DIFFERENTIABLE EFFICIENT OPERATOR SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

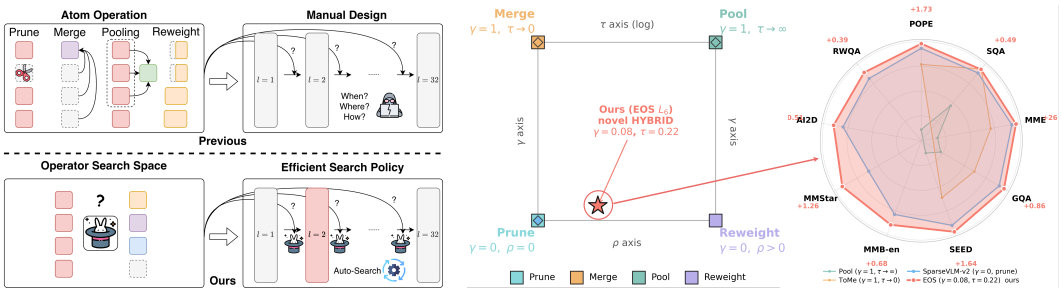
Efficient vision-language models have largely relied on human-designed reduction operators, such as pruning, merging, pooling, and adaptive reweighting, which could happen at any time and anywhere. We show that these seemingly different methods can be unified as different operating regimes of a single shared operator space. Based on this observation, we introduce **Efficient Operator Search**, a unified framework in which continuous parameters control whether token information is removed, sharply merged, uniformly pooled, or softly redistributed. Instead of hand-crafting operator compositions, we define an ***efficient search space***: a differentiable parameterization of layer activation, retention budget, and operator regime, and an ***efficient search policy***: minimizing the expectation of the task loss under one-sided budget and cost constraints. The pipeline leads to a broader shift from an ***efficient design problem*** to an ***operator search problem***, suggesting a new paradigm for future efficient modeling. Interestingly, we find that some prior mainstream baselines can be treated as special cases of this shared operator space, and further reveal consistent operator patterns across different benchmarks. Even as an early exploration of this new paradigm, our proposed method achieves competitive results on various benchmarks. We believe that it provides a unified view of prior methods, a useful lens for understanding their differences, and a promising basis for exploring future efficient modeling.

1 INTRODUCTION

Scaling multi-modal foundation models has led to substantial performance gains across visual reasoning, multi-modal understanding, instruction following, and long-context multimodal reasoning (Liu et al., 2023; Li et al., 2023b; Team et al., 2023; 2024; Hurst et al., 2024; Wang et al., 2024; Li et al., 2024). However, these gains often come with substantial inference cost, as dense visual tokens are repeatedly processed by large language backbones, especially in high-resolution, multi-image, and long-context scenarios (Zhang et al., 2024a; Liu et al., 2026; Shi et al., 2025; 2026; Yang et al., 2025b; Shang et al., 2025; Pei et al., 2025a). Existing work improves the efficiency of foundation models through human-designed token reduction operators, including pruning, merging, pooling, and adaptive reweighting. For example, SparseVLM Zhang et al. (2024b) removes visually unimportant tokens according to attention scores, while ToMe Bolya et al. (2022) merges redundant tokens to shorten the input sequence while preserving information. Although these methods demonstrate promising accuracy-efficiency trade-offs, they are usually developed as separate compression recipes with manually chosen layers, budgets, and operator forms, leaving the shared structure behind these corner operators largely underexplored.

Despite their different implementations, existing reduction behaviors can be interpreted as different ways of discarding, transferring, or redistributing token information. This interpretation motivates a shared formulation in which representative efficient operators are controlled by continuous parameters. Specifically, an information-transfer gate γ determines whether discarded token information is removed or preserved, while an assignment temperature τ controls whether the preserved information is sharply merged, uniformly pooled, or softly redistributed. Under this view, pure pruning, hard merging, average pooling, and adaptive reweighting naturally emerge as different operating regimes of the same operator space (Figure 1). Based on this observation, we propose **Efficient Operator Search**, a unified framework that reframes efficient models from an *operator design problem* into an *operator search problem*. Instead of manually crafting another fixed compression rule, our method searches over where to compress, how many tokens to retain, and which reduction behavior to apply.

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.



(a) Manual design vs. automatic search.

(b) Unified operator space and performance.

Figure 1: **Overview of Efficient Operator Search.** (a) EOS replaces manually designed reduction recipes with automatic operator search. (b) The searched hybrid operator lies inside the unified operator space and improves performance under the same token budget.

This enables flexible combinations of multiple compression primitives under different accuracy-efficiency constraints. We conduct extensive experiments on representative multi-modal benchmarks. The results show that our framework can recover strong hand-designed baselines and further discover better operator configurations, achieving improved performance-efficiency trade-offs.

Our contributions can be summarized from three aspects. (1) We introduce a unified operator space that reinterprets existing token reduction methods as different regimes of one shared formulation. (2) We propose Efficient Operator Search, which jointly searches layer selection, token budgets, and continuous operator parameters instead of relying on manually designed recipes. (3) We demonstrate the proposed framework can recover existing baselines and discover stronger hybrid configurations for efficient multimodal foundation models.

2 Method

We aim to transform efficient foundation models from manually designed compression recipes into a searchable operator space. Given a frozen multimodal foundation model with L decoder layers, existing token-reduction methods typically hand-specify *where* to reduce visual tokens, *how many* tokens to retain, and *which* operator to apply. In contrast, our **Efficient Operator Search** (EOS) parameterizes these choices as learnable search variables and jointly optimizes them with differentiable relaxations under task and efficiency constraints. As shown in Figure 2, EOS selects important visual-token anchors at each active reduction layer, then applies a unified operator to decide whether the remaining token information is discarded, transferred, pooled, or reweighted.

2.1 Preliminary

Let $\mathbf{X}^{(l)} \in \mathbb{R}^{N_l \times d}$ denote the visual hidden states at decoder layer l, where N_l is the number of visual tokens and d is the hidden dimension. A token-reduction operation partitions $\mathbf{X}^{(l)}$ into a retained anchor set and a reduction-candidate set:

$$\mathbf{X}^{(l)} = \mathbf{A}^{(l)} \cup \mathbf{D}^{(l)}, \qquad \mathbf{A}^{(l)} \in \mathbb{R}^{K_l \times d}, \quad \mathbf{D}^{(l)} \in \mathbb{R}^{M_l \times d}, \tag{1}$$

where K_l is the number of retained anchors, $M_l = N_l - K_l$ is the number of reduction candidates, and $\mathbf{A}^{(l)} \cap \mathbf{D}^{(l)} = \emptyset$. The anchors are selected according to an importance score $\mathbf{q}^{(l)} \in \mathbb{R}^{N_l}$, which can be instantiated by attention-based or similarity-based token importance following prior efficient inference methods. The remaining question is how to process $\mathbf{D}^{(l)}$ after anchor selection. Pure pruning directly removes $\mathbf{D}^{(l)}$, token merging transfers each discarded token to a similar anchor, pooling distributes discarded information uniformly, and adaptive reweighting rescales anchors according to discarded-token importance. EOS treats these behaviors as different operating regimes of one shared operator family.

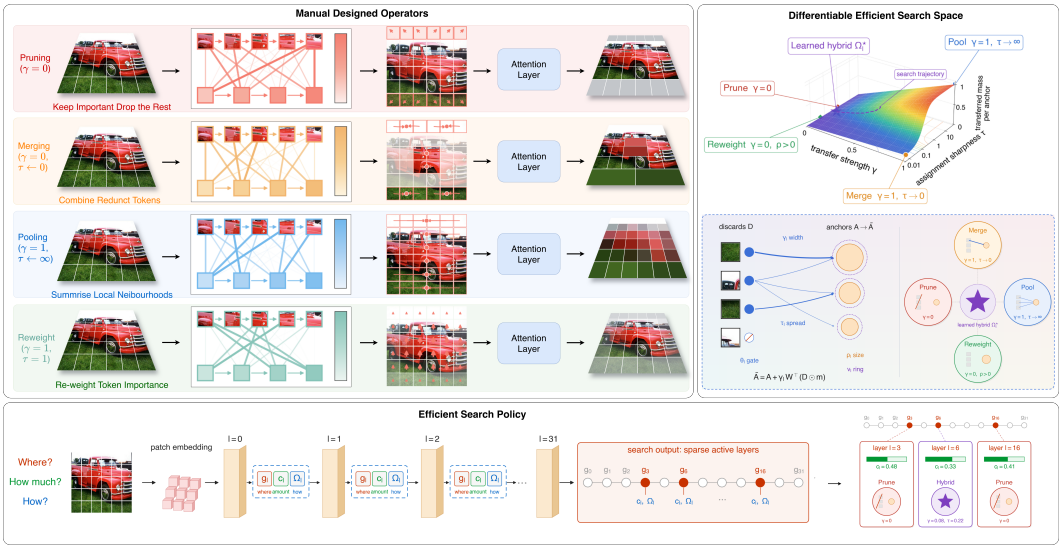


Figure 2: **Overview of Efficient Operator Search.** Given a frozen multimodal foundation model, EOS parameterizes token reduction at each decoder layer by three coupled components: layer activation g_l, retention budget c_l, and operator regime $\Omega_l = (\gamma_l, \tau_l, \theta_l, \rho_l, \nu_l)$. At each active layer, important visual tokens are retained as anchors, while the remaining candidates are processed by a unified reduction operator. Depending on the learned operator parameters, discarded information can be removed, sharply merged, uniformly pooled, or softly redistributed to the anchors.

2.2 Unified Reduction Operator

For simplicity, we omit the layer index l when the context is clear. Given anchors $\mathbf{A} \in \mathbb{R}^{K\times d}$ and reduction candidates $\mathbf{D} \in \mathbb{R}^{M\times d}$, we first compute the normalized similarity matrix:

$$\mathbf{S} = \bar{\mathbf{D}}\bar{\mathbf{A}}^\top \in \mathbb{R}^{M\times K}, \qquad \bar{\mathbf{x}}_i = \frac{\mathbf{x}_i}{\|\mathbf{x}_i\|_2 + \epsilon}, \tag{2}$$

where $\mathbf{x}_i$ denotes a row token feature from either $\mathbf{A}$ or $\mathbf{D}$, the normalization is applied token-wise, and $\epsilon > 0$ is a small constant used for numerical stability. Each entry S_{ij} measures the cosine similarity between reduction candidate $\mathbf{d}_i$ and anchor token $\mathbf{a}_j$. The assignment from reduction candidates to anchors is given by

$$\mathbf{W} = \mathrm{softmax}_{\mathrm{anchor}}\left(\frac{\mathbf{S}}{\tau_l}\right) \in \mathbb{R}^{M\times K}, \tag{3}$$

where $\mathrm{softmax}_{\mathrm{anchor}}(\cdot)$ normalizes each row over the anchor dimension, and $\tau_l > 0$ is an assignment temperature. A smaller τ_l induces sharper nearest-anchor assignment, while a larger τ_l yields smoother assignment across anchors.

We further define a per-token transfer gate:

$$m_i = \sigma\left(\beta(\max_j S_{ij} - \theta_l)\right), \qquad \mathbf{m} = [m_1, \ldots, m_M]^\top, \tag{4}$$

where θ_l is a similarity threshold and β controls the gate sharpness. The unified information-transfer operation is then given by

$$\tilde{\mathbf{A}} = \mathbf{A} + \gamma_l\, \mathbf{W}^\top(\mathbf{D} \odot \mathbf{m}), \tag{5}$$

where $\mathbf{m}$ is broadcast along the feature dimension, and $\gamma_l \in [0, 1]$ is an information-transfer gate. When $\gamma_l = 0$, the reduction candidates have no feature transfer and are effectively pruned. When $\gamma_l > 0$, their information is transferred to the retained anchors via $\mathbf{W}$.

To include adaptive reweighting and stabilize the output scale, we further apply

$$\hat{\mathbf{A}} = \tilde{\mathbf{A}} \odot (1 + \rho_l\, \mathbf{s}_{\mathrm{imp}}), \qquad \hat{\mathbf{A}} \leftarrow (1 - \nu_l)\hat{\mathbf{A}} + \nu_l \frac{\|\mathbf{A}\|_F}{\|\hat{\mathbf{A}}\|_F + \epsilon}\hat{\mathbf{A}}, \tag{6}$$

Table 1: **Prior methods as fixed points in the EOS search space.** EOS optimizes these dimensions jointly, allowing both corner operators and interior hybrid operators to be discovered.

Method	Structure			Operator Regime					Searchable
	Layers $\mathcal{R}$	Budget c_l	Gate g_l	γ	τ	θ	ρ	ν	
FastV Chen et al. (2024a)	fixed	fixed	fixed	0	–	–	0	–	✗
ToMe Bolya et al. (2022)	fixed/all	uniform	fixed	1	$\to 0$	$\to -1$	0	0	✗
SparseVLM Zhang et al. (2024b)	fixed	schedule	fixed	0	–	–	0	–	✗
VisionZip Yang et al. (2025b)	fixed	fixed	fixed	0	–	–	0	–	✗
PruMerge+ Shang et al. (2025)	fixed	fixed	fixed	mixed	manual	–	0	–	✗
EOS	learned	learned	learned	learned	learned	learned	learned	learned	✓

where ρ_l denotes the anchor-reweighting strength, ν_l denotes the norm-preservation coefficient, and $\mathbf{s}_{\text{imp}} \in \mathbb{R}^K$ denotes the anchor-level importance induced by reduction candidates. Here, γ_l controls explicit feature transfer, while ρ_l controls indirect anchor rescaling. In practice, $\mathbf{s}_{\text{imp}}$ is given by

$$\mathbf{s}_{\text{imp}} = \text{softmax}_{\text{anchor}}(\mathbf{S})^\top \text{softmax}_{\text{candidate}}(\mathbf{q}_{\mathcal{D}}), \tag{7}$$

where $\mathbf{q}_{\mathcal{D}}$ denotes the importance scores of the reduction candidates, and $\text{softmax}_{\text{candidate}}(\cdot)$ normalizes over the reduction-candidate dimension.

Definition 2.1 (Unified efficient operator)**.** *At layer l, an efficient reduction operator is defined by the parameter tuple*

$$\Omega_l = (\gamma_l, \tau_l, \theta_l, \rho_l, \nu_l), \tag{8}$$

and maps the visual hidden states from $\mathbf{X}^{(l)} \in \mathbb{R}^{N_l \times d}$ to a reduced representation $\hat{\mathbf{A}}^{(l)} \in \mathbb{R}^{K_l \times d}$ through Eqs. 2–6. The parameters γ_l and τ_l determine the dominant reduction regime, while θ_l, ρ_l, and ν_l provide token-level gating, anchor reweighting, and scale preservation.

The parameter tuple Ω_l defines the reduction behavior at a single layer. EOS further couples this operator regime with layer activation and token budget, yielding a searchable space over where to compress, how much to compress, and how to process the reduced tokens.

2.3 Efficient Search Space

EOS searches over three coupled dimensions: layer activation, token budget, and operator regime. For all decoder layers, the complete search space is given by

$$\boldsymbol{\Theta} = \left\{ \underbrace{g_l}_{\text{which layers}}, \underbrace{c_l}_{\text{how much}}, \underbrace{\gamma_l, \tau_l, \theta_l, \rho_l, \nu_l}_{\text{which operator } \Omega_l} \right\}_{l=0}^{L-1} \tag{9}$$

where g_l determines whether layer l applies token reduction, c_l controls the fraction of visual tokens to discard, and $(\gamma_l, \tau_l, \theta_l, \rho_l, \nu_l)$ specify the operator regime. All variables are represented by unconstrained learnable parameters and mapped to valid ranges through differentiable transformations:

$$\begin{aligned} g_l &= \sigma(w_l^g \cdot T_g), & c_l &= \sigma(w_l^c \cdot T_c), & \gamma_l &= \sigma(w_l^\gamma), \\ \tau_l &= \text{softplus}(w_l^\tau) + \epsilon, & \theta_l &= 2\sigma(w_l^\theta) - 1, & \rho_l &= \sigma(w_l^\rho), & \nu_l &= \sigma(w_l^\nu), \end{aligned} \tag{10}$$

where T_g and T_c are optional scaling temperatures used to sharpen the structural gates, and the same numerical constant $\epsilon > 0$ ensures a strictly positive assignment temperature. The effective number of retained visual tokens at layer l is given by

$$K_l = \max\left(K_{\min}, \lfloor (1 - g_l c_l) N_l \rfloor\right). \tag{11}$$

When g_l is close to zero, the layer is effectively inactive and the visual sequence is passed to the next layer unchanged. When g_l is active, c_l determines the compression strength, and the operator tuple Ω_l determines how reduction candidates influence retained anchors. Although Eq. 11 defines the discrete forward retain number, EOS optimizes its underlying variables through the relaxation introduced in Sec. 2.4. The structural variables (g_l, c_l) answer *where* and *how much* to compress, while the operator variables $(\gamma_l, \tau_l, \theta_l, \rho_l, \nu_l)$ answer *which reduction behavior* to apply. Such factorization enables EOS to recover existing hand-designed methods by fixing part of $\boldsymbol{\Theta}$, while also allowing new hybrid configurations to be discovered through continuous optimization.

2.4 Differentiable Token Selection

Since top-K_l anchor selection is discrete, we use a straight-through soft boundary relaxation to preserve the forward behavior while allowing gradients to update the retention budget. Let $q_{(K_l)}$ denote the detached K_l-th largest importance score, and define

$$p_i^{\text{keep}} = \sigma\left(\alpha(q_i - q_{(K_l)})\right) \cdot \frac{1 - g_l c_l}{1 - \text{sg}(g_l c_l)}, \tag{12}$$

where α controls the boundary sharpness and $\text{sg}(\cdot)$ denotes the stop-gradient operation. The first term approximates whether token i lies above the selection threshold, while the second term provides a straight-through gradient path from g_l and c_l without changing the forward value. The relaxed keep probabilities estimate the differentiable token count, while inference still uses hard top-K anchors and reduction candidates in Eq. 5.

2.5 Differentiable Search Policy

The backbone multimodal foundation model is kept frozen, and only the search parameters $\mathbf{\Theta}$ are optimized. Given an input-output pair $(\mathbf{x}, \mathbf{y})$, EOS minimizes the task loss under one-sided efficiency constraints:

$$\min_{\mathbf{\Theta}} \quad \underbrace{\mathcal{L}_{\text{task}}(\mathbf{\Theta})}_{\text{task loss}} + \underbrace{\lambda_b \left[\frac{N_{\text{final}}(\mathbf{\Theta})}{B} - 1\right]_+^2}_{\text{token budget: } N_{\text{final}} \le B} + \underbrace{\lambda_c \left[\sum_{l=0}^{L-1} g_l - C\right]_+^2}_{\text{inference cost: active layers} \le C} \tag{13}$$

where $[\cdot]_+ = \max(0, \cdot)$, B is the target final token budget, and C is the target maximum number of active reduction layers. Here, $N_{\text{final}}(\mathbf{\Theta})$ is estimated by $\bar{N}_0 = N_0$, $\bar{N}_{l+1} = (1 - g_l c_l)\bar{N}_l$, and $N_{\text{final}}(\mathbf{\Theta}) = \bar{N}_L$. At deployment, it is replaced by the corresponding discrete retain count after top-K selection. The first regularizer penalizes configurations that exceed the token budget, while the second penalizes configurations that activate too many reduction layers. Both constraints are one-sided: no penalty is applied once the configuration satisfies the desired budget. This design allows the search to prioritize task performance within a feasible efficiency region, rather than forcing unnecessary compression when the budget has already been met. The task loss is given by

$$\mathcal{L}_{\text{task}}(\mathbf{\Theta}) = \underbrace{\mathcal{L}_{\text{CE}}\left(\mathbf{y}, f_{\text{MFM}}(\mathbf{x}; \mathbf{\Theta})\right)}_{\text{text prediction}} + \underbrace{\lambda_a \frac{1}{|\mathcal{A}|} \sum_{l \in \mathcal{A}} \left\| \mathbf{H}_l^{\text{red}}[\text{text}] - \text{sg}\left(\mathbf{H}_l^{\text{full}}[\text{text}]\right) \right\|_2^2}_{\mathcal{L}_{\text{align}}\text{: hidden-state alignment}} \tag{14}$$

where $\mathcal{L}_{\text{CE}}$ is the standard language-modeling loss, $\mathbf{H}_l^{\text{red}}$ denotes the hidden states produced by the reduced forward pass, and $\mathbf{H}_l^{\text{full}}$ is obtained from a reference forward pass without token reduction. The operator $\text{sg}(\cdot)$ blocks gradients through the full-model reference states, so the alignment term only updates the search parameters through the reduced forward pass. The set $\mathcal{A}$ contains selected decoder layers for measuring the deviation between reduced and full computation. The hidden-state alignment term is important because cross-entropy mainly evaluates the final textual prediction and may be insensitive to whether discarded visual information is removed, transferred, or softly redistributed. The constraints in Eq. 13 are one-sided: they penalize only budget violations and vanish once the target efficiency region is reached, allowing the search to focus on preserving task accuracy.

Table 2: One-sided constraints.

Constraint	Active	Inactive
Token budget	$N_{\text{final}} > B$	$N_{\text{final}} \le B$
Active layers	$\sum_l g_l > C$	$\sum_l g_l \le C$

2.6 Optimization and Deployment

We initialize the search from a known efficient configuration by activating a small number of reduction layers and setting γ_l close to zero, corresponding to a pruning-like regime. This warm start provides a stable accuracy-efficiency point, after which gradient descent adjusts layer gates, token budgets, and operator parameters under Eq. 13. After optimization, the learned variables are converted into a deterministic inference configuration: layers with large g_l become active reduction layers, c_l determines the retain schedule, and the operator parameters define the final reduction behavior. The

resulting configuration $\mathbf{\Theta}^\star$ is fixed during evaluation. For different retain budgets, we keep the learned active layers and operator regime fixed, and only rescale the final retain schedule. Thus, EOS does not require retraining a separate reducer for every budget, but learns a general operator configuration deployable under multiple efficiency settings.

2.7 Corner Operators as Special Cases

After defining the search space and optimization policy, we now discuss its relation to existing hand-designed reduction operators. The key property of EOS is that its operator regime Ω_l admits several standard operators as limiting cases. This establishes that pruning, merging, pooling, and adaptive reweighting are not independent mechanisms, but different operating regimes of the same parameterized operator family.

Definition 2.2 (Pure pruning). *Pure pruning is recovered by setting*

$$\gamma_l = 0, \qquad \rho_l = 0. \tag{15}$$

In this case, Eq. 5 reduces to

$$\tilde{\mathbf{A}} = \mathbf{A} + 0 \cdot \mathbf{W}^\top (\mathbf{D} \odot \mathbf{m}) = \mathbf{A}. \tag{16}$$

Thus, the reduction candidates do not contribute to the retained representation and are directly removed from subsequent computation.

Definition 2.3 (Hard token merging). *Hard token merging is recovered by setting*

$$\gamma_l = 1, \quad \tau_l \to 0^+, \quad \theta_l \to -1, \quad \rho_l = 0, \quad \nu_l = 0. \tag{17}$$

The setting $\theta_l \to -1$ makes $m_i \to 1$, disabling the transfer gate. Since $\mathbf{W} = \mathrm{softmax}_{\mathrm{anchor}}(\mathbf{S}/\tau_l)$, the limiting case $\tau_l \to 0^+$ yields

$$W_{ij} \to \mathbf{1}\left[j = \arg\max_k S_{ik}\right], \qquad \hat{\mathbf{a}}_j = \mathbf{a}_j + \sum_{i:\, j = \arg\max_k S_{ik}} \mathbf{d}_i. \tag{18}$$

Therefore, each reduction candidate is transferred to its nearest anchor, which corresponds to a hard nearest-neighbor additive merge.

Definition 2.4 (Uniform pooling). *Uniform pooling is recovered by setting*

$$\gamma_l = 1, \quad \tau_l \to \infty, \quad \theta_l \to -1, \quad \rho_l = 0, \quad \nu_l = 0. \tag{19}$$

The setting $\theta_l \to -1$ makes $m_i \to 1$, disabling the transfer gate. In this limiting case, the assignment distribution becomes uniform:

$$W_{ij} = \frac{\exp(S_{ij}/\tau_l)}{\sum_{k=1}^{K} \exp(S_{ik}/\tau_l)} \to \frac{1}{K}, \qquad \hat{\mathbf{a}}_j = \mathbf{a}_j + \frac{1}{K} \sum_{i=1}^{M} \mathbf{d}_i. \tag{20}$$

Therefore, every anchor receives the same averaged contribution from the reduction candidates, corresponding to uniformly pooling discarded information into the retained anchors.

Definition 2.5 (Adaptive reweighting). *Adaptive reweighting is recovered by setting*

$$\gamma_l = 0, \qquad \rho_l > 0, \qquad \nu_l = 0. \tag{21}$$

Since $\gamma_l = 0$, no reduction candidate is explicitly transferred to the anchors through Eq. 5. The output is instead given by the reweighting term in Eq. 6:

$$\hat{\mathbf{a}}_j = (1 + \rho_l \, s_{\mathrm{imp},j}) \, \mathbf{a}_j. \tag{22}$$

Thus, reduction candidates affect the retained representation only through anchor-level importance rescaling, without explicit token transfer.

Property 2.1 (Operator unification). *The unified operator in Definition 2.1 admits pure pruning, hard merging, uniform pooling, and adaptive reweighting as special cases under different parameter settings of Ω_l.*

The four corner operators arise from different regimes of Ω_l, and Table 1 places representative baselines into the proposed search space. Existing methods can be interpreted as manually specified points in $\mathbf{\Theta}$, since they typically fix the active layers, token budgets, or operator regimes by design. EOS instead makes these dimensions searchable, thereby generalizing representative reduction behaviors and enabling intermediate hybrid operators beyond the manually selected corners.

Table 3: **Empirical verification.** Interestingly, our unified operator reproduces existing token-reduction methods using special parameter values, and further improves performance after search.

Method	Search-space setting	POPE	SQA	MME	GQA	TextVQA	Match?
Original code implementations:							
SparseVLM+ (original code)	—	85.79	64.80	1869	60.95	45.25	—
SparseVLM v1 (original code)	—	85.22	64.15	1825	59.45	44.17	—
Special cases of our unified operator:							
Ours ≡ SparseVLM+	γ=0, $\mathcal{R}$={2, 6, 15}, c_l=V2 sched.	85.96	65.20	1853	60.13	43.86	≈ ✓
Ours ≡ SparseVLM v1	γ=0, $\mathcal{R}$={2, 6, 15}, c_l=V1 sched.	85.22	64.15	1825	59.45	44.17	**Exact ✓**
Searched operators in our unified space:							
Ours (searched, layers fixed)	γ_6=0.08, τ_6=0.22, rest γ=0	85.61	**65.05**	1867	60.81	44.87	≈ V2
Ours (searched)	γ_6=0.08, τ_6=0.22	**85.78**	64.85	**1876**	**60.92**	**45.29**	**Beats V2 ✓**

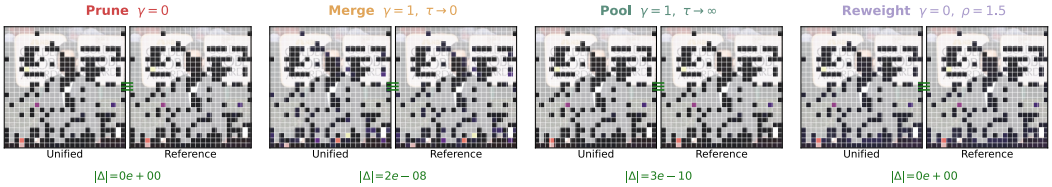


Figure 3: **Numerical verification of corner operators.** Our unified operator reproduces PRUNE, MERGE, POOL, and REWEIGHT under their corresponding settings in Ω_l.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Model and baselines. We evaluate EOS on frozen LLaVA Liu et al. (2023) and compare it with representative corner operators, including pruning-based SPARSEVLM-V1/V2 Zhang et al. (2024b), merging-based TOME Bolya et al. (2022), and pooling-based POOL. All methods use the same retained visual-token budgets for fair comparison.

Benchmarks. We evaluate on representative multimodal benchmarks. Following previous baselines, the prior benchmark group includes POPE Li et al. (2023c), SQA Lu et al. (2022), MME Fu et al. (2023), GQA Hudson & Manning (2019), TextVQA Singh et al. (2019), and SEED Li et al. (2023a). To provide a broader evaluation, the extended benchmark group includes MMStar Chen et al. (2024b), RealWorldQA (RWQA), AI2D Kembhavi et al. (2016), OCRBench Liu et al. (2024b), ChartQA Masry et al. (2022), and MMBench-en Liu et al. (2024a). For space, OCRBench and ChartQA are reported in the appendix together with the complete efficiency measurements.

Evaluation protocol. We report results across four retained-token budgets: $r \in \{192, 128, 64, 16\}$, corresponding to progressively stronger visual-token reduction. For TOME and POOL, we use the same reducer-layer placement $\mathcal{R} = \{2, 6, 15\}$ for controlled comparison with pruning-based baselines. EOS uses the searched operator configuration $\mathbf{\Theta}^\star$ and keeps it fixed during evaluation.

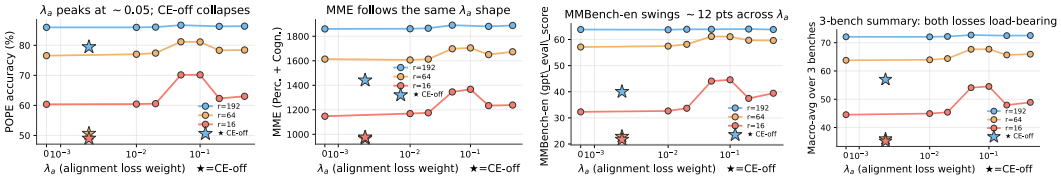
3.2 MAIN RESULTS

Table 4 presents the main comparison across different retained-token budgets. EOS consistently achieves competitive or superior performance across both prior and extended benchmarks. We additionally report prefill computation, latency, and memory usage to directly characterize the efficiency of different reduction strategies.

Analysis. At the moderate retain budgets of $r = 192$ and $r = 128$, EOS performs on par with or slightly better than strong pruning-based baselines, suggesting that the searched operator can recover competitive hand-designed configurations. The advantage becomes clearer as the budget decreases. At $r = 64$, EOS improves POPE from 82.79 to 84.52, MME from 1755 to 1781, GQA from 54.05 to 54.91, and SEED from 57.50 to 59.14 over SparseVLM-v2, while also improving MMStar, RWQA, AI2D, and MMBench-en. Under the most aggressive setting of $r = 16$, EOS achieves the largest gain, substantially increasing POPE from 67.19 to 76.86, MME from 1432 to 1510, GQA from 44.53 to 46.08, and SEED from 44.63 to 47.60. These results indicate that the searched hybrid operator

Table 4: **Main results across retained-token budgets.** EOS is compared with pruning-, merging-, and pooling-based baselines under the same nominal retained-token budgets.

Method	Prior Benchmarks						Extended Benchmarks				Efficiency		
	POPE	SQA	MME	GQA	TextVQA	SEED	MMStar	RWQA	AI2D	MMB-en	TFLOPs↓	Prefill↓	Mem.↓
Retain = 192 visual tokens													
SparseVLM-v1	85.22	64.15	1825	59.41	44.24	64.26	34.97	53.59	54.73	63.92	3.83	55.1	17.47
SparseVLM-v2	**85.79**	64.80	1869	**60.95**	45.25	65.59	33.98	54.77	55.38	63.75	3.84	67.0	17.46
ToMe $\mathcal{R}$={2, 6, 15}	85.09	64.45	1858	59.64	43.25	64.62	33.77	53.73	55.18	63.40	3.84	64.8	13.40
Pool $\mathcal{R}$={2, 6, 15}	52.38	60.54	1287	43.57	8.17	39.06	24.64	43.92	50.36	24.14	3.84	65.3	13.40
EOS	85.78	**64.85**	**1876**	60.92	**45.29**	**65.71**	**34.27**	**55.03**	**55.67**	**63.83**	3.99	66.1	13.41
Retain = 128 visual tokens													
SparseVLM-v1	84.88	64.06	1806	58.38	42.55	63.60	34.77	52.16	54.44	63.66	3.01	52.1	17.43
SparseVLM-v2	85.70	64.85	1845	59.49	**42.52**	**64.72**	34.06	**54.90**	54.63	**63.40**	3.00	61.8	17.42
ToMe $\mathcal{R}$={2, 6, 15}	84.83	**65.49**	1830	57.43	36.51	62.80	34.02	52.16	54.73	62.46	3.00	61.4	13.37
Pool $\mathcal{R}$={2, 6, 15}	43.50	60.14	1150	39.23	7.85	36.16	23.57	43.01	50.23	21.22	3.00	60.8	13.37
EOS	**85.89**	64.85	**1856**	**59.92**	42.33	64.55	**34.77**	**54.90**	**54.95**	63.23	3.16	63.3	13.38
Retain = 64 visual tokens													
SparseVLM-v1	82.58	64.80	1727	53.79	34.21	56.78	31.86	48.50	52.72	59.97	2.17	48.8	17.54
SparseVLM-v2	82.79	65.05	1755	54.05	**32.49**	57.50	31.53	50.98	53.08	60.40	2.17	56.1	17.42
ToMe $\mathcal{R}$={2, 6, 15}	76.77	**65.59**	1622	49.59	22.69	50.45	29.96	45.88	51.68	53.01	2.17	55.1	13.36
Pool $\mathcal{R}$={2, 6, 15}	42.11	60.44	1135	38.95	8.13	35.97	23.36	43.40	50.52	20.70	2.17	55.2	13.36
EOS	**84.52**	65.54	**1781**	**54.91**	31.11	**59.14**	**32.79**	**51.37**	**53.59**	**61.08**	2.39	56.9	13.37
Retain = 16 visual tokens													
SparseVLM-v1	63.28	65.00	1472	44.20	18.36	47.00	28.12	42.61	51.46	41.84	1.92	47.0	17.58
SparseVLM-v2	67.19	**65.64**	1432	44.53	**18.66**	44.63	27.89	44.58	50.65	41.07	1.90	53.7	17.42
ToMe $\mathcal{R}$={2, 6, 15}	43.29	61.28	1150	39.38	8.73	37.15	24.02	43.40	50.29	23.63	1.90	54.8	13.36
Pool $\mathcal{R}$={2, 6, 15}	41.78	60.68	1142	38.86	8.18	35.95	22.98	43.27	50.26	20.62	1.90	53.5	13.36
EOS	**76.86**	64.85	**1510**	**46.08**	16.84	**47.60**	**28.41**	**45.88**	**50.81**	**44.59**	2.13	55.0	13.37



(a) POPE (b) MME (c) MMBench-en (d) Macro average

Figure 4: **Effect of the alignment weight λ_a.** Sweeping λ_a under fixed retained-token budgets shows that EOS is stable across $\lambda_a \in [0.01, 0.5]$. The CE-off variant removes the cross-entropy loss.

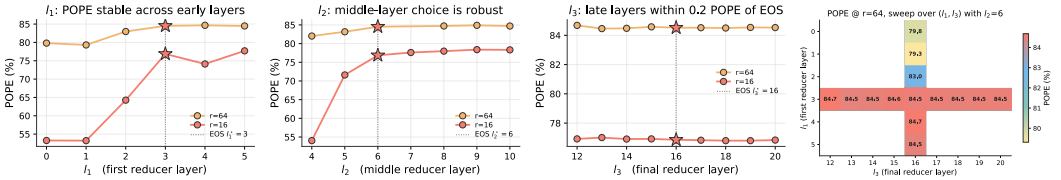
is especially beneficial when the retained-token budget is extremely limited, where fixed pruning, merging, or pooling operators tend to lose critical visual information. The efficiency results show that all reduction methods substantially reduce prefill computation. For EOS, prefill computation decreases from 3.99 TFLOPs at $r = 192$ to 2.13 TFLOPs at $r = 16$, while the measured prefill latency decreases from 66.1 ms to 55.0 ms.

4 Ablation Studies

We analyze EOS from three complementary aspects: (i) the *search policy*, controlled by the hidden-state alignment weight λ_a in Eq. 14; (ii) the *search space*, including the active reducer layers $\mathcal{R}$ and the layer-wise operator regime $\Omega_l = (\gamma_l, \tau_l, \theta_l, \rho_l, \nu_l)$; and (iii) the robustness of the searched configuration $\mathbf{\Theta}^\star$ across different retained-token budgets. All ablations use the same training data, seed, and LLaVA-1.5-7B backbone, and vary only the factor under study.

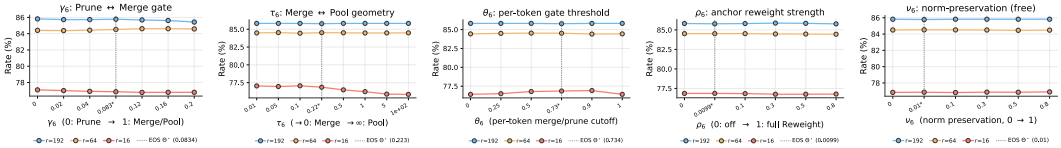
Effect of the Alignment Weight λ_a. The alignment term in Eq. 14 regularizes the reduced forward pass against the unreduced reference model, with weight λ_a controlling the trade-off between text-prediction loss and hidden-state fidelity. We sweep $\lambda_a \in \{0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$ and include a CE-off variant that optimizes alignment alone. Figure 4 shows that EOS is stable across a broad range of λ_a, while removing alignment leads to a clear drop. It reveals that the alignment objective provides a necessary optimization signal for discovering non-trivial operator regimes.

Effect of Active Reducer Layers $\mathcal{R}$. We next study the structural component of the search space, where token reduction is applied. Let $\mathcal{R}^\star = \{l_1, l_2, l_3\}$ denote the active reducer layers extracted from the searched configuration $\mathbf{\Theta}^\star$. We sweep each layer index while keeping the other two fixed,



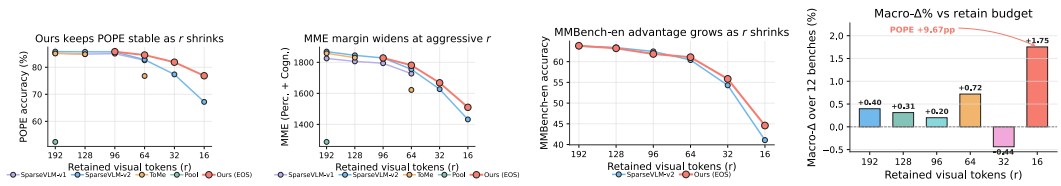
(a) First reducer layer l_1 (b) Middle reducer layer l_2 (c) Final reducer layer l_3 (d) (l_1, l_3) sweep

Figure 5: **Effect of active reducer layers $\mathcal{R}$.** Subfigures (a)–(c) sweep one reducer layer while fixing the others, and subfigure (d) jointly sweeps (l_1, l_3).



(a) γ_6 (b) τ_6 (c) θ_6 (d) ρ_6 (e) ν_6

Figure 6: **Effect of the operator search space $\Omega_6 = (\gamma_6, \tau_6, \theta_6, \rho_6, \nu_6)$.** Each panel sweeps one parameter in the central reducer while fixing the remaining components at $\boldsymbol{\Theta}^\star$.



(a) POPE (b) MME (c) MMBench-en (d) Δ over 12 benchmarks

Figure 7: **Robustness across retained-token budgets.** EOS reuses the same searched configuration $\boldsymbol{\Theta}^\star$ across all budgets. The margin over SparseVLM-v2 increases as r decreases.

and evaluate POPE under retained-token budgets $r \in \{64, 16\}$. Figure 5 reports one-dimensional slices of the layer-placement space and a two-dimensional heatmap over (l_1, l_3) with l_2 fixed. The searched layer placement lies on a high-performing plateau, indicating that EOS does not rely on a cherry-picked layer configuration.

Effect of Operator Regime Ω_l. We study the operator component of the search space by sweeping γ_6, τ_6, θ_6, ρ_6, and ν_6, corresponding to information transfer, assignment temperature, transfer threshold, anchor reweighting, and norm preservation. Figure 6 reports POPE under $r \in \{192, 64, 16\}$. The results show that τ_6 and γ_6 are the most influential dimensions under aggressive compression, while θ_6, ρ_6, and ν_6 mainly provide secondary adjustment. This supports the EOS design, where γ_l and τ_l determine the dominant operating regime and the remaining parameters refine token-level behavior.

Cross-Budget Robustness. We evaluate whether the searched configuration generalizes across different retained-token budgets. Although $\boldsymbol{\Theta}^\star$ is optimized under limited retain budgets, we deploy it unchanged across $r \in \{16, 32, 64, 96, 128, 192\}$. Figure 7 compares EOS with SparseVLM-v1, SparseVLM-v2, ToMe, and Pool on POPE, MME, and MMBench-en, and summarizes the margin over SparseVLM-v2 across all twelve benchmarks. The advantage of EOS becomes more pronounced as the retained-token budget decreases, suggesting that the searched hybrid operator is particularly effective under aggressive token reduction.

5 Conclusion

We presented **Efficient Operator Search** (EOS), a differentiable framework that moves efficient multimodal inference beyond manually designed token-reduction operators. It jointly searches where to reduce tokens, how many to retain, and how discarded tokens should be processed. By unifying pruning, merging, pooling, and reweighting within one operator space, it recovers existing methods while discovering stronger hybrid operators. Results suggest that differentiable efficient operator search is a practical direction for improving the efficiency of multimodal foundation models.

AI USE STATEMENT

We used generative AI tools for language editing and improving the clarity of author-written text. All AI-assisted text was reviewed and revised by the authors, who take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed implementation and evaluation information in the supplementary material, including the training setup, hyperparameters, optimization objective, random seed, software environment, hardware settings, retained-token budgets, and operator configurations. We also provide derivations of the unified operator, additional benchmark results, system-efficiency measurements, and controlled experiments on layer placement and operator regimes. All backbone parameters are kept frozen during the search, and the searched operator configuration is fixed during evaluation unless otherwise stated. The code and configurations will be made publicly available.

REFERENCES

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*, 2022.

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 *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087, 2024b.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6700–6709, 2019.

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

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023a.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023b.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023c.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Xuyang Liu, Ziming Wang, Junjie Chen, Yuhang Han, Yingyao Wang, Jiale Yuan, Jun Song, Siteng Huang, and Honggang Chen. Global compression commander: Plug-and-play inference acceleration for high-resolution large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 7350–7358, 2026.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pp. 216–233. Springer, 2024a.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024b.

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. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2263–2279, 2022.

Xiaohuan Pei, Yanxi Li, Minjing Dong, and Chang Xu. Neural architecture retrieval. *arXiv preprint arXiv:2307.07919*, 2023.

Xiaohuan Pei, Tao Huang, and Chang Xu. Cross-self kv cache pruning for efficient vision-language inference. *arXiv preprint arXiv:2412.04652*, 2024.

Xiaohuan Pei, Yuxing Chen, Siyu Xu, Yunke Wang, Yuheng Shi, and Chang Xu. Action-aware dynamic pruning for efficient vision-language-action manipulation. *arXiv preprint arXiv:2509.22093*, 2025a.

Xiaohuan Pei, Tao Huang, YanXiang Ma, and Chang Xu. Rethinking causal mask attention for vision-language inference. *arXiv preprint arXiv:2505.18605*, 2025b.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22857–22867, 2025.

Yuheng Shi, Xiaohuan Pei, Minjing Dong, and Chang Xu. Catching the details: Self-distilled roi predictors for fine-grained mllm perception. *arXiv preprint arXiv:2509.16944*, 2025.

Yuheng Shi, Xiaohuan Pei, Linfeng Wen, Minjing Dong, and Chang Xu. Q-zoom: Query-aware adaptive perception for efficient multimodal large language models. *arXiv preprint arXiv:2604.06912*, 2026.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8317–8326, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

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

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024a.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024b.