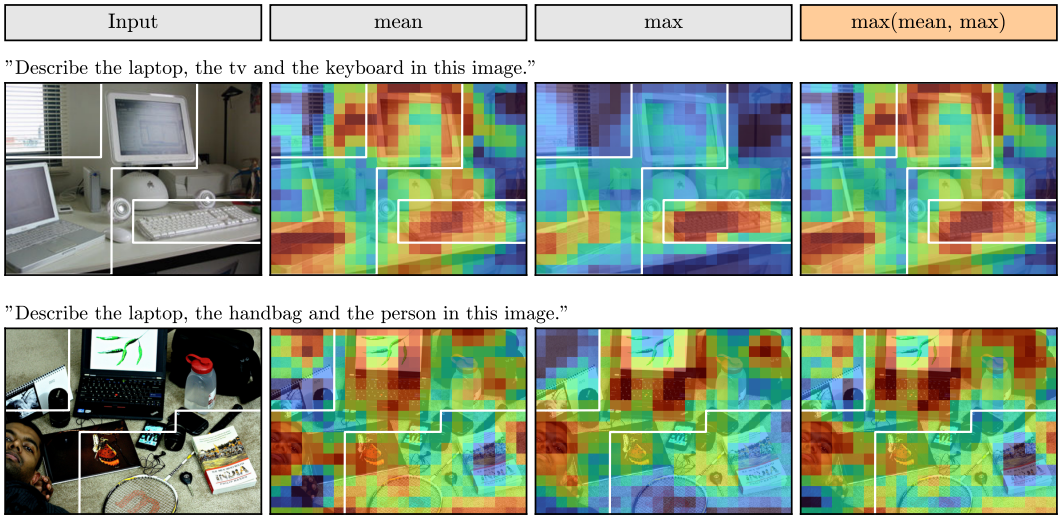


Figure 8: The three reductions of Equation 5, each smoothed with the same kernel, with the named objects outlined.

summarizes a 2×2 block of the patch grid. Qwen2.5-VL's instruction-tuning mixture is not public, so we train on the same 10% slice of LLaVA-1.5 665K used throughout.

As Table 7 shows, DUETTO retains **97.7%** of the full-token accuracy at a 33% keep rate, **95.6%** at 22% and **91.9%** at 11%, leading the fine-tuned VisionZip‡ at every budget, by 1.4, 1.9 and 0.3 points, with the thinnest margin where the merged grid leaves fewest tokens to choose between.

A.2.2 Larger Backbone

To separate what the selector learns from the capacity of the decoder it is trained inside, we apply it to LLaVA-1.5-13B, which retains the CLIP (Radford et al., 2021) encoder and the 24×24 patch grid of the 7B model and scales the decoder alone, so that the budgets, the compression rates and the training recipe of §A.1.1 all carry over unchanged and the backbone scale is the only variable.

As Table 8 shows, the ordering of the LLaVA-1.5-7B results holds at the larger scale, where DUETTO retains **99.4%** of the full-token accuracy while keeping a third of the vision tokens, **98.8%** while keeping 22% and **97.3%** while keeping 11%, leading the fine-tuned VisionZip‡ by 0.3, 1.0 and 2.0 points and Dynamic-LLaVA by 0.8, 1.9 and 6.6. The 7B backbone reaches 99.1%, 98.5%

Table 10: Instruction-relevance channel designs at the 64-token budget.

Channel	GQA	SQA^{I}	VQA^{T}	POPE	MME	VQA^{v2}	MMB	RelAcc.
DUETTO	**58.8**	**68.8**	**56.7**	**84.6**	**1817**	**75.4**	**62.2**	**97.2%**
learned cross-attention	58.3	68.6	55.5	81.8	1786	74.8	**62.1**	95.9%
vision-only	57.3	**68.8**	56.5	82.5	1761	74.3	62.0	95.8%

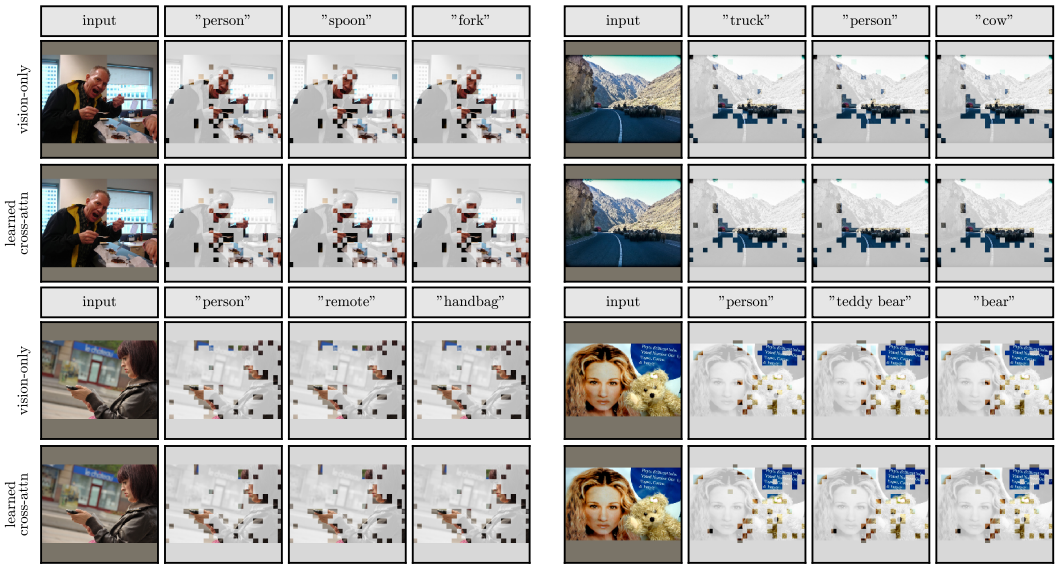


Figure 9: Retained tokens for three questions about the same image: the input at the left of each row, the vision-only selector above, and the learned cross-attention channel below. Similarly to the vision-only variant, the learned instruction-relevance channel is instruction-agnostic.

and 97.4% over the same six benchmarks, so retention sits within roughly three tenths of a point at either scale, which indicates that the selection does not lean on decoder capacity to recover from the selection.

A.3 Extended Ablations

We ablate each remaining design choice one at a time on LLaVA-1.5-7B at the 128 and 64 token budgets. Each table varies a single axis within each budget against DUETTO that heads the panel, and Δ denotes the relative accuracy difference to DUETTO.

A.3.1 Instruction Relevance Channel

Table 9 shows that reducing the patch-question similarity by taking the per-patch maximum over instruction tokens outperforms averaging, which dilutes a sharp match across the many uninformative prompt tokens. Taking the maximum of both reductions gives a further improvement, so we adopt this setting. At 128 tokens, Gaussian and uniform smoothing perform comparably with only a 0.1 point difference, while at 64 tokens, the uniform kernel improves relative accuracy by 0.6 over the Gaussian alternative, motivating our choice of the uniform kernel.

Figure 8 shows the three reductions of Equation 5 and how they interact. In the first row, the maximum concentrates on the laptop and the keyboard, leaving the television without a single retained token, which the mean recovers by reaching all three. In the second row, the mean spreads across the frame and reaches the laptop and the person but none of the handbag, and the maximum places every one of its 64 tokens inside the three named objects, and the per-token maximum of the two inherits

Table 11: Fusion weight α at the 128- and 64-token budgets, Δ against **DUETTO** on each budget, which is the adopted α=0.5.

Variant	GQA	SQA^{I}	VQA^{T}	POPE	MME	MMB	Acc.	RelAcc.	Δ
Retain 128 Tokens (22.2%, ↓ 77.8%)									
DUETTO	60.9	67.9	57.6	84.9	1844	63.4	66.8	**98.6%**	—
$\alpha = 0.25$	60.6	68.3	57.3	84.9	1822	63.9	66.7	98.5%	−0.1
$\alpha = 0.75$	60.2	68.0	57.6	84.5	1861	63.2	66.7	98.5%	−0.1
$\alpha = 1.0$	60.2	68.4	57.4	84.6	1835	63.1	66.5	98.2%	−0.4
$\alpha = 2.0$	60.3	68.1	56.9	84.2	1803	63.4	66.2	97.8%	−0.8
Retain 64 Tokens (11.1%, ↓ 88.9%)									
DUETTO	58.8	68.8	56.7	84.6	1817	62.2	66.0	**97.4%**	—
$\alpha = 0.25$	58.6	68.8	57.0	83.0	1769	62.3	65.5	96.7%	−0.7
$\alpha = 0.75$	59.0	68.1	56.7	84.5	1748	62.2	65.5	96.6%	−0.8
$\alpha = 1.0$	58.6	68.2	56.6	83.2	1751	62.8	65.3	96.4%	−1.0
$\alpha = 2.0$	57.9	68.1	55.3	82.9	1718	61.7	64.5	95.2%	−2.2

Table 12: Selector (fusion arithmetic and self-attention width) at the 128- and 64-token budgets, Δ against **DUETTO** on each budget, which uses min-max fusion at width 4096.

Variant	GQA	SQA^{I}	VQA^{T}	POPE	MME	MMB	Acc.	RelAcc.	Δ
Retain 128 Tokens (22.2%, ↓ 77.8%)									
DUETTO	60.9	67.9	57.6	84.9	1844	63.4	66.8	**98.6%**	—
fusion = raw-add	60.7	68.0	57.7	84.5	1817	62.5	66.4	98.0%	−0.6
fusion = z-score	60.2	68.5	57.9	84.4	1842	62.8	66.6	98.4%	−0.2
attn dim = 512	60.4	68.1	57.0	84.9	1806	63.1	66.3	97.9%	−0.7
attn dim = 2048	60.4	68.1	57.7	84.2	1825	63.0	66.4	98.1%	−0.5
Retain 64 Tokens (11.1%, ↓ 88.9%)									
DUETTO	58.8	68.8	56.7	84.6	1817	62.2	66.0	**97.4%**	—
fusion = raw-add	58.6	68.5	56.8	83.3	1766	62.1	65.4	96.5%	−0.9
fusion = z-score	58.7	68.3	56.7	84.0	1755	62.2	65.4	96.6%	−0.8
attn dim = 512	58.1	67.8	55.4	83.1	1721	61.6	64.6	95.3%	−2.1
attn dim = 2048	58.2	68.2	56.6	83.3	1766	62.2	65.2	96.3%	−1.1

that focus. Neither reduction alone fully covers both cases, and taking their per-token maximum follows whichever one is the stronger at each token, as supported by the results in Table 9.

A.3.2 Learned Instruction-Relevance Channel

Since the vision channel is trained end to end, it is natural to ask whether the instruction-relevance channel should be learned as well, so we replace the parameter-free cosine with a learned cross-attention channel, in which the instruction tokens supply the keys and values that attend into the vision span inside the selector before the score head reads it out, and we train this variant at the 64-token budget under the identical recipe.

Table 10 shows that the learned channel reaches 95.9% relative accuracy, standing 0.1 points above the vision-only selector that carries no instruction channel at all, whereas the parameter-free cosine reaches **97.2%**.

To examine how the learned channel responds to the instruction, Figure 9 fixes the image and varies the object asked about across columns. If the learned channel captured instruction relevance, its selected tokens should change as the queried object changes. Instead, similarly to the vision-only variant, the learned channel selections remain nearly unchanged, providing little additional instruction-dependent information.

Table 13: Pruning layer ℓ at the 128- and 64-token budgets, Δ against **DUETTO** on each budget, which is the adopted ℓ=2.

Variant	GQA	SQA^{I}	VQA^{T}	POPE	MME	MMB	Acc.	RelAcc.	Δ
Retain 128 Tokens (22.2%, ↓ 77.8%)									
DUETTO	60.9	67.9	57.6	84.9	1844	63.4	66.8	**98.6%**	—
$\ell = 1$	60.5	68.1	57.1	84.9	1822	62.7	66.4	98.0%	−0.6
$\ell = 4$	60.7	68.2	57.4	84.6	1866	63.1	66.8	98.6%	+0.0
$\ell = 8$	60.9	67.8	57.6	85.3	1843	63.0	66.7	98.5%	−0.1
$\ell = 16$	61.7	68.7	57.4	85.0	1843	63.3	67.0	98.9%	+0.3
Retain 64 Tokens (11.1%, ↓ 88.9%)									
DUETTO	58.8	68.8	56.7	84.6	1817	62.2	66.0	**97.4%**	—
$\ell = 1$	58.6	68.5	56.7	83.0	1734	62.2	65.2	96.2%	−1.2
$\ell = 4$	58.6	68.9	56.5	84.4	1753	61.8	65.5	96.6%	−0.8
$\ell = 8$	59.4	68.9	56.9	84.6	1790	62.8	66.1	97.5%	+0.1
$\ell = 16$	61.6	68.7	56.9	85.1	1852	63.4	67.0	98.9%	+1.5

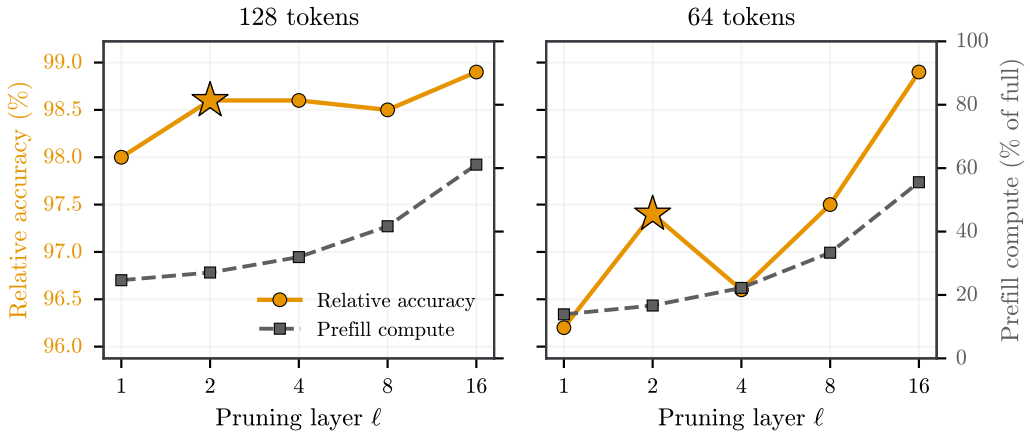


Figure 10: Relative accuracy and prefill compute against the pruning layer ℓ at both budgets, the adopted ℓ=2 starred.

Since the language-modeling loss is the only supervision reaching the learned channel, and the vision channel already provides a useful signal for this objective, there is therefore no direct incentive for the learned channel to specialize in instruction relevance. In contrast, the parameter-free channel computes its score directly from the embedding-space similarity between vision tokens and the instruction, independently of the learned channel. We therefore keep the instruction channel parameter-free.

A.3.3 Score Fusion

Fusion weight. Sweeping the fusion weight α, which sets how strongly the relevance signal steers selection relative to the learned channel, leaves relative accuracy on a broad plateau for the smaller weights (98.2-98.6% for $\alpha \in \{0.25, 0.5, 0.75, 1.0\}$) and only degrades once the text term is allowed to dominate the score (97.8% at α=2.0), where it begins to override the learned channel and the redundancy structure it has captured is lost, seen in Table 11. At 64 tokens, the same sweep is sharper, with the adopted α=0.5 standing 0.7 and 0.8 points above its neighbors on either side and 2.2 above α=2.0, placing the optimum where the text term supports the learned channel instead of overriding it.

Score normalization. Bringing the unbounded vision logit and the bounded relevance score onto a common range before adding them, whether by min-max or by standardization, beats the raw addition of mismatched scales, and the min-max normalization we adopt leads to both alternatives, illustrated in Table 12. The three variants sit within 0.6 points at 128 tokens and spread to 0.9 at 64, so the arithmetic of the fusion is a minor interface at a loose budget and a consequential one once the score decides which few tokens survive. Accuracy also rises with the working width of the vision channel's self-attention block in the same table, which is the width of the hidden states the channel reads.

A.3.4 Pruning Layer

Table 13 reports the performance obtained when applying the selector at different decoder layers. The results show that the accuracy gains in the early layers are not consistent across the 64- and 128-token budgets, while applying the selector at deeper layers can improve performance, particularly at the 64-token budget. At layer 16, for example, accuracy improves by 1.5 points relative to the adopted layer $\ell{=}2$.

As illustrated in Figure 10, accuracy at the 128-token budget reaches a plateau from layer 2 onward. At the 64-token budget, accuracy varies more substantially across layers, as layer 2 improves over its neighboring layers 1 and 4, matches the accuracy obtained at layer 8, and increases further at layer 16. However, this accuracy improvement comes with a substantial efficiency cost, as when the selector is applied later, more decoder layers process the full set of vision tokens, increasing the fraction of prefill computation from approximately 20% at layer 2 to nearly 60% at layer 16. Based on this accuracy-efficiency trade-off, we adopt $\ell{=}2$ as the selector layer. This choice is also consistent with the findings of Chen et al. (2024b), which identify layer 2 as an effective point for vision token compression within the decoder.

A.4 Qualitative Analysis

Figures 11 and 12 show the tokens each of the three learned selectors retains at the 64-token budget, on twenty examples drawn from GQA (Hudson & Manning, 2019), POPE (Li et al., 2023) and TextVQA (Singh et al., 2019) and using the checkpoints behind the 64-token block of Table 1.

The last row of each figure is a failure case. Asked which color the shirt made of cloth is, the selector keeps tokens on a shirt other than the intended one and answers from it, and asked which food is round, it retains tokens across several foods on the plate and answers with the apple in place of the sandwich. In both, the budget lands on evidence that is consistent with the question read loosely, which is the price of steering selection with a relevance signal that carries no notion of which candidate the question intends.

A.5 Limitations

The fusion of §3.2.3 combines the two channels through α, which acts as an exchange rate between the vision score and the relevance score, and we hold it at $\alpha{=}0.5$ for every backbone and every budget we report. A single exchange rate cannot tell an instruction that names one object apart from one that is broad or uninformative, so the relevance term carries the same weight in both cases, and the selection follows the instruction even where the instruction has little to say. The failure cases of §A.4 illustrate this effect, as an instruction naming an attribute that several objects in the scene share steers the budget onto the wrong candidate while the answer it supports remains consistent with the question read loosely.

DUETTO holds the highest relative accuracy at every budget we report, but consistently underperforms on POPE (Li et al., 2023), falling behind CDPruner (Zhang et al., 2025b) at every budget on both LLaVA backbones and DivPrune (Alvar et al., 2025) on LLaVA-1.5. POPE asks whether a named object is absent, which is settled by establishing that nothing in the scene supports it, so no region holds the answer, and the relevance term steers the budget toward whatever most resembles the object the question names, spending it on the background features the question exists to rule out. The diversity-based methods spread their tokens across the scene instead, retaining the broad coverage that verifying an absence needs.

A.6 Evaluation Benchmarks

GQA (Hudson & Manning, 2019). A large-scale visual question answering benchmark built on scene graphs from the Visual Genome dataset (Krishna et al., 2017), designed to evaluate compositional reasoning and visual understanding.

ScienceQA (Lu et al., 2022). A multimodal multiple-choice question answering benchmark spanning diverse scientific domains, including natural science, language science, and social science. A subset of questions is accompanied by an image, and we evaluate this image-containing subset separately, denoted SQA^{I}.

TextVQA (Singh et al., 2019). A benchmark designed to evaluate a model's ability to recognize and reason about text embedded in images, covering diverse scenarios such as signs, billboards, and product packaging. Following the LLaVA-1.5 (Liu et al., 2024a) protocol, the prompt also carries reference optical character recognition (OCR) tokens to assist the model in interpreting the textual content. In the experimental results, this benchmark is denoted VQA^{T}.

POPE (Li et al., 2023). An object hallucination benchmark based on COCO (Lin et al., 2014) images, where the model is asked to determine whether a specified object is present in the image. The benchmark measures the degree of object hallucination, with results reported as the mean F1-score across the adversarial, random, and popular splits.

MME (Fu et al., 2025). A comprehensive benchmark designed to evaluate both perception and cognition across 14 subtasks. The perception tasks cover object existence, counting, position, and color, the identification of specific posters, celebrities, scenes, landmarks, and artworks, and OCR. The cognition tasks assess common-sense reasoning, numerical calculation, text translation, and code reasoning. We report the summed perception and cognition score.

VQAv2 (Goyal et al., 2017). A large-scale benchmark comprising 204,721 images from COCO (Lin et al., 2014) and 1.1M open-ended questions, designed to evaluate open-ended visual question answering and visual-language understanding. In the experimental results, we denote this benchmark VQA^{v2}.

MMBench Liu et al. (2024d). A multiple-choice benchmark covering a broad range of vision-language capabilities, organized hierarchically into increasingly fine-grained tasks. MMBench introduces a CircularEval strategy, which asks each question once per rotation of its answer options and counts it correct only when the model answers correctly under all of them.

MSVD-QA (Xu et al., 2017). A video question answering benchmark constructed from the Microsoft Research Video Description Corpus (Chen & Dolan, 2011), comprising 1,970 short video clips and 50,505 question-answer pairs.

MSRVTT-QA (Xu et al., 2017). A video question answering benchmark based on the Microsoft Research Video-to-Text dataset (Xu et al., 2016). Compared with MSVD-QA, MSRVTT-QA features longer videos containing more complex and diverse activities, requiring models to capture and reason over temporal information.

TGIF-QA (Jang et al., 2017). A video question answering benchmark based on animated GIFs from the Tumblr GIF dataset (Li et al., 2016), comprising four tasks, three of which require spatio-temporal reasoning, alongside a frame-level task posed as in image question answering.

ActivityNet-QA (Yu et al., 2019). A large-scale video understanding benchmark comprising 58,000 question-answer pairs over 5,800 complex, untrimmed web videos from ActivityNet (Heilbron et al., 2015).

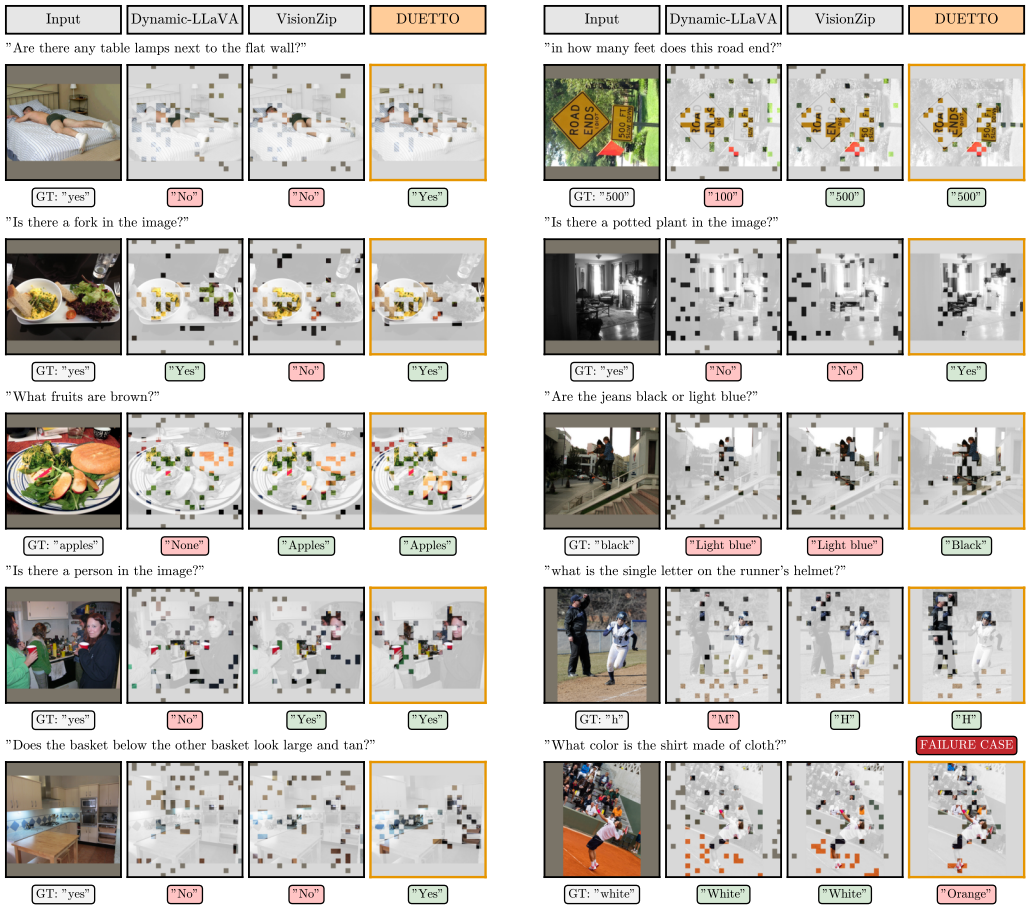


Figure 11: Comparison between models at the 64-token budget. DUETTO retains both visually salient and instruction-relevant tokens.

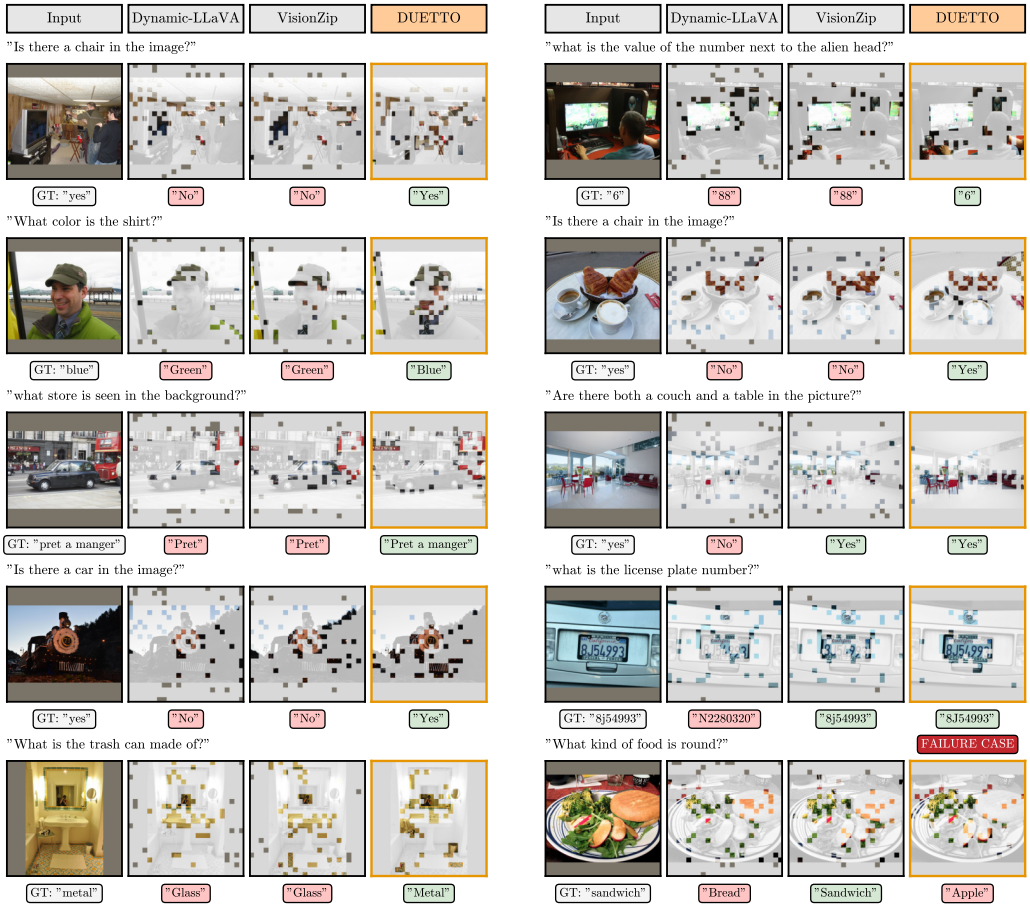


Figure 12: Comparison between models at the 64-token budget, continued.