Glance, then Select: Evidence-Tempered Visual Token Selection for Extremely Compressed Multimodal LLMs

Anonymous CVPR submission

Abstract

At a 5% visual-token budget a multimodal LLM sees about one token in twenty, and two costs remain: the vision encoder still processes every patch of a large image, and the few surviving tokens must still contain the answer. We present Glance-ETD, a two-stage pipeline that decides where to look *and* which tokens to keep. *A glance encoder encodes the whole image at a quarter of the patches, picks 25% of the coarse cells by max–min diversity and re-encodes only those at native resolution, producing a native-size token grid at a sixth of the encoder FLOPs. **E**vidence-**T**empered **D**iversity (ETD) then selects the final tokens by multiplying the max–min diversity gain of each token by its relative evidence $(Np_i)^{\beta t}$, where p and the text confidence t come from a 1.9M-parameter scorer trained on box annotations with the MLLM frozen. With the MLLM frozen, ETD raises retention at strict 5% from 75.8% (DivPrune) to 84.2%. On Qwen3.5-4B trained with on-policy self-distillation, the same training recipe with DivPrune retains 88.8%; ETD raises it to 91.7%, and the glance encoder then trades 1.4 points for a 2.9× shorter time-to-first-token on 4K images (483 to 65 prefill TFLOPs), giving 90.3±0.8%. The released LT-OPD checkpoint retains 85.4% at the same budget. On held-out document benchmarks ETD raises DocVQA from 52.5 to 74.5.*

1. Introduction and Scope

Multimodal LLMs spend most of their prefill on visual tokens. A 4K image produces tens of thousands of patches, and even a 1K image over a thousand merged tokens, for a question usually answered in a few words. Visual token reduction removes most of these tokens before the language model, and LT-OPD [9] shows that a student distilled on its own rollouts from a full-token copy can work with 5% of them, rather than the full visual sequence.

Two problems remain at this budget (Fig. 2). The first is cost. Pruning after the vision encoder shortens the LLM prefill, but the encoder still attends over every patch: on HR-Bench 4K it takes 2.6 s of a 3.9 s time to first token, so removing 95% of the tokens gives only a 1.20× speed-up

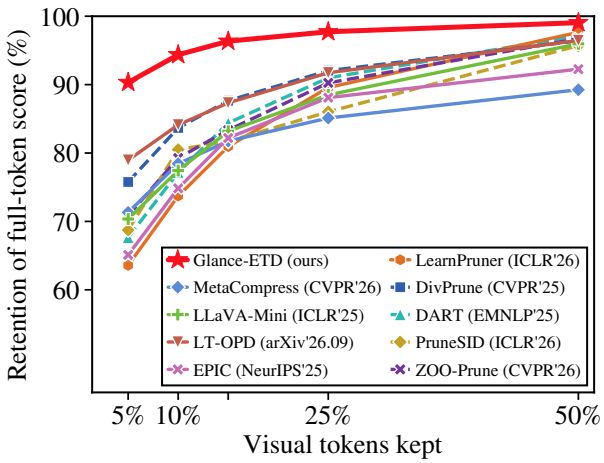


Figure 1. Retention across visual-token budgets on Qwen3.5-4B: the mean over nine benchmarks of each score divided by our non-thinking full-token score, with the same number of tokens per image. Solid lines are trained models; dashed lines are training-free and run on the released model, as in their papers. At 5% of the tokens, Glance-ETD keeps 90.3% (the mean of three training seeds; the other budgets use one seed), 14.5 points above DivPrune.

(Fig. 2a). The second is selection: with one token in twenty, which tokens survive decides whether the answer is still in the input. Diversity selectors such as DivPrune [1] cover the scene but rarely keep the few tokens of a phone number or a street sign; on TextVQA, DivPrune keeps 10.4% of the answer tokens and loses the answer entirely for one question in five (Fig. 2b,c). Relevance alone does not solve this: question–token similarity keeps 2.1% of the answer tokens, and the top tokens of a learned evidence map find the text but lose the coverage that existence and counting questions need (GQA 54.4→51.2, Tab. 3, same budget).

Training the MLLM at the budget does not remove the selection problem. With the self-distillation recipe of LT-OPD and a 10× encoder learning rate, DivPrune retains 88.8% of the full-token score, and the remaining loss is concentrated on reading: TextVQA falls from 80.5 to 70.4 and OCRBench from 848 to 489 (Tab. 1). With the same recipe and the same K, changing only the selector raises them to 76.0 and 579, so on these questions the selector, not the

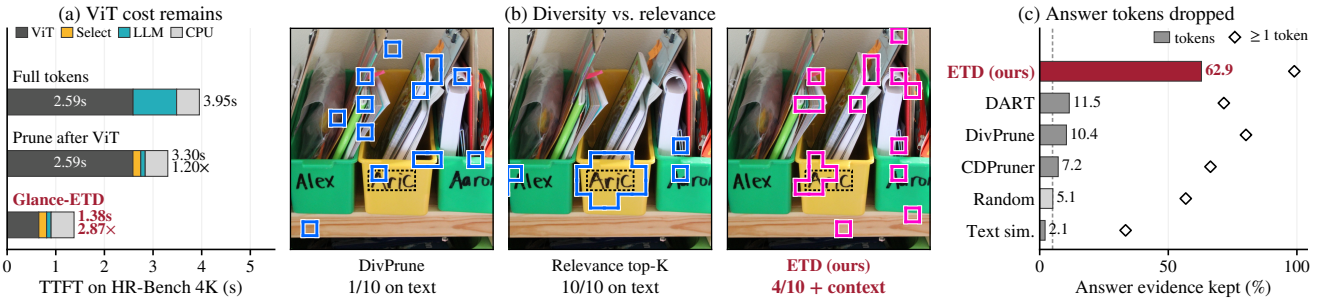


Figure 2. Two problems at a strict 5% visual-token budget on Qwen3.5-4B. (a) Time to first token on HR-Bench 4K for the full-token model, pruning after the ViT, and Glance-ETD (Tab. S13). Each bar is split into vision encoding, selection, LLM prefill and CPU preprocessing. Pruning after the ViT still spends 2.59 s in the encoder; the glance encoder cuts that stage to 0.65 s. (b) One TextVQA image at the same budget: outlines are kept tokens and the dotted box is the answer. DivPrune keeps 1 of 10 text tokens, relevance top-K keeps all 10 but little context, and ETD keeps 4 plus context. (c) On 4,123 TextVQA validation questions with the LT-OPD-recipe model, bars are the share of tokens overlapping the answer OCR box and diamonds the share of questions keeping at least one. The dashed line is the 5% budget.

training, sets the limit on these reading benchmarks.

We address both with **Glance-ETD**, a two-stage pipeline that first decides where to look and then which tokens to keep. The *glance encoder* sees the image at a quarter of the patches, picks a quarter of the coarse cells by diversity, and re-encodes only these at native resolution. Its output keeps the native token grid, so any selector applies afterwards. *Evidence-Tempered Diversity* (ETD) then keeps the DivPrune greedy loop and multiplies each candidate's diversity gain d_i by its evidence raised to a temperature,

$$i^\star = \arg\max_i \; d_i \, (N p_i)^{\beta t},$$

where p is the evidence map of a small query-conditioned scorer and $t \in [0, 1]$ is its confidence that the evidence is local text. A flat evidence map or a non-text question gives DivPrune; a confident reading question moves the selection toward the evidence. The scorer is trained on box annotations with the MLLM frozen, so ETD is a learned selector, and we compare it with selectors learned in the same way. ETD covers 62.9% of the answer tokens and halves the teacher KL relative to DivPrune.

On nine benchmarks with Qwen3.5-4B at a strict 5% budget, Glance-ETD retains 90.3±0.8% of the full-token score over three training seeds, 14.5 points above training-free DivPrune and 11.3 above LT-OPD, and reaches the first token on HR-Bench 4K 2.9× sooner than the full-token model (Fig. 1 and Tab. S13). The same pipeline retains 89.5±1.1% on Gemma 4 E4B-it, where LT-OPD retains 76.3% under that same strict 5% budget.

Our contributions are a glance encoder and ETD, together with a controlled study at a strict 5% budget.

- **A glance encoder** that spends full resolution on a quarter of the image and keeps the native token grid. It cuts time-to-first-token on 4K images by 2.9× at a cost of 1.4 retention points against a dense encoder trained with the same recipe and the same budget.
- **ETD**, a selection rule that keeps the diversity loop of DivPrune and tilts it by a supervised evidence map. With the MLLM frozen it improves strict-5% retention by 8.4 points over DivPrune on Qwen3.5-4B and 10.4 on Gemma 4, and after self-distillation by 2.9 points at the same recipe. Most of the gain is on reading.
- **A controlled study at a strict 5% budget**: three training regimes at the same K per image, the recipe separated from the components (a 10× encoder learning rate alone adds 3.5 points), three training seeds, and eight held-out benchmarks (DocVQA +22.0, InfoVQA +10.6 over DivPrune, held out of the experimental design).

2. Related Work and Context

2.1. Training-Free Visual Token Reduction.

Training-free methods reduce the visual sequence without updating the multimodal language model. They either prune tokens inside the language model using text-to-image attention (FastV [2], SparseVLM [25], and PyramidDrop [19]) or select/merge tokens before the language model using encoder-side signals (VisionZip [20] and PruMerge [11]). A central design choice is the trade-off between coverage and relevance. DivPrune [1] treats selection as max–min diversity and is a strong question-agnostic baseline, whereas CDPruner [22] introduces question-conditioned diversity. DART [16] removes duplicated tokens, and VisPruner [23] combines encoder attention with redundancy removal. Recent methods use zeroth-order sensitivity (ZOO-Prune [7]), within-group redundancy (PruneSID [5]), attention with text-guided complements (VisionTrim [21]) or head-weighted text-to-image attention (HAWK [26]); SPARE [8] replaces cosine diversity by subspace reconstruction, and MetaCompress [15] learns the compression across regimes.

2.2. Training with Reduced Visual Tokens.

Training can recover part of the accuracy lost by aggressive inference-time reduction. LLaVA-Mini [24] compresses visual information into a small number of query tokens before the language model, while EPIC [17] progressively distills a compressed student from a full-token teacher. LearnPruner [13] learns a pruning module after the vision encoder and combines it with text-guided pruning inside the language model. LT-OPD [9], closest to our setting, distils a student on its own rollouts from a frozen full-token copy with a curriculum over the token budget; we use its recipe and study the selector and the encoder on top of it.

2.3. Evidence-Aware High-Resolution Selection.

Text-rich inputs make evidence placement critical. ETPrune [4] prunes inside the LLM at about half of the tokens, reading question-conditioned evidence from decoder attention, protecting text-like regions and setting a per-sample budget; DocPrune [3] removes background and question-irrelevant blocks of document images. ETD differs in operating at 5% before the LLM, taking evidence from a separately trained box-supervised scorer, and keeping a diversity term, and it is combined with a reduced-cost encoder. Visual CoT [12] supplies grounded boxes, which we use only to train the scorer, and V* [18] and divide–conquer systems [14] search high-resolution images over several steps; the glance encoder instead lowers the cost of a single pass through the encoder on large images.

3. The Glance-ETD Method

Glance-ETD reduces the visual input in two stages (Fig. 3 and Alg. S1): the glance encoder (Sec. 3.2) decides where to look at full resolution, and ETD (Sec. 3.3) chooses the K tokens passed to the language model. Section 3.4 describes the teacher-KL diagnostic, and Sec. 3.5 describes training.

3.1. Preliminaries and Budgets

Problem setup. An image is encoded into N visual tokens $X = \{x_i\}_{i=1}^{N}$ in the LLM embedding space, after the patch merger. Given a question q and a budget $K \ll N$, a selector returns an index set $S \subset \{1, \dots, N\}$ with $|S| = K$, and only X_S is fed to the language model, with its original positions. We use two budget rules. *Strict*: $K = \lceil rN \rceil$ with $r = 5\%$, so that $\sum K / \sum N = 5.0\%$ on every benchmark. *Floor-32*: $K = \max(32, \lceil 0.05N \rceil)$, the rule of the LT-OPD release, which keeps 7.1% of all tokens because small images hit the floor. All selectors compared in this paper keep the same K per image and differ only in which tokens they keep, not in how many visual tokens each image receives.

Diversity selection. DivPrune [1] starts from the token closest to the centroid and greedily adds

$$i^\star = \arg\max_{i \notin S} \; d_i, \qquad d_i = \min_{j \in S} \big(1 - \cos(x_i, x_j)\big), \quad (1)$$

until $|S| = K$. The max–min rule spreads the budget over the image and ignores q. We use it in both stages: in Stage 1 to choose which regions to see at native resolution, and in Stage 2 as the diversity term of ETD.

On-policy self-distillation. LT-OPD [9] trains a student MLLM that sees only X_S to match a frozen copy of itself that sees all of X. The student generates answers with the pruned input, and the teacher provides token-level supervision along these rollouts. We adopt this objective for every model trained in this paper as well.

3.2. Stage 1: Glance Encoder

Post-encoder selection leaves the encoder untouched, and for large images the encoder dominates: on HR-Bench 4K, 89% of its FLOPs are global attention over about 50k patches. The glance encoder decides, before full-resolution encoding, which parts of the image to see in detail.

Two-pass encoding. (i) *Glance.* The image is average-pooled 2×2 and encoded with full attention at a quarter of the patches, giving one coarse token per 2×2 cell of the native merged grid. (ii) *Look.* DivPrune (Eq. (1)) on the coarse tokens selects 25% of the cells; the native patches of all selected cells are encoded as one sequence with global attention among them, with the absolute and rotary position codes of their native positions. (iii) *Merge.* After the patch merger, selected cells take their second-pass tokens and every other cell copies its coarse token to its four native slots. The grid has the size of the dense encoder's, so any post-encoder selector applies unchanged, and the LLM gives every slot its own M-RoPE position. The four copies are identical features, so once one is kept the others have $d_i = 0$ and are not selected again. The encoder costs about a sixth of the dense FLOPs, and since the 5% budget is a fifth of the high-resolution pool, the kept tokens come mostly from regions seen at native resolution.

Training. The glance encoder is trained inside the MLLM by the self-distillation objective, with an auxiliary cosine loss (weight 0.1) towards the merger output of the dense encoder. The cell choice is a discrete max–min selection and receives no gradient: training adapts the encoder features along the chosen path, not which cells are chosen. It is applied to images with at least 4096 patches; smaller images are encoded densely at native resolution.

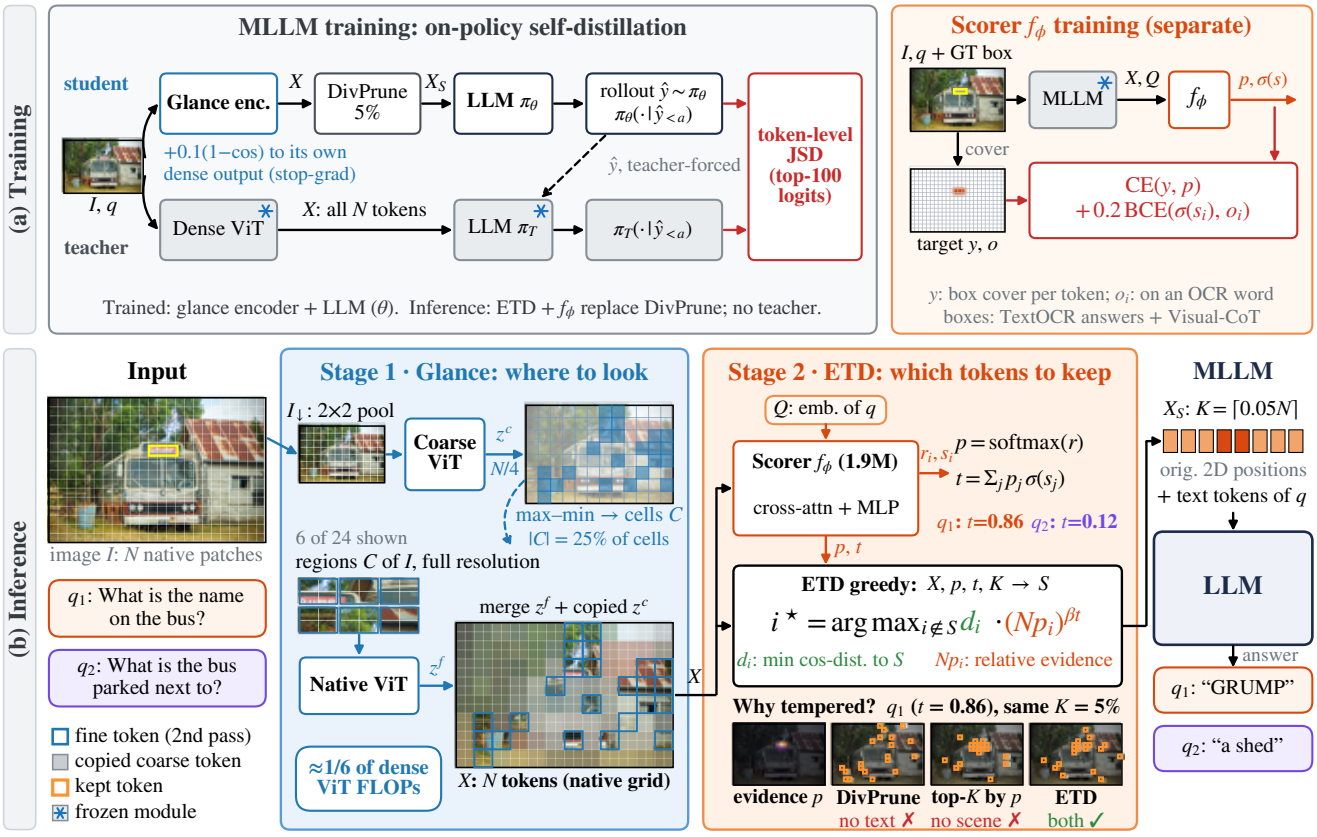


Figure 3. **Glance-ETD.** (a) Training: the student (glance encoder and LLM) selects with DivPrune, rolls out, and is distilled token by token towards a frozen full-token teacher; the scorer f_ϕ is trained separately on box targets with the MLLM frozen. (b) Inference: Stage 1 (Sec. 3.2) encodes the image at a quarter of the patches, picks 25% of the coarse cells by max–min diversity and re-encodes only these at native resolution, giving a grid X with all N native positions. Stage 2 (Sec. 3.3) maps X and the question to an evidence map p and a text confidence t, and ETD greedily selects K tokens by $d_i(Np_i)^{\beta t}$ (selections on an illustrative evidence map).

3.3. Stage 2: Evidence-Tempered Diversity

ETD keeps the diversity loop of Eq. (1) and tilts it towards query evidence by an amount set per question. It has two parts: a scorer that produces an evidence map and a confidence, and a greedy rule that uses them.

Evidence scorer. A small network f_ϕ reads the visual tokens and the question embeddings and outputs, for every visual token, a relevance logit r_i and a text logit s_i. Visual tokens are projected to $h = 256$ dimensions and given a 2D sine–cosine position code; one cross-attention layer lets them attend to the projected question tokens, producing a context c_i; an MLP on $[h_i,\, c_i,\, h_i \odot \bar{q}]$, with $\bar{q}$ the mean question feature, produces (r_i, s_i). The scorer has about 1.9M parameters and reads the MLLM's own token features, so no extra encoder is needed; with the glance encoder it reads the merged native grid, the same input that ETD selects from. From its outputs we define the evidence distribution and the *text-evidence confidence*

$$p = \mathrm{softmax}(r), \qquad t = \sum_{j=1}^{N} p_j\, \sigma(s_j) \in [0, 1], \tag{2}$$

i.e. the probability, under the scorer's own evidence map, that the evidence lies on text tokens.

Selection rule. ETD replaces the greedy score of Eq. (1) with the evidence-tempered score below,

$$i^\star = \arg\max_{i \notin S}\; d_i \left(Np_i\right)^{\beta t}. \tag{3}$$

The factor Np_i is the evidence of token i relative to a uniform map, and βt acts as an inverse temperature. If the scorer finds no evidence (p uniform), every weight is one and ETD is exactly DivPrune. For scene and object questions t is small (median 0.12 on GQA and POPE) and the selection stays close to diversity; for reading questions t is large (median 0.84 on TextVQA, 0.91 on OCRBench, on

the dense LT-OPD-recipe model) and high-evidence tokens dominate. No threshold or question classifier is involved, and we set $\beta = 1$ throughout. Because the distance term remains, a text region does not absorb the whole budget: once a few tokens of a word are kept, its neighbours have small d_i and the budget moves on. Hard splits and gates between the two criteria are compared in Sec. 4.4. We compute Eq. (3) in log space with the weights shifted by their maximum, which keeps the ranking exact.

Scorer supervision. The scorer is trained from boxes, not from benchmark labels. For TextVQA training questions we align answer words to TextOCR word boxes; for Visual-CoT [12] we use the annotated evidence boxes, which cover both text and general objects. The relevance target of token i is the fraction of its cell covered by evidence boxes, normalised over the image. The loss is the listwise cross-entropy between this target and p, plus a binary cross-entropy (weight 0.2) on $\sigma(s_i)$ for whether the cell touches an OCR word. Some images overlap evaluation sets (*e.g.* 491 of 5,349 DocVQA images); a scorer trained without them is in Sec. D.3. The 422K training records (28K TextVQA, the rest Visual-CoT) are seen for three epochs, with 60K training samples in each epoch.

3.4. Teacher-KL Diagnostic

Kept tokens stay at their native positions, and each new index maximises its cosine distance to that set, scaled by evidence (Eq. (3)). For a full-token greedy answer A of at most 32 tokens, the cost of a subset S is

$$\mathcal{L}(S) = \sum_{a=1}^{|A|} \mathrm{KL}\big(\pi(\cdot \mid X, q, A_{<a}) \,\big\|\, \pi(\cdot \mid X_S, q, A_{<a})\big), \tag{4}$$

computed with teacher forcing along A on the untrained model. On 8,000 training questions at a fixed K, ETD reaches half the $\mathcal{L}$ of DivPrune, almost all of it on text questions, and a power law in the kept fraction gives $K_m \propto N_m^{0.45}$ (Secs. D.1 and 4.4).

3.5. Training and Inference

MLLM. We train Qwen3.5-4B with on-policy self-distillation (Sec. 3.1); the student rolls out with 5% of the tokens selected by DivPrune. The training loss is the token-level generalised Jensen–Shannon divergence ($\alpha = 0.5$) between teacher and student along the student's rollouts, computed on a top-100 token support with an aggregated tail bucket; it is distinct from the KL of Eq. (4). We use one rollout per prompt and a 256-token answer cap, which shortens training from 17.4 h to 4.7 h on 8 A800 GPUs at the same accuracy (Sec. B.5). Training uses LT-OPD-14K with 1.5K spatial-reasoning samples replacing the same number of originals, for 175 steps. The vision encoder uses learning rate 2×10^{-6}, ten times the LT-OPD default, so that the glance encoder can adapt to its two-resolution input. This is a change of training recipe, not part of either component: it alone raises the dense model with DivPrune from 85.3% to 88.8%, and every component comparison in Sec. 4.3 holds this encoder learning rate fixed.

Scorer and inference. The scorer is trained once on the features of the untrained MLLM and shared by all Qwen models. Retraining it on the features of each trained model gives 91.5% on the dense-encoder model (91.9% with the shared scorer, seed 0) and $90.0 \pm 0.4\%$ on the glance model over three seeds ($90.3 \pm 0.8\%$), so the shared scorer is kept. At inference, the glance encoder produces the token grid, the scorer computes p and t, and ETD selects K tokens, which enter the LLM with their original positions. The MLLM is thus trained with DivPrune and evaluated with ETD. This keeps the MLLM independent of the scorer, so the same rule applies to the untrained model; training with ETD instead is compared in Sec. 4.3.

4. Experiments and Results

4.1. Setup, Benchmarks, and Budgets

Models. All experiments use Qwen3.5-4B [10] in non-thinking mode. **Glance-ETD** is the self-distilled model with the glance encoder (Sec. 3.5), evaluated with ETD; the *dense-encoder variant* is trained identically without the glance encoder; *untrained* is the released model. Methods are grouped by what is trained: nothing; an added module with the MLLM frozen (ETD's scorer and the learned baselines); or the MLLM itself in that group.

Benchmarks and metric. We report the nine benchmarks of LT-OPD: V* [18], HR-Bench 4K [14], GQA, MMMU, MMBench (circular), MME, POPE, TextVQA and OCR-Bench, summarised by the *retention*, the mean ratio of each score to the same model's full-token score (first row of Tab. 1; repeated full-token runs in Sec. B.2). The mean hides per-benchmark drops, so Tab. 1 lists every score: Glance-ETD keeps 65% of the full-token OCRBench score at 90.3% retention. Eight benchmarks added after all design decisions serve as a held-out test (Sec. D.2).

Budgets and baselines. The main budget is strict 5% (Sec. 3.1); floor-32 is in Sec. E.1. Baselines are peer-reviewed methods from 2025–2026 (Tab. 1), re-run on Qwen3.5-4B with the same K per image as ETD. Training-free methods are ported from the official code to select after the vision merger (Sec. B.3). The modules of LLaVA-Mini, LearnPruner and MetaCompress are trained on LT-OPD-14K with the MLLM frozen. DPP-r keeps CDPruner's

Table 1. Strict 5% budget, same K per image for all methods. Rows are grouped by what is trained: nothing, an added module (MLLM frozen; ETD trains only its scorer), or the MLLM. DPP-r: CDPruner's DPP on our scorer. Self-distillation + DivPrune / + ETD: the same training recipe with either selector; + ETD is ours without glance. Ret.: mean ratio to full tokens over the nine LT-OPD benchmarks; the five held-out columns are excluded. Ours: mean±std of three seeds (Tab. S1). –: running; ‡: GQA lower-cased.

	General					Fine-grained high resolution					Text and document				
Method	GQA	MMMU	MMB	MME	POPE	V*	HR-4K	HR-8K	MME-RW	HRScene	TextVQA	OCRB	DocVQA	InfoVQA	Ret. (%)
Training-free								*Qwen3.5-4B*							
Full tokens (100%)	60.0	51.3	79.0	2218	89.9	80.6	78.0	72.6	48.9	62.5	80.5	848	94.6	79.4	100.0
DivPrune [1] (CVPR'25)	45.3	46.3	63.4	1695	77.4	70.2	73.4	70.5	38.1	49.6	49.2	264	32.1	21.7	75.8
DART [16] (EMNLP'25)	45.0	48.2	53.6	1529	77.6	49.7	61.5	60.4	30.1	45.0	49.6	125	24.9	21.8	67.7
CDPruner [22] (NeurIPS'25)	35.7	47.2	59.7	1605	74.3	53.9	62.7	56.4	29.7	43.3	43.3	215	21.8	20.5	67.6
PruneSID [5] (ICLR'26)	44.3	47.0	55.0	1602	77.9	69.6	73.8	68.8	35.0	49.2	25.1	108	22.3	20.7	68.7
VisionTrim [21] (ICLR'26)	41.8	45.8	55.4	1514	69.4	48.7	63.8	59.0	23.9	43.8	20.3	89	15.8	21.5	61.4
HAWK [26] (CVPR'26)	30.7	47.4	49.0	1336	70.9	49.2	60.4	56.2	22.5	44.3	16.5	72	9.7	18.9	56.9
ZOO-Prune [7] (CVPR'26)	46.2	46.4	60.1	1646	81.3	72.3	72.2	67.9	36.8	49.4	26.8	118	25.5	20.8	70.9
Frozen MLLM, learned module								*Qwen3.5-4B*							
LLaVA-Mini [24] (ICLR'25)	43.5	47.9	65.3	1579	75.7	56.5	68.6	61.7	32.4	50.1	34.9	236	34.7	27.4	70.3
LearnPruner [13] (ICLR'26)	43.3	45.6	55.1	1505	72.1	48.2	69.0	60.9	29.2	43.7	28.2	87	12.7	19.3	63.6
MetaCompress [15] (CVPR'26)	41.9	46.9	66.1	1576	69.8	63.4	70.4	–	–	–	41.2	241	–	–	71.3
DPP-r	40.6	46.1	64.3	1736	85.5	79.1	73.8	69.1	42.0	51.5	70.7	420	68.1	34.5	82.5
ETD (ours)	48.4	47.3	62.3	1750	84.3	77.0	76.4	73.0	41.8	51.8	72.2	427	74.3	39.2	**84.2**
MLLM adaptation								*Qwen3.5-4B*							
EPIC‡ [17] (NeurIPS'25)	42.1	44.4	57.0	1233	66.3	54.5	64.8	59.8	39.8	48.2	50.9	120	25.4	22.9	65.1
LT-OPD [9] (arXiv'26.09)	47.4	48.6	66.4	1771	81.9	67.0	69.6	65.1	40.5	51.4	55.8	341	40.6	25.6	79.0
Self-distillation + DivPrune	54.2	48.3	73.5	1988	85.9	75.0	77.1	73.5	45.9	56.2	70.4	489	68.4	36.0	88.8±0.2
Self-distillation + ETD (ours)	55.1	48.7	75.2	2000	87.8	77.1	76.2	74.4	46.1	55.9	76.0	579	79.7	45.8	91.7±0.2
Glance-ETD (ours)	54.5	48.2	72.8	1932	86.7	77.0	76.7	72.1	45.5	53.9	75.2	553	76.4	43.4	**90.3**±0.8
Training-free								*Gemma 4 E4B-it*							
Full tokens (100%)	41.3	46.9	73.6	1686	88.1	58.1	61.1	54.4	42.4	54.6	71.9	764	88.3	62.0	100.0
DivPrune [1] (CVPR'25)	37.1	42.9	60.5	1103	83.9	49.2	53.0	46.1	23.7	43.7	38.9	184	14.7	15.0	74.8
DART [16] (EMNLP'25)	33.2	41.3	52.5	934	72.4	34.6	40.2	35.8	17.9	34.7	19.9	83	9.1	12.3	60.2
CDPruner [22] (NeurIPS'25)	35.3	42.2	54.7	962	78.4	28.8	44.4	38.6	14.3	37.3	28.9	138	10.4	13.5	64.0
PruneSID [5] (ICLR'26)	36.8	41.1	60.2	1164	79.9	27.2	41.2	38.4	21.4	39.7	30.9	246	12.3	16.1	67.5
VisionTrim [21] (ICLR'26)	33.5	38.4	55.7	1046	78.0	41.4	54.4	47.5	22.7	39.6	24.0	138	9.6	15.6	66.7
HAWK [26] (CVPR'26)	30.4	40.2	50.4	874	69.2	21.5	30.3	26.9	10.8	31.3	9.6	85	7.4	13.6	52.1
ZOO-Prune [7] (CVPR'26)	36.5	43.1	60.1	1062	81.8	34.0	41.2	36.6	19.9	39.9	24.9	151	10.6	15.2	66.5
Frozen MLLM, learned module								*Gemma 4 E4B-it*							
LLaVA-Mini [24] (ICLR'25)	30.3	39.9	30.5	1018	65.5	38.7	32.5	35.4	29.2	34.9	11.9	62	8.5	18.3	53.2
LearnPruner [13] (ICLR'26)	32.6	40.9	47.9	952	72.7	28.3	36.3	37.3	14.6	35.3	13.3	25	4.1	12.0	55.6
MetaCompress [15] (CVPR'26)	35.6	40.6	59.4	1063	77.7	27.2	33.9	31.2	17.6	36.8	22.9	196	10.8	16.5	62.7
ETD (ours)	37.0	43.0	62.3	1230	82.2	54.5	60.2	54.0	31.1	41.7	61.0	440	42.3	23.2	**85.2**
MLLM adaptation								*Gemma 4 E4B-it*							
EPIC [17] (NeurIPS'25)	40.5	43.1	55.3	1362	78.2	44.5	47.6	42.2	34.5	41.5	24.5	171	13.7	17.9	71.7
LT-OPD [9] (arXiv'26.09)	39.3	43.1	63.5	1176	82.5	37.7	52.2	47.8	27.5	42.6	39.1	349	16.2	16.1	76.3
Self-distillation + DivPrune	41.3	42.9	70.8	1386	87.3	49.7	57.3	53.0	30.2	44.4	56.8	440	25.8	18.8	87.2±1.2
Self-distillation + ETD (ours)	40.9	43.9	70.8	1471	86.1	48.9	59.1	53.1	33.9	45.8	64.1	513	43.2	25.2	90.1±0.9
Glance-ETD (ours)	40.8	44.1	69.5	1528	86.1	45.9	57.5	47.4	31.8	45.0	60.8	553	35.8	24.7	**89.5**±1.1

conditional DPP and replaces its question–token similarity by the min-max normalised relevance logits of the same scorer. LT-OPD is trained with its LT-OPD-14K recipe on each model and passes the same token subset as the other methods (Sec. B.4 states that shared recipe).

Implementation. Evaluation follows VLMEvalKit/lmms-eval protocols with greedy decoding on A800 GPUs; unresolved MMBench groups are adjudicated by fixed rules without a judge model (Sec. B.2). Inference is not bit-exact: the same model scored 73.3 and 71.2 on V* in two runs, so 1–2 points on V* are noise.

4.2. Main Results across Benchmarks

Frozen MLLM. With the MLLM untouched, ETD keeps 84.2% at strict 5%, 8.4 points above DivPrune, the strongest training-free baseline (Tab. 1). The learned modules of LLaVA-Mini and LearnPruner stay below DivPrune. DPP-r reaches 82.5%, 6.7 of ETD's 8.4 points over DivPrune, so most of that gain is the supervised evidence rather than the selection rule. The other 1.7 points come from GQA (40.6 against ETD's 48.4). Conditional DPP multiplies its kernel by the relevance on every question. ETD multiplies max–min diversity by $(Np_i)^{\beta t}$ (Eq. (3)), and t is small on scene questions, so those selections stay close to

Table 2. Component ablation at a strict 5% budget (same K per image). Glance: two-pass glance encoder (Sec. 3.2); without it the dense encoder is used. ETD: evidence-tempered selection (Sec. 3.3); without it DivPrune. The 10× columns share one recipe and differ only in the two components; 1× is the LT-OPD recipe; No train leaves the MLLM untouched. TFLOPs: prefill (encoder + LLM, attention included) on HR-Bench 4K, set by the encoder and K; the ETD scorer adds about 0.02. Trained entries are means over three training seeds, ± the standard deviation of Ret.

	10× LR				1× LR		No train	
Glance	–	–	✓	✓	–	–	–	–
ETD	–	✓	–	✓	–	✓	–	✓
V*	75.0	77.1	74.7	77.0	73.8	77.0	70.2	77.0
HR-4K	77.1	76.2	76.7	76.7	77.2	76.8	73.4	76.4
GQA	54.2	55.1	53.4	54.5	52.3	53.1	45.3	48.4
MMMU	48.3	48.7	49.1	48.2	48.9	48.9	46.3	47.3
MMB	73.5	75.2	71.7	72.8	70.6	71.3	63.4	62.3
MME	1988	2000	1902	1932	1865	1858	1695	1750
POPE	85.9	87.8	84.5	86.7	85.2	87.9	77.4	84.3
TextVQA	70.4	76.0	70.2	75.2	64.7	73.2	49.2	72.2
OCRB	489	579	474	553	394	488	264	427
Ret.	88.8 ±0.2	91.7 ±0.2	87.7 ±0.4	90.3 ±0.8	85.3 ±0.2	88.7 ±0.4	75.8	84.2
TFLOPs	373	373	65	65	373	373	373	373

DivPrune. V* moves the other way (79.1 against 77.0). On the trained glance model the same substitution costs 0.2 points (Tab. 3). The importance-based methods (ZOO-Prune, PruneSID, VisionTrim, HAWK; 56.9–70.9%) concentrate a 5% budget on a few salient regions and lose most of the text (TextVQA 17–27, OCRBench 72–118, against 49.2 and 264 for DivPrune and 72.2 and 427 for ETD). ETD also improves V*, HR-Bench and POPE over DivPrune (+6.8, +3.0, +6.9) and leads at every budget from 1% to 25% (Sec. E.2 reports that sweep).

MLLM adaptation. Trained rows are means over three training seeds (Tab. S1). LT-OPD retains 79.0%. Under our self-distillation recipe, which differs only in the selector, DivPrune retains 88.8% and ETD 91.7%; Glance-ETD retains 90.3±0.8%. Section 4.3 splits the gap to LT-OPD into recipe and components. EPIC (65.1%) stays below DivPrune, mostly on reading. At floor-32 (7.1% of the tokens) LT-OPD retains 86.1%, the LT-OPD recipe with DivPrune 92.2% and Glance-ETD 96.0±0.4% (Sec. E.1).

Gemma 4. On Gemma 4 E4B-it [6] (∼1,100 tokens per image) with the MLLM frozen, ETD keeps 85.2%, 10.4 points above DivPrune and above the other training-free methods (52.1–67.5%), mostly on reading (TextVQA 38.9→61.0, OCRBench 184→440; Sec. E.5). With MLLM adaptation, Glance-ETD retains 89.5±1.1% and ours without glance 90.1±0.9% (LT-OPD 76.3%).

4.3. Component Ablation Study

Table 2 removes each stage at strict 5%. In the 10× columns both encoders use the same recipe with a 10× encoder learning rate and differ only in the two components. The learning rate alone adds 3.5 points to the LT-OPD recipe with DivPrune (85.3→88.8%). At this recipe ETD adds 2.9 (91.7%), and the glance encoder costs 1.4 while cutting HR-Bench 4K prefill from 373 to 65 TFLOPs and time to first token from 3305 to 1376 ms on HR-Bench 4K.

ETD. ETD improves every model at the same K: +2.9 on the dense encoder, +2.6 on the glance encoder, +3.4 with the LT-OPD recipe and +8.4 without MLLM training, mostly on TextVQA and OCRBench (70.2→75.2 and 474→553 on the glance encoder). The MLLM is trained with DivPrune and evaluated with ETD. Training with ETD too adds 1.1 on the dense 1×-LR model (85.1 / 87.9 / 89.0 for DivPrune→DivPrune / DivPrune→ETD / ETD→ETD, one seed), and that model falls to 83.7 with DivPrune: the MLLM co-adapts with its selector. On the glance model training with ETD gives no gain (89.2 *vs*. 89.5 with DivPrune training, seed 0) and the model falls from 87.5 to 86.2 with DivPrune, so we keep DivPrune training, which also leaves the MLLM independent of the scorer.

Glance. At the same recipe the glance encoder costs 1.1 points with DivPrune and 1.4 with ETD. V* and HR-Bench 4K are unchanged (77.0 *vs*. 77.1, 76.7 *vs*. 76.2); the loss is on MMBench (−2.4) and MME (−68). MMBench images are below the glance threshold and encoded densely, so this loss comes from training the shared encoder (Sec. D.4). Glance is thus an efficiency component.

4.4. What matters in the selection rule

Table 3 changes the selection rule on Glance-ETD (Qwen, seed 0) and training-free ETD (Gemma); dense model results are reported in Tab. S5 as well.

Evidence and diversity do different jobs. Dropping the evidence term leaves DivPrune and costs 2.0 points on Qwen and 10.4 on Gemma, mostly on reading (Gemma TextVQA 61.0→38.9, OCRBench 440→184). Dropping diversity and keeping the top K tokens by p costs 2.4 and 3.9: it raises V* and HR-Bench on Qwen but lowers GQA from 54.4 to 51.2 and OCRBench from 552 to 459. Giving the scorer half of the budget costs 1.9 and 2.8, and a hard gate at $t > 0.5$ costs 1.3 and 4.3; on Qwen the gate's V*, GQA and POPE are those of DivPrune.

Table 3. Selection-rule ablation at a strict 5% budget and the same K per image. Qwen is Glance-ETD; Gemma 4 is training-free ETD. Ret. is over nine benchmarks against that model's full-token score; six are shown. Δ is the difference in nine-benchmark retention. Qwen rows are single runs of training seed 0 (90.3 in Tab. 1 is the three-seed mean); three-seed comparisons are in Sec. 4.4. –: not run.

	Qwen3.5-4B, Glance-ETD								Gemma 4 E4B-it, ETD (training-free)							
Setting	V*	HR-4K	GQA	POPE	TVQA	OCRB	Ret.	Δ	V*	HR-4K	GQA	POPE	TVQA	OCRB	Ret.	Δ
Full pipeline	75.9	75.9	54.4	87.3	75.4	552	89.5	–	54.5	60.2	37.0	82.2	61.0	440	85.2	–
Stage 2: which tokens (same K per image)																
w/o evidence: DivPrune ($\beta t = 0$)	73.3	77.4	53.6	84.6	70.3	475	87.5	-2.0	49.2	53.0	37.1	83.9	38.9	184	74.8	-10.4
w/o diversity: top-K by p ($\beta = 64$, no t)	76.4	77.8	51.2	86.5	72.6	459	87.1	-2.4	52.4	57.2	37.3	82.9	51.7	358	81.3	-3.9
w/o confidence t: $d_i(Np_i)$	77.0	75.1	53.0	87.5	75.6	542	88.9	-0.6	50.3	65.4	35.7	79.6	60.7	434	84.7	-0.5
t replaced by constant 0.25	75.4	76.9	54.7	87.5	75.7	553	90.4	+0.9	55.0	62.0	37.7	83.3	59.9	396	84.8	-0.4
hard split: top-$K/2$ by p + DivPrune	73.3	75.2	52.6	85.3	73.1	505	87.6	-1.9	49.7	60.9	37.0	83.3	55.3	363	82.4	-2.8
hard gate: split only if $t > 0.5$	73.3	76.9	53.5	84.5	72.8	504	88.2	-1.3	50.3	54.1	37.0	83.3	55.0	362	80.9	-4.3
$\beta = 0.5$	74.9	76.8	54.3	86.6	75.7	554	89.9	+0.4	51.8	57.1	37.6	82.6	60.8	423	84.0	-1.2
$\beta = 2$	78.5	75.9	54.4	87.6	75.6	530	89.6	+0.1	52.4	62.4	37.1	80.7	60.2	426	84.6	-0.6
scorer given a wrong question	74.9	75.8	53.2	86.1	71.3	464	87.4	-2.1	54.5	57.2	36.4	81.8	54.7	319	81.5	-3.7
DPP-r instead of ETD	74.3	75.9	51.7	88.0	75.9	556	89.3	-0.2	–	–	–	–	–	–	–	–

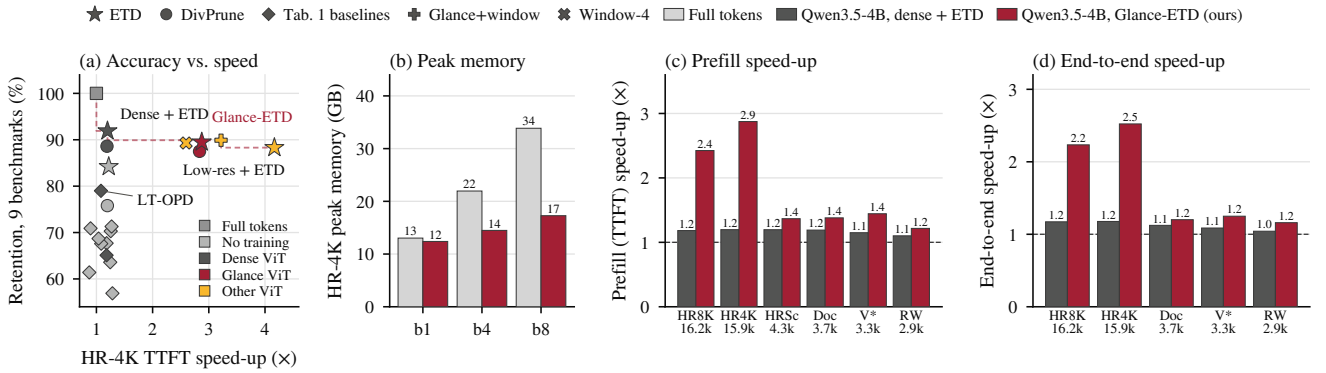


Figure 4. Efficiency of Qwen3.5-4B at strict 5% (one A800, bf16). (a) Retention (9 benchmarks, seed 0 of Tab. S4; baselines: Tab. 1) against HR-Bench 4K TTFT speed-up; stars ETD, circles DivPrune, diamonds the other Tab. 1 methods (Tab. S14), dashed line the Pareto front. (b) HR-Bench 4K peak memory at batch 1, 4 and 8. (c) Prefill-stage (TTFT: ViT, selection, LLM prefill) and (d) end-to-end speed-up over full tokens at batch 1 on many-token benchmarks at native size (visual tokens per image below each name; omitted when no bar exceeds 1.15×), 30 questions each, same run as Tab. S13. Resolution sweep, numbers and further views: Tab. S13 and Fig. S2.

Scorer or rule? Over three training seeds of the glance model, a constant exponent of 0.25 is 0.5 points above βt (paired bootstrap over questions, 95% interval $[0.1, 1.0]$), and DPP-r is 0.2 below ETD ($[-0.8, 0.5]$); $\beta \in \{0.5, 2\}$ and dropping t move retention by at most a point. As a detector, t separates reading questions (TextVQA, OCR-Bench) from perception questions with AUROC 0.90, but the best tilt does not grow with it: in each t quintile the best constant exponent is about 0.25, and choosing it per quintile gains only 0.1 over a single 0.25. The gain comes from p, not from t. Replacing p matters more: the ETD rule with question–token similarity gives 87.2 (0.3 below DivPrune), and a scorer fed a different question costs 2.1 points on Qwen and 3.7 on Gemma. ETD matches DPP-r without a kernel or a determinant.

Does ETD keep the answer? On TextVQA val (4,123 questions, strict 5%), a kept token counts when its cell overlaps the answer OCR box: 62.9% of those tokens and at least one on 99.0% of questions, against 10.4% and 80.2% for DivPrune and at most 11.5% for the other selectors (Fig. 2c, Sec. E.4).

4.5. Latency, Memory, and FLOPs

Figure 4 reports speed and memory (Sec. F). On HR-Bench 4K the ViT is 2.59 s of a 3.96 s TTFT, and dense-encoder selectors stay at 0.87–1.28× (LT-OPD 1.08×, ETD 1.22×; peak memory 13.0–13.3 GB, VisionTrim 14.7 GB). Glance cuts the ViT to 0.65 s (2.87× prefill, 2.52× end to end) for 1.4 points and halves batch-8 memory from 34.7 to 17.7 GB; low-resolution and windowed encoders lose 3–8 points on V* and HR-Bench (Fig. 4 and Tab. S14).

5. Conclusion and Limitations

Glance-ETD splits an extreme visual-token budget into two decisions: the glance encoder chooses where the image is seen at full resolution, removing encoder cost that token selection cannot touch, and ETD tilts diversity selection towards a supervised evidence map, which carries most of its gain on the reading benchmarks.

Limitations. Stage 1 ignores the question and covers text evidence about as well as random cells; the scorer needs box supervision per backbone; small images gain no encoder saving; and the best of 16 subsets is 8 points above ETD.

Statement on LLM Usage

We used Claude Code (Anthropic, Claude models) as a coding and writing assistant. It wrote and ran parts of the training, evaluation and analysis scripts under the authors' direction, generated the LaTeX tables from the evaluation outputs with scripts, and drafted and revised the manuscript text, which the authors edited and checked. It did not generate experimental measurements: every reported number comes from evaluation runs of the models, and MMBench adjudication is rule-based without a judge model. Citations were checked against the source papers by the authors. No generative model is an author of this paper.

References

[1] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *CVPR*, 2025. 1, 2, 3, 6

[2] 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 *ECCV*, 2024. 2

[3] Joonmyung Choi, Sanghyeok Lee, Jongha Kim, Sehyung Kim, Dohwan Ko, Jihyung Kil, and Hyunwoo J. Kim. DocPrune: Efficient document question answering via background, question, and comprehension-aware token pruning. In *CVPR*, 2026. 3

[4] Zizhong Ding, Junxian Li, Kai Liu, Shaoqiu Zhang, Xiao Xiao, Linghe Kong, and Yulun Zhang. ET-Prune: Evidence-aware dynamic budgeting for visual token pruning in text-rich MLLMs. *arXiv preprint arXiv:2608.01979*, 2026. 3

[5] Zhengyao Fang et al. Prune redundancy, preserve essence: Vision token compression in VLMs via synergistic importance-diversity. In *ICLR*, 2026. 2, 6

[6] Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. 7

[7] Kim et al. ZOO-Prune: Training-free token pruning via zeroth-order gradient estimation in vision-language models. In *CVPR*, 2026. 2, 6

[8] Jaeyeon Lee, Shunjie Wen, and Dong-Wan Choi. Moving beyond diversity: Visual token pruning as subspace reconstruction for efficient VLMs. *arXiv preprint arXiv:2606.18681*, 2026. 2

[9] Junxian Li, Ruixuan Yang, Tianao Zhang, Tiange Xu, Weisheng Dong, and Yulun Zhang. Fewer tokens, more self-teaching: On-policy self-distillation for extreme visual token reduction. *arXiv preprint arXiv:2609.32353*, 2026. 1, 3, 6

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

[11] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. LLaVA-PruMerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024. 2

[12] Hao Shao et al. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. In *NeurIPS Datasets and Benchmarks*, 2024. 3, 5

[13] Takezoe et al. LearnPruner: Rethinking attention-based token pruning in vision language models. In *ICLR*, 2026. 3, 6

[14] Wenbin Wang et al. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *AAAI*, 2025. 3, 5

[15] Yi Wang, Haofei Zhang, et al. Rethinking token reduction for large vision-language models. In *CVPR*, 2026. 2, 6

[16] 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 *EMNLP*, 2025. 2, 6

[17] Zichen Wen, Shaobo Wang, et al. Efficient multi-modal large language models via progressive consistency distillation. In *NeurIPS*, 2025. 3, 6

[18] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal LLMs. In *CVPR*, 2024. 3, 5

[19] Long Xing et al. PyramidDrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *CVPR*, 2025. 2

[20] Senqiao Yang et al. VisionZip: Longer is better but not necessary in vision language models. In *CVPR*, 2025. 2

[21] Hanxun Yu et al. VisionTrim: Unified vision token compression for training-free MLLM acceleration. In *ICLR*, 2026. 2, 6

[22] Qizhe Zhang et al. Beyond attention or similarity: Maximizing conditional diversity for token pruning in MLLMs. In *NeurIPS*, 2025. 2, 6

[23] Qizhe Zhang et al. Beyond text-visual attention: Exploiting visual cues for effective token pruning in VLMs. In *ICCV*, 2025. 2

[24] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-Mini: Efficient image and video large multimodal models with one vision token. In *ICLR*, 2025. 3, 6

[25] Yuan Zhang et al. SparseVLM: Visual token sparsification for efficient vision-language model inference. In *ICML*, 2025. 2

[26] Zhu et al. HAWK: Head importance-aware visual token pruning in multimodal models. In *CVPR*, 2026. 2, 6