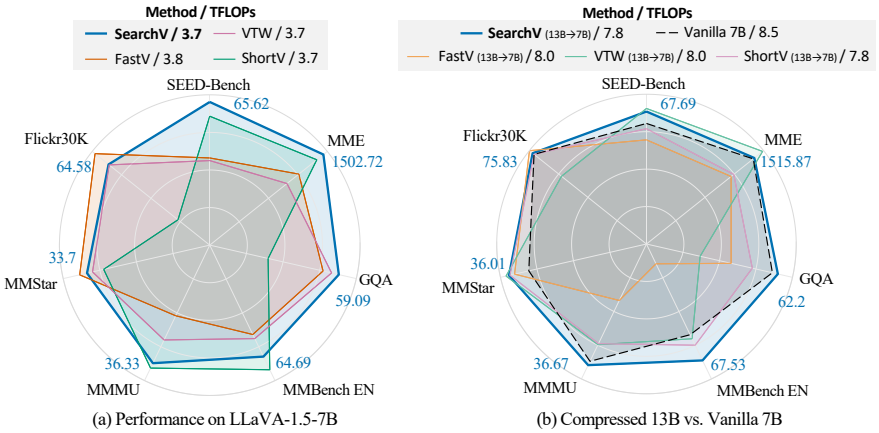
SearchV: Evolutionary Fine-Grained Visual-Token Skipping for Efficient Vision-Language Models

Xinrui Chen[1], **Zhen Huang**[1], **Shuwei Li**[1], **Fanyi Zeng**[1], **Dongxu Yue**[1], **Hao Yu**[1], **Huaisong Zhang**[1], **Xitong Ling**[1], **Yongxian Wei**[1], **Feng Lu**[2]*, **Chun Yuan**[1]*
[1]Shenzhen International Graduate School, Tsinghua University
[2]Faculty of Computer Science and Artificial Intelligence, Shenzhen University of Advanced Technology
cxr22@tsinghua.org.cn, lufeng@suat-sz.edu.cn, yuanc@sz.tsinghua.edu.cn



(a) Performance on LLaVA-1.5-7B (b) Compressed 13B vs. Vanilla 7B

Figure 1: Effectiveness of **SearchV** across model scales. (a) **SearchV** achieves superior balance across all benchmarks on LLaVA-1.5-7B. (b) **SearchV** on LLaVA-1.5-13B is the only method outperforming the Vanilla 7B baseline while maintaining a lower computational cost.

Abstract

Large vision–language models (VLMs) incur substantial computational costs due to massive parameter counts and the deep propagation of long visual sequences. While recent studies highlight visual token redundancy, existing methods often rely on local heuristics or coarse skipping policies that treat Transformer layers as isolated units, thereby limiting the optimization landscape. We propose **SearchV**, a framework that redefines VLM acceleration as a discrete policy search problem guided by a holistic fidelity objective. Central to our approach is Global Contribution (GC) fitness, a metric that captures end-to-end performance drift induced by complete skipping configurations rather than isolated components. By leveraging a fine-grained search space and evolutionary mutation that decouple attention and feed-forward pathways, **SearchV** identifies surgical interventions that preserve multi-benchmark robustness more effectively. Experimental results demonstrate that **SearchV** defines a new efficiency frontier for VLMs. Notably, optimized 13B policies exceed the vanilla 7B average score while using fewer decoder FLOPs, showing that global structural optimization can retain high-capacity accuracy under heavy compression. Code is available at `https://github.com/chenxinrui-tsinghua/SearchV`.

*Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

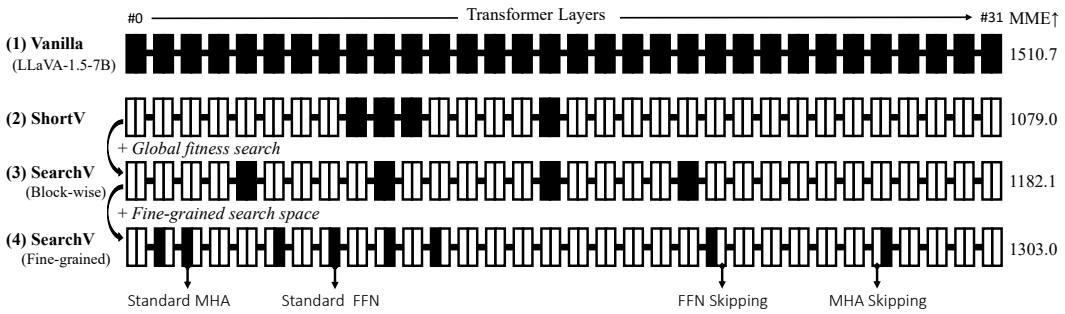


Figure 2: Motivation for global and fine-grained policy search for visual token skipping.

1 Introduction

Large vision–language models (VLMs) [19, 20, 2, 37, 43, 50], including the InternVL family [51, 39], have revolutionized multimodal reasoning by coupling powerful visual perception with generative language backbones. However, their practical deployment is increasingly hindered by the massive computational overhead required to process extensive visual sequences across every Transformer layer of the decoder [29, 30]. Recent evidence suggests that visual computation is largely redundant: visual tokens are under-attended in deeper layers [4, 18], and many layers contribute little when transforming them [46]. This observation has motivated several training-free acceleration techniques designed to prune or freeze visual tokens to reclaim inference efficiency. Prior work, FastV [4], identified the inefficiency of attention computation over visual tokens in the deep layers of VLMs and proposed to prune visual tokens with the lowest attention scores from a selected layer onward. Since unstructured, attention-based token pruning is often fragile and difficult to integrate with KV cache optimizations or attention kernels, recent works have therefore moved from dynamic token pruning toward structured, layer-wise skipping. VTW [18] withdraws all vision tokens after a designated layer. ShortV [46] instead keeps the visual sequence in place and uses Layer Contribution (LC) to find layers whose visual-token updates are ineffective. LC measures the KL divergence between the dense-model logits and those obtained when visual-token updates are frozen at a target layer, and ShortV freezes visual tokens in the selected layers while leaving text computation unchanged.

Despite these empirical gains, existing methods are often constrained by a fragmentary view of model behavior, which limits both the global optimality and the reliability of the resulting token skipping policy. Specifically, current approaches predominantly rely on two highly restrictive and widely overlooked paradigms: *localized importance heuristics* and *coarse-grained action spaces*.

First, a reliance on localized importance heuristics, such as single-layer or per-token importance assessment, ignores the complex non-linear dependencies inherent in Transformer architectures. Because altering a specific layer fundamentally changes the input distribution for all subsequent layers, isolated interventions can trigger cascading errors that degrade global performance [6, 32, 25, 26, 22]. Second, applying uniform, layer-wise skipping rules overlooks the functional heterogeneity between Multi-Head Attention (MHA) and Feed-Forward Networks (FFN). For example, cross-modal grounding may still require MHA even when FFN updates for visual tokens have become redundant. Consequently, such coarse granularity imposes a structural bottleneck on the Pareto efficiency frontier, preventing these methods from reaching the optimal trade-off between cost and performance [31, 49].

We argue that overcoming these limitations necessitates a paradigm shift from local heuristics toward global, fine-grained policy optimization. As illustrated in Figure 2, local assessment masks often suffer from significant fidelity loss because they fail to account for the non-linear impact of cumulative interventions. While introducing a search process at the layer-level partially recovers performance, such a search space remains too restrictive to capture the functional heterogeneity of sub-layer components. In contrast, our proposed framework, **SearchV**, enables independent, per-submodule decisions across the entire model depth. This flexibility allows for the discovery of specialized configurations, thereby unlocking a superior Pareto frontier that reconciles aggressive compute savings with high model fidelity. We summarize our primary contributions as follows:

- We propose **Global Contribution (GC) fitness**, a holistic metric that quantifies the end-to-end performance drift induced by complete visual skipping configurations of VLMs, capturing the non-linear inter-dependencies between layers that local heuristics overlook.
- We design a **fine-grained search space** that decouples the attention mechanism and feed-forward pathways within each Transformer block, enabling surgical, per-submodule skipping decisions to preserve critical functional components while aggressively pruning redundancy.
- We introduce **SearchV**, a training-free VLM compression framework integrating the above via **evolutionary mutation**. It defines a new efficiency frontier, slashing LLaVA-1.5-13B decoder FLOPs by 60% while retaining 97.1% performance, surpassing ShortV by 6.1%.

As visualized in Figure 1, **SearchV** yields a more comprehensive and balanced performance profile, surpassing existing methods in efficiency and cross-benchmark accuracy at the same time.

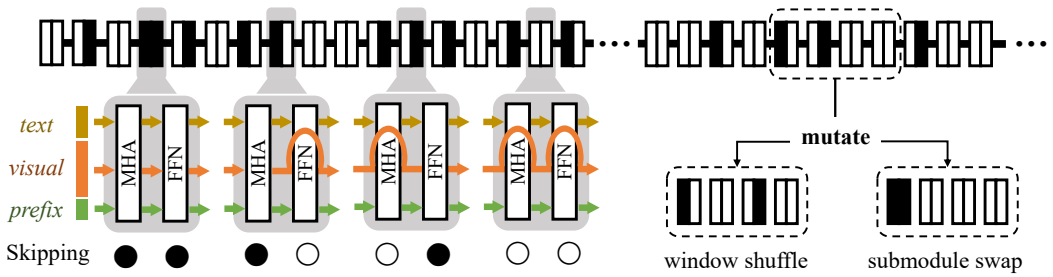
2 Related Work

Vision-Language Models (VLMs). Modern large vision–language models [37, 20, 24, 40, 45] typically pair a pretrained visual encoder [28] with a decoder-only large language model [42, 35]. This architecture maps image patches into a long token sequence that is concatenated with text before joint decoding. Vision Transformer (ViT) backbones [9] and contrastive image–text pretraining [28] supply the visual encoder in most VLMs, while self-supervised objectives such as MAE [11] are widely used in visual pretraining. Subsequent alignment modules [17] map visual features into the LLM embedding space for subsequent language decoding. The paradigm of bootstrapping from frozen encoders alongside LLM capacity was popularized by BLIP-2 [17], which was followed by instruction-tuned series such as LLaVA [21] and its improved variants [19, 20]. These models illustrate how the pursuit of higher resolution and richer visual semantics inevitably inflates token length, creating a significant computational bottleneck in the decoder [4]. Parallel families include early multimodal generalists such as Flamingo [1], which utilizes cross-attention, and recent large-scale stacks like Gemini [33] that emphasize native long-context multimodal understanding. To effectively capture the information within images, VLMs must encode visual data into hundreds or even thousands of tokens [44, 38]. These visual sequences are much more numerous than text tokens and are typically concatenated with text before being processed by the LLM backbone. For example, LLaVA-1.5 [19] converts each 336×336 resolution image into 576 tokens. Similarly, LLaVA-NeXT [20] uses an AnyRes strategy to process images into as many as 2,880 visual tokens. While these architectures advance model capability, they inherit quadratic attention scaling and heavy per-layer feed-forward costs. These challenges have motivated research into efficient kernels [8, 7, 14] as well as acceleration methods designed for VLMs when visual sequences grow long [4, 46, 18].

Redundancy of visual tokens in VLMs. Early research on visual token redundancy focused on spatial redundancy by identifying and removing unimportant visual tokens. CrossGET [30] and LLaVA-PruMerge [29] utilize modal attention or similarity-based merging to condense the visual stream. FastV [4] identifies an inefficient attention phenomenon where visual tokens receive little attention in deep layers, and proposes to prune visual tokens with the lowest attention scores from a selected layer onward. However, such dynamic pruning complicates KV-cache reuse because the kept subset can vary across decoding steps. The pruning layer must also materialize attention scores that FlashAttention [8, 7] does not provide, which can prevent theoretical FLOP reductions from translating into wall-clock speedups. To maintain hardware compatibility, a second generation applies structured layer-wise skipping rather than pruning a changing token subset. Visual Tokens Withdrawal (VTW) [18] identifies information migration, where visual features are absorbed into text tokens in initial layers. This allows visual tokens to be withdrawn from subsequent attention calculations without losing critical information. Similarly, ShortV [46] introduces the Layer Contribution (LC) metric to quantify layer-wise divergence. By identifying ineffective layers, ShortV freezes visual token updates in those layers while leaving text computation intact. This approach maintains KV cache continuity and ensures better compatibility with hardware acceleration techniques.

2.1 Research Gap

Despite these advancements, existing state-of-the-art methods remain limited by:



(a) Search space decoupling (b) Mutation operators

Figure 3: The core of the **SearchV** framework: fine-grained search space and evolutionary operators. (a) The decoupled skipping mechanism for MHA and FFN submodules. (b) Mutation strategies designed for navigating the combinatorial landscape of visual processing policies.

Suboptimality of Localized Heuristics: Current metrics for token and layer skipping (e.g., similarity scores in PruMerge [29] or LC in ShortV [46]) are predominantly restricted to isolated components. Evaluating layers in isolation fails to capture the non-linear coupling across the Transformer stack.

Inflexibility of Coarse-grained Action Spaces: Current methods still treat each Transformer layer as an atomic unit for visual skipping. FastV [4] prunes tokens from a chosen layer onward, VTW [18] withdraws all vision tokens after a designated layer, and ShortV [46] freezes visual updates in LC-selected layers without decoupling MHA from FFN. This coarse granularity misses asymmetric submodule configurations that could further improve the efficiency–accuracy trade-off.

These gaps motivate our **SearchV** framework, which shifts toward a **global fidelity objective** and a **decoupled search space** to uncover more effective surgical intervention policies.

3 Method

To overcome the limitations of localized heuristics and coarse-grained action spaces, **SearchV** redefines VLM acceleration as an end-to-end policy optimization problem. Unlike previous methods that treat Transformer layers as monolithic units, **SearchV** explores a fine-grained, decoupled search space that enables independent skipping decisions for MHA and FFN at each Transformer layer. The core of our **SearchV** framework consists of a fine-grained search space and an evolutionary mutation strategy, as illustrated in Figure 3. By optimizing a Global Contribution (GC) fitness metric, we identify surgical configuration policies $\mathbf{c} \in \mathcal{C}$ that capture the non-linear functional dependencies across the entire model depth. In the following, we formulate the decision variables and the budgeted objective, then define GC fitness and the evolutionary search used in the rest of this section.

3.1 Problem Formulation

We consider policies that alter how visual tokens are processed in the decoder stack of VLMs, while leaving the original computation paths for text tokens unchanged at every layer.

Decision Variables. The decoder consists of L layers indexed $\ell = 1, \ldots, L$. At each layer, we choose a configuration pair $(s_\ell^{\text{mha}}, s_\ell^{\text{ffn}})$ from the finite sets $\mathcal{S}_{\text{mha}} \times \mathcal{S}_{\text{ffn}}$. This specifies how visual tokens participate in MHA and the FFN. A complete configuration is defined as:

$$\mathbf{c} = \{(s_\ell^{\text{mha}}, s_\ell^{\text{ffn}})\}_{\ell=1}^{L} \in \mathcal{C} = \left(\mathcal{S}_{\text{mha}} \times \mathcal{S}_{\text{ffn}}\right)^L.$$

Configurations that freeze or skip visual updates in selected branches instantiate the same physical mechanism as ShortV [46] but allow MHA and FFN to be controlled independently.

Objective. Let $\Phi(\mathbf{c})$ denote the estimated decoder FLOPs under configuration $\mathbf{c}$, let B be the prescribed decoder FLOPs budget, and let $\mathcal{D}_{\text{cal}} = \{\mathbf{x}_i\}_{i=1}^{N}$ be a small calibration set of multimodal inputs used for search. We define a scalar proxy fitness $\text{GC}(\mathbf{c})$ that measures the distributional drift

of the skipping policy $\mathbf{c}$ relative to a full capacity reference run on $\mathcal{D}_{\text{cal}}$. The optimization goal is

$$\min_{\mathbf{c}\in\mathcal{C}} \ \mathrm{GC}(\mathbf{c}) \quad \text{s.t.} \quad \Phi(\mathbf{c}) \le B. \tag{1}$$

Lower values of GC indicate closer agreement with the reference on the calibration distribution.

3.2 Global Contribution (GC) Fitness

Metrics such as per-token saliency or single-layer importance assessment assume that component contributions are additive and weakly coupled. However, Transformer architectures exhibit complex non-linear dependencies. We introduce Global Contribution as a holistic fitness proxy that scores an entire skipping configuration by measuring its distributional drift relative to the original model.

For each input $\mathbf{x}_i$ in the calibration set, the reference run uses the original model to yield logits $\mathbf{z}_i^{\text{ref}}$, while the candidate run applies configuration $\mathbf{c}$ to yield $\mathbf{z}_i^{\mathbf{c}}$. Let $p_i^{\text{ref}} = \mathrm{softmax}(\mathbf{z}_i^{\text{ref}})$ and $p_i^{\mathbf{c}} = \mathrm{softmax}(\mathbf{z}_i^{\mathbf{c}})$ be the next-token distributions at the final context position. We define Global Contribution as the mean forward Kullback-Leibler divergence from the candidate to the reference:

$$\mathrm{GC}(\mathbf{c}) = \frac{1}{N}\sum_{i=1}^{N} D_{\mathrm{KL}}\big(p_i^{\mathbf{c}} \,\big\|\, p_i^{\text{ref}}\big), \qquad D_{\mathrm{KL}}(p\|q) = \sum_v p_v\big(\log p_v - \log q_v\big). \tag{2}$$

GC measures how configuration $\mathbf{c}$ shifts the full-network next-token distribution relative to the teacher under an end-to-end forward pass. This metric captures the joint effect of a complete policy and accounts for inter-layer dependencies across the Transformer stack. The last-position distribution is taken at the final prompt token immediately before generation: after the question and option tokens for multiple-choice tasks, and at the end of the instruction prompt for generative tasks. The dense-model reference is computed once on the calibration set and cached, so its cost is amortized across all subsequent evaluations of candidate configurations throughout evolutionary search.

3.3 Fine-grained Search Space and Budget Constraints

As illustrated in Figure 3(a), we define a decoupled search space where visual tokens can independently bypass the MHA and FFN submodules. We categorize these operations into two distinct modes: full participation and vision skipping. In the vision skipping mode, visual tokens are excluded from the feed-forward updates and the query and output projections of the attention mechanism, as in ShortV [46]. Crucially, visual tokens are retained in the key and value cache to preserve the spatial and semantic context necessary for cross-modal grounding across subsequent decoding steps.

To facilitate evolutionary search under a specific computational limit, we utilize a precise FLOPs accounting model based on the chosen configuration. The total computational cost of the decoder is a linear summation of the costs incurred by individual sub-layer components, which are determined by the effective sequence length processed by each operation. The detailed mapping between the search modes and their corresponding computational complexities is established in Table 1. By formulating the total budget as a linear functional of the submodule decisions, we ensure that the search process can efficiently identify policies that minimize GC within a prescribed FLOPs budget.

Table 1: Per-layer computational cost and effective sequence length mapping.

Component	**Effective Sequence Length S_{eff}**	**FLOPs Approximation**
FFN Updates	$S_{\text{tot}} - (1 - s_\ell^{\text{ffn}})v$	$6 \cdot S_{\text{eff}} \cdot m \cdot h^2$
MHA: K/V Projections	S_{tot}	$4 \cdot S_{\text{tot}} \cdot h^2$
MHA: Q / Output Projections	$S_{\text{tot}} - (1 - s_\ell^{\text{mha}})v$	$4 \cdot S_{\text{eff}} \cdot h^2$
MHA: Attention Core	$S_{\text{tot}} - (1 - s_\ell^{\text{mha}})v$	$4 \cdot S_{\text{eff}} \cdot S_{\text{tot}} \cdot h$

Note: $s_\ell^{\{\cdot\}} \in \{0, 1\}$ toggles between full ($s = 1, S_{\text{eff}} = S_{\text{tot}}$) and vision skipping ($s = 0, S_{\text{eff}} = S_{\text{tot}} - v$) modes. v and S_{tot} denote visual and total token lengths. m and h are expansion factor and hidden width.

3.4 Evolutionary Mutation Operators

The discrete and high-dimensional nature of the configuration space $\mathcal{C}$ necessitates a robust yet conservative variation scheme. To avoid the performance collapse often associated with large-scale

structural rewrites, we design our mutation operators based on two guiding principles: *locality* and *budget neutrality*. First, each mutation perturbs only a localized window of w contiguous layers to facilitate incremental refinement. Second, we implement strict FLOP-preserving transitions, ensuring that every operator preserves the aggregate decoder FLOPs of the parent configuration $\mathbf{c}$.

Specifically, our evolutionary search utilizes two primary operators to navigate the combinatorial landscape, as illustrated in Figure 3(b):

- **Window Shuffle:** This operator randomly permutes the sequence of w configuration pairs $\{(s_\ell^{\text{mha}}, s_\ell^{\text{ffn}})\}_{\ell \in \mathcal{W}}$, where $\mathcal{W}$ denotes the selected mutation window.
- **Submodule Swap:** This operator randomly picks two configuration pairs within the window and swaps only their MHA modes or only their FFN modes.

Crucially, since the total computational cost $\Phi(\mathbf{c})$ is a linear functional of the independent submodule decisions, these operations inherently leave the total FLOPs unchanged.

3.5 SearchV

The **SearchV** framework begins by building an initial pool of I feasible configurations. The first parent is initialized with the layer contribution metrics [46]. We then generate the remaining population by applying mutation operators to copies of this initial parent to reach the pool size I. To minimize computational overhead, we adopt a staged evaluation schedule [31]. Early stages evaluate candidates using few calibration instances to retain only a small survivor set. Later stages increase the sample budget so that expensive scoring is only applied to promising policies. We also implement a fitness cache and a tabu set. The fitness cache avoids redundant evaluations of the same genotype, while the tabu set prevents the proposal of previously explored configurations.

The evolutionary search process is detailed in Algorithm 1. The evolutionary loop proceeds through G generations with a population of P individuals. In each generation, we evaluate individuals using GC, select the top survivors, and generate offspring through tournament selection and local mutations. After the final generation, the best configuration found is returned. The optimization convergence stability of our evolutionary search process is visualized in Figure 4 and discussed in Appendix C.

Algorithm 1 SearchV: evolutionary search under GC

Require: Target budget B; calibration batches and reference logits; initial pool size I, population size P, generations G, offspring count R; mutation window w; initial ranking budget N_0; staged parameters $\{(K_s, N_s)\}_{s=1}^{S}$; tournament size k

$\mathcal{I} \leftarrow$ initial pool of I configurations (initialized parent + mutations)
$\mathcal{P} \leftarrow$ top P configurations selected from $\mathcal{I}$ using N_0 samples
for $g = 1 \dots G$ **do**
  $\mathcal{O} \leftarrow R$ children via tournament selection and mutation from $\mathcal{P}$
  for each stage (K_s, N_s) **do**
    if $s = S$ **then**
      Add parents from $\mathcal{P}$ not in $\mathcal{O}$ for elitism
    end if
    Re-rank $\mathcal{O}$ with N_s samples and retain top K_s
  end for
  $\mathcal{P} \leftarrow \mathcal{O}$
end for
return configuration in $\mathcal{P}$ with best fitness

4 Experiments

4.1 Experimental Setup

Models. We evaluate **SearchV** on prevalent models including LLaVA-1.5-7B, LLaVA-1.5-13B [19], LLaVA-NeXT-7B, LLaVA-NeXT-13B [20], Qwen2.5-VL-7B-Instruct [3], Gemma-3-4B-it [34], and InternVL3-8B [51]. We keep the weights frozen because our compression approach is training-free. This follows the standard protocol established in previous research on visual token efficiency.

Table 2: Comparison of **SearchV** and competitive baselines across various VLM backbones. Benchmarks include MME (MME perception), MMB (MMBench-EN), MMMU (val), MMStar, SEED (SEED-Bench), GQA, and Flickr (Flickr30K CIDEr). In each model group, the best performance among efficiency-oriented methods is highlighted in **bold**.

Method	TFLOPs	Ratio	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
LLaVA-1.5-7B											
Vanilla	8.5	100%	1510.7	64.1	36.3	33.7	66.1	61.9	74.9	58.9	100%
FastV (K=2, R=65%)	3.8	45%	1394.1	59.8	31.8	**34.1**	62.6	57.4	**73.2**	55.5	94.3%
VTW (K=12)	3.7	44%	1473.6	**67.6**	**36.8**	32.8	64.9	51.4	20.3	49.6	84.3%
ShortV (N=24)	3.7	44%	1341.7	60.7	34.1	33.4	62.5	58.3	64.2	54.3	92.2%
SearchV	3.7	44%	**1502.7**	64.7	36.3	33.7	**65.6**	**59.1**	64.6	**57.0**	**96.8%**
LLaVA-1.5-13B											
Vanilla	16.6	100%	1531.3	68.9	35.4	36.1	68.2	63.3	79.6	61.2	100%
FastV (K=2, R=70%)	6.6	40%	1364.3	53.7	30.3	34.7	63.3	58.8	**76.9**	55.1	90.1%
VTW (K=14)	6.9	42%	**1538.5**	**68.1**	34.4	35.8	**67.6**	53.8	36.4	53.3	87.1%
ShortV (N=32)	6.6	40%	1298.8	64.5	33.6	36.0	63.2	59.3	68.4	55.7	91.0%
SearchV	6.6	40%	1480.1	68.0	**36.0**	**36.7**	67.5	**61.3**	72.4	**59.4**	**97.1%**
LLaVA-NeXT-7B											
Vanilla	42.7	100%	1519.0	67.1	36.4	37.1	70.2	64.1	69.7	60.1	100%
FastV (K=2, R=65%)	16.2	38%	1407.4	56.7	32.7	35.6	66.4	**62.1**	61.8	55.1	91.6%
VTW (K=12)	16.5	39%	1473.4	**66.2**	34.9	36.5	68.3	50.9	10.3	48.7	81.0%
ShortV (N=24)	16.0	37%	**1504.1**	65.4	35.1	**37.7**	**70.2**	55.8	38.3	54.0	89.8%
SearchV	16.0	37%	1501.9	66.0	**36.6**	36.1	69.4	61.7	**64.4**	**58.5**	**97.3%**
Qwen2.5-VL-7B-Instruct											
Vanilla	8.5	100%	1646.5	82.8	50.2	60.8	74.3	59.2	33.0	63.2	100%
FastV (K=2, R=60%)	4.2	49%	1539.2	79.9	**48.9**	52.3	64.9	53.9	**32.7**	58.5	92.5%
VTW (K=13)	4.4	52%	1357.3	70.1	46.9	44.4	60.0	42.2	11.9	49.0	77.6%
ShortV (N=16)	4.2	49%	1583.5	79.3	44.9	54.2	73.2	56.8	30.4	59.7	94.4%
SearchV	4.2	49%	**1620.3**	**80.1**	46.9	**55.9**	**73.7**	**56.9**	31.5	**60.9**	**96.3%**
Gemma-3-4B-it											
Vanilla	2.1	100%	1636.0	68.0	40.2	36.8	65.6	40.6	3.7	48.1	100%
VTW (K=12)	1.0	48%	1431.5	64.5	39.0	35.2	62.7	32.8	0.9	43.8	91.1%
ShortV (N=24)	1.0	48%	1533.3	**65.5**	37.1	**35.3**	63.8	40.5	2.6	45.9	95.4%
SearchV	1.0	48%	**1542.2**	64.6	**40.7**	35.1	**64.4**	**40.7**	**3.7**	**46.6**	**96.9%**
InternVL3-8B											
Vanilla	48.9	100%	2355.8	84.1	54.0	66.2	77.6	62.5	86.7	78.4	100%
VTW (K=14)	24.9	51%	1854.8	69.2	51.2	45.9	60.7	43.2	34.1	56.7	72.3%
ShortV (N=14)	25.2	52%	2055.8	71.3	48.3	56.5	62.2	44.5	33.7	59.9	76.4%
SearchV	24.6	50%	**2129.5**	**81.6**	**52.3**	**60.2**	**76.1**	**60.2**	**79.3**	**73.7**	**94.0%**

Search settings. **SearchV** optimizes GC fitness using a calibration set of 64 samples per task across five benchmarks. We employ a staged evaluation strategy to balance speed and accuracy: the initial generation is ranked with $N_0 = 64$ samples, while subsequent generations filter $R = 32$ offspring through a rough screening stage ($K_1 = 8, N_1 = 32$) and a final re-ranking stage ($K_2 = 3, N_2 = 64$). The search maintains a population of 3, using a tournament size of 3 and mutation window of $w = 4$. Optimization terminates after 20 generations or a patience of 3. We reduce the total execution time by using a fitness cache and a tabu set to avoid redundant evaluations of previously seen configurations.

Evaluation. To validate the effectiveness of **SearchV**, we export the best searched skipping policy and evaluate it using the LMMs-Eval framework [15, 48]. We test the models on seven benchmarks, which are GQA [12], MMMU [47], MME [10], SEED-Bench [16], Flickr30K [27], MMStar [5], and MMBench [23]. We maintain a constant text length of 64 tokens across all evaluations. The visual token counts are set to 576 for LLaVA-1.5 and Qwen2.5-VL, 2880 for LLaVA-NeXT, 256 for Gemma-3-4B-it, and 3328 for InternVL3-8B, with the LLaVA-NeXT count matching the AnyRes upper bound used for FLOPs accounting. Our two-stage evaluation for offspring screening consists of a quick pass on half of the calibration set followed by a full pass on the entire set. We evaluate models in FP16 precision using NVIDIA L20 GPUs and RTX 3090 GPUs.

4.2 Main Results

The comparative performance of **SearchV** against several state-of-the-art visual token efficiency methods is detailed in Table 2. Across six vision-language models spanning the LLaVA, Qwen, Gemma, and InternVL families, our approach demonstrates a superior capacity to balance computational reduction with benchmark accuracy. Evaluation results indicate that **SearchV** maintains high faithfulness to original dense models while substantially lowering computational overhead. On the

Table 3: Multi-image benchmark results. MMIE is scored on a 0–6 scale.

Model	Method	Ratio	MUIR-Bench	MMIE	MMR
LLaVA-1.5-7B	Vanilla	100%	40.8	3.9	52.9
	ShortV (N=24)	44%	4.0	2.5	28.2
	SearchV	44%	**24.4**	**3.2**	**40.2**
LLaVA-1.5-13B	Vanilla	100%	39.2	3.8	71.8
	ShortV (N=28)	47%	6.2	3.2	49.3
	SearchV	47%	**17.7**	**3.6**	**60.2**
InternVL3-8B	Vanilla	100%	36.2	3.7	73.5
	ShortV (N=14)	52%	32.7	2.2	53.5
	SearchV	50%	**35.4**	**3.6**	**71.4**

LLaVA-1.5-7B model, for instance, our method preserves 96.8% of the vanilla performance using only 44% of the original TFLOPs. This efficiency scales effectively to larger architectures. In the LLaVA-1.5-13B setting, **SearchV** achieves an average score of 59.4, retaining 97.1% of vanilla accuracy despite a 60% reduction in TFLOPs. Such results suggest that the evolutionary search strategy accurately identifies redundant submodules that can be bypassed without degrading the fundamental reasoning capabilities of the underlying models. Beyond theoretical FLOP reduction, **SearchV** offers significant advantages in hardware compatibility and practical optimization. Attention-based token pruning as in FastV yields theoretical savings, but the kept visual subset can change across decoding steps and the pruning layer must materialize attention scores. This complicates standard KV-cache reuse and FlashAttention-style kernels, and can prevent theoretical FLOP reductions from translating into wall-clock speedups. In contrast, **SearchV** skips visual updates at the MHA/FFN submodule level, leaving a contiguous K/V layout that integrates with common hardware acceleration libraries.

The robustness of **SearchV** is further evidenced by its consistent performance across diverse benchmarks, particularly in tasks requiring fine-grained visual localization where existing efficiency-oriented methods often struggle. While alternative strategies frequently exhibit performance fluctuations depending on the specific task, **SearchV** maintains stability across the entire evaluation suite. For example, our method sustains a competitive score of 64.7 on MMBench for LLaVA-1.5-7B at 44% FLOPs while delivering top-tier results on perception-heavy MME and GQA. This reliability is rooted in our Global Contribution objective and evolutionary search strategy, which together capture the holistic predictive behavior of the model. The same training-free procedure transfers beyond LLaVA and Qwen. On Gemma-3-4B-it at 48% FLOPs, **SearchV** retains 96.9% of vanilla performance, outperforming ShortV by 1.5% and VTW by 5.8%. The margin is larger on InternVL3-8B at a matched $\sim$50% budget: **SearchV** retains 94.0%, versus 76.4% for ShortV and 72.3% for VTW. The gap is most pronounced on perception-heavy and localization-sensitive benchmarks (MME, MMBench, SEED, and Flickr30K), where coarse layer-wise skipping collapses while the decoupled policy preserves cross-modal grounding. Additional results appear in Appendix A (Table 9); the frontier under varying FLOPs ratios is analyzed in Appendix B (Table 10).

Multi-image evaluation. **SearchV** skips visual computation at the MHA/FFN submodule level and is therefore agnostic to the number of input images: Global Contribution, FLOP-preserving mutations, and staged evaluation transfer without algorithmic changes. Table 3 reports results on MUIR-Bench [36], MMIE [41], and MMR [13]. Coarse layer-wise skipping is particularly brittle on multi-image reasoning. On LLaVA-1.5-7B, ShortV drops to 4.0% on MUIR-Bench and 28.2% on MMR, while **SearchV** recovers a large fraction of the lost accuracy and reaches 24.4% and 40.2%. On InternVL3-8B, the searched policy nearly matches the dense model, with 35.4% versus 36.2% on MUIR-Bench and 71.4% versus 73.5% on MMR. These results confirm that the searched fine-grained policy generalizes to multi-image inputs without changing the search algorithm.

Speedup. Table 4 evaluates the acceleration and performance retention of **SearchV** relative to the ShortV baseline across different hardware architectures. Our method consistently translates theoretical FLOP reductions into significant inference speedups while preserving superior model fidelity. On the LLaVA-1.5-7B backbone at a 34% FLOPs ratio, **SearchV** achieves a 1.91$\times$ speedup on an RTX 3090 with 88.8% performance retention, surpassing ShortV by 7.6% in retention. The efficacy of our fine-grained evolutionary search is most pronounced in extreme compression regimes.

Notably, on LLaVA-NeXT-7B at a 27% FLOPs ratio, the performance of ShortV collapses to 21.2%, whereas **SearchV** maintains a robust 86.3% and reaches approximately 2.0× speedup.

Table 4: Inference speedup and performance comparison against the ShortV baseline on NVIDIA RTX 3090 and L20 GPUs. Performance refers to the average retention relative to the vanilla model.

Model	TFLOPs	FLOPs Ratio	Speedup		Performance (%)	
			RTX 3090	L20	ShortV	SearchV
LLaVA-1.5-7B	8.5	100%	1.00×	1.00×	100	100
	3.7	44%	1.68×	1.63×	92.2	**96.8**
	2.9	34%	1.91×	1.82×	81.2	**88.8**
LLaVA-NeXT-7B	42.7	100%	1.00×	1.00×	100	100
	16.0	37%	1.75×	1.76×	89.8	**97.3**
	11.6	27%	1.99×	2.02×	21.2	**86.3**

Search Cost. Table 5 shows that the evolutionary search is inexpensive on a single GPU. LLaVA-1.5-7B finishes in 1.66–2.89 hours depending on the target FLOPs ratio, and LLaVA-NeXT-7B remains under 4 hours (2.07–3.61h) despite its longer visual sequence. Peak memory stays between 13.5 and 14.5 GB for every target FLOPs ratio reported in the single-GPU runs.

Table 5: Search cost analysis including time and memory consumption across generations.

Model	TFLOPs	FLOPs Ratio	Time/Gen (min)	GPU Mem. (GB)	Total Gens	Total Time (hours)
LLaVA-1.5-7B	3.7	44%	9.96	13.58	10	1.66
	2.9	34%	8.67	13.58	20	2.89
LLaVA-NeXT-7B	16.0	37%	24.80	14.52	5	2.07
	11.6	27%	24.09	14.52	9	3.61

4.3 Ablations

We evaluate the core components of **SearchV** regarding performance and efficiency, focusing on module and mutation insights. For extended technical analysis, we refer readers to the Appendix for discussions on fitness objectives (Appendix D, Table 11), calibration set sensitivity (Appendix F, Table 14), search initialization strategies (Appendix E, Tables 13 and 12), multi-seed stability (Appendix G, Table 15), expanded component ablations (Appendix H, Table 16), skipping mechanics (Appendix I), and task-level failure modes (Appendix J, Table 17).

Component Efficacy. To evaluate the fine-grained search space and Global Contribution objective, we ablate **SearchV** variants on LLaVA-1.5-7B at 34% FLOPs.

Table 6 shows that global optimization significantly outperforms the LC-selected ShortV baseline (81.2% retention). While layer-level search improves retention by 4.0%, our full decoupled sub-layer framework achieves a superior 88.8% performance retention at the same budget as ShortV.

Table 6: Ablation study of **SearchV** components on LLaVA-1.5-7B. We fix the FLOPs ratio to 34% to evaluate different search granularities across seven benchmarks. "Per." denotes the average performance retention relative to the Vanilla model.

Method	Ratio	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
Vanilla	100%	1510.7	64.1	36.3	33.7	66.1	61.9	74.9	58.9	100%
ShortV	34%	1079.0	57.9	31.0	30.2	56.0	52.0	53.6	47.8	81.2%
ShortV w/ Search	34%	1182.1	60.6	**33.8**	30.9	57.7	52.2	**57.2**	50.2	85.2%
SearchV (Ours)	34%	**1303.0**	**61.8**	33.3	**33.7**	**60.7**	**55.3**	56.2	**52.3**	**88.8%**

Mutation Window Size. The mutation window size w defines the range of structural changes during search. Based on Table 7, we evaluate $w \in \{2, 4, 6\}$ and find $w = 4$ yields the highest average score

by balancing policy refinement with exploration. Specifically, $w = 2$ limits the neighborhood of each mutation, underperforming on reasoning tasks like MMStar. Conversely, while $w = 6$ maximizes MME to 1534.4, it degrades overall generalization. This indicates that excessive mutations may disrupt the synergy between vision-language sub-modules, favoring isolated recognition gains over robustness across the remaining suite of perception and reasoning benchmarks.

Table 7: Ablation of mutation window length w on LLaVA-1.5-7B at a 44% FLOPs ratio. A window size of $w = 4$ achieves the most robust performance across diverse benchmarks.

Window	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.
$w = 2$	1516.8	**64.7**	36.2	33.0	65.1	58.8	63.6	56.7
$w = 4$	1502.7	**64.7**	**36.3**	**33.7**	65.6	**59.1**	**64.6**	**57.0**
$w = 6$	**1534.4**	64.0	36.1	33.6	**65.7**	58.4	63.8	56.9

Mutation Operators. Window Shuffle exploits layer locality by permuting configurations inside a contiguous depth window, while Submodule Swap exchanges only MHA or only FFN decisions across layers. Table 8 isolates the two operators on LLaVA-1.5-7B at 34% FLOPs. Submodule Swap alone retains 88.4%, only 0.4% below the full combination, whereas Window Shuffle alone drops to 85.2%. The 3.2% gap indicates that exchanging heterogeneous MHA/FFN decisions is more effective than layer-wise reordering. Combining both operators remains best because they explore complementary neighborhoods of the discrete policy space and recover the full result.

Table 8: Ablation of mutation operators on LLaVA-1.5-7B at 34% FLOPs. Both Window Shuffle (WS) and Submodule Swap (SS) help, but Submodule Swap (SS) contributes more.

Method	Ratio	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
SearchV (WS + SS)	34%	**1303.0**	**61.8**	33.3	**33.7**	**60.7**	**55.3**	56.2	**52.3**	**88.8%**
Submodule Swap only	34%	1292.8	61.2	**34.8**	32.7	59.8	55.0	56.4	52.1	88.4%
Window Shuffle only	34%	1182.1	60.6	33.8	30.9	57.7	52.2	**57.2**	50.2	85.2%

5 Limitations

While **SearchV** is robust across diverse benchmarks, several limitations merit future exploration. First, the effectiveness of the Global Contribution metric is inherently tied to the diversity of the multimodal calibration set. Consequently, rare failure modes or long-tail instructions in specialized domains may not be fully captured during calibration. Second, despite optimizations like staged screening and fitness caching, the evolutionary search still incurs non-trivial overhead for forward passes on the calibration set. Additionally, our current FLOPs estimation assumes fixed token counts and homogeneous templates, which may not fully reflect the variability of real-world workloads with diverse image resolutions and task complexities. Lower decoder FLOPs also do not automatically imply lower peak memory or easier deployment when comparing a compressed large model with a smaller dense model; our matched-budget comparisons therefore emphasize accuracy under identical FLOP ratios, hardware, and speedup settings. Future work will explore task-aware multi-objective fitness functions and test-time dynamic re-scoring for more adaptive skipping strategies.

6 Conclusion

In this paper, we address the pervasive inference inefficiency in large vision-language models stemming from the dense processing of long visual sequences. We propose a paradigm shift from localized, heuristic-based pruning to a holistic optimization framework, **SearchV**, which scores complete skipping policies with a Global Contribution metric in a fine-grained, decoupled search space. By independently modulating decisions for multi-head attention and feed-forward network sub-modules, **SearchV** uncovers critical structural redundancies that conventional coarse-grained or layer-wise policies often overlook. Our findings across multiple model families demonstrate that a global, compositional perspective is essential for effectively navigating the complex accuracy-efficiency trade-offs inherent in multimodal inference. We believe this methodology provides a robust baseline for future research into adaptive, hardware-aware acceleration for vision-language systems.

Acknowledgment

This work was supported by the SSTIC Grants under Nos. KJZD20230923115106012, KJZD20230923114916032, and GJHZ20240218113604008.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: A visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[3] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[4] 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*, pages 19–35. Springer, 2024.

[5] 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, 2024.

[6] Xinrui Chen, Hongxing Zhang, Fanyi Zeng, Yongxian Wei, Yizhi Wang, Xitong Ling, Guanghao Li, and Chun Yuan. Prune&Comp: Free lunch for layer-pruned LLMs via iterative pruning with magnitude compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 20316–20324, 2026.

[7] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[8] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[10] 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.

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

[12] 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*, pages 6700–6709, 2019.

[13] Donggon Jang, Yucheol Cho, Suin Lee, Taehyeon Kim, and Dae-Shik Kim. MMR: A large-scale benchmark dataset for multi-target and multi-granularity reasoning segmentation. In *International Conference on Learning Representations*, volume 2025, pages 67406–67431, 2025.

[14] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of 29th Symposium on Operating Systems Principles*, pages 611–626, 2023.

[15] Bo Li, Peiyuan Zhang, Kaichen Zhang, Fanyi Pu, Xinrun Du, Yuhao Dong, Haotian Liu, Yuanhan Zhang, Ge Zhang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Accelerating the development of large multimodal models, March 2024.

[16] 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*, 2023.

[17] 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*, pages 19730–19742. PMLR, 2023.

[18] Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5334–5342, 2025.

[19] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[20] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, 2024.

[21] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

[22] Songwei Liu, Chao Zeng, Lianqiang Li, Chenqian Yan, Lean Fu, Xing Mei, and Fangmin Chen. FoldGPT: Simple and effective large language model compression scheme. *arXiv preprint arXiv:2407.00928*, 2024.

[23] 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*, pages 216–233. Springer, 2024.

[24] Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. DeepSeek-VL: Towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.

[25] Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-pruner: On the structural pruning of large language models. *Advances in Neural Information Processing Systems*, 36:21702–21720, 2023.

[26] Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20192–20204, 2025.

[27] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k Entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2641–2649, 2015.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[29] 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*, pages 22857–22867, 2025.

[30] Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. CrossGET: Cross-guided ensemble of tokens for accelerating vision-language transformers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 44960–44990, 2024.

[31] Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. EvoPress: Accurate dynamic model compression via evolutionary search. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 55556–55590, 2025.

[32] Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. SLEB: Streamlining LLMs through redundancy verification and elimination of transformer blocks. *arXiv preprint arXiv:2402.09025*, 2024.

[33] 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.

[34] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Franceschi, Alexandre Hassani, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[35] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[36] Fei Wang, Xingyu Fu, James Y. Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, et al. MuirBench: A comprehensive benchmark for robust multi-image understanding. In *International Conference on Learning Representations*, volume 2025, pages 62624–62650, 2025.

[37] 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.

[38] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. CogVLM: Visual expert for pretrained language models. *Advances in Neural Information Processing Systems*, 37:121475–121499, 2024.

[39] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[40] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. DeepSeek-VL2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024.

[41] Peng Xia, Siwei Han, Shi Qiu, Yiyang Zhou, Zhaoyang Wang, Wenhao Zheng, Zhaorun Chen, Chenhang Cui, Mingyu Ding, Linjie Li, et al. MMIE: Massive multimodal interleaved comprehension benchmark for large vision-language models. In *International Conference on Learning Representations*, volume 2025, pages 25842–25875, 2025.

[42] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[43] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of LMMs: Preliminary explorations with GPT-4v(ision). *arXiv preprint arXiv:2309.17421*, 2023.

[44] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. MiniCPM-V: A GPT-4V level MLLM on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[45] Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Guoyin Wang, Heng Li, Jiangcheng Zhu, Jianqun Chen, et al. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2024.

[46] Qianhao Yuan, Qingyu Zhang, Yanjiang Liu, Jiawei Chen, Yaojie Lu, Hongyu Lin, Jia Zheng, Xianpei Han, and Le Sun. ShortV: Efficient multimodal large language models by freezing visual tokens in ineffective layers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 329–339, 2025.

[47] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[48] Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. LMMs-Eval: Reality check on the evaluation of large multimodal models, 2024.

[49] Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. BlockPruner: Fine-grained pruning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5065–5080, 2025.

[50] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[51] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Additional Results at Complementary FLOPs Budgets

Table 9 reports operating points that complement the main-text comparison: a milder 47% budget on LLaVA-1.5-13B, LLaVA-NeXT-13B at 34% FLOPs, and Qwen2.5-VL-7B-Instruct under a more aggressive 24% budget. On LLaVA-1.5-13B, **SearchV** retains 98.4% of vanilla performance at 47% FLOPs, outperforming the efficiency-oriented baselines in average score. On LLaVA-NeXT-13B, it maintains an average score of 57.9, clearly above FastV and VTW at a matched compression ratio.

Table 9: Additional results on various VLMs. **SearchV** consistently outperforms baselines across different model families.

Method	TFLOPs	Ratio	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
LLaVA-1.5-13B											
Vanilla	16.6	100%	1531.3	68.9	35.4	36.1	68.2	63.3	79.6	61.2	100%
FastV (K=2, R=60%)	8.0	48%	1404.9	54.6	30.4	35.4	63.9	59.6	**77.9**	56.0	91.5%
VTW (K=17)	8.0	48%	**1555.4**	64.6	34.7	**36.4**	**68.1**	57.9	56.6	56.6	92.5%
ShortV (N=28)	7.8	47%	1417.6	65.5	34.6	35.9	65.4	60.8	74.9	58.3	95.2%
SearchV	7.8	47%	1515.9	**67.5**	**36.7**	36.0	67.7	**62.2**	75.8	**60.2**	**98.4%**
LLaVA-NeXT-13B											
Vanilla	81.8	100%	1570.0	70.5	35.9	39.9	71.9	65.7	66.7	61.3	100%
FastV (K=2, R=70%)	27.1	33%	1503.1	52.2	29.7	36.1	65.3	**62.6**	38.9	51.4	83.9%
VTW (K=13)	27.7	34%	1476.1	**66.3**	**36.1**	38.6	68.8	49.7	7.3	48.6	79.4%
ShortV (N=32)	27.5	34%	1468.4	65.8	35.2	38.9	69.3	60.5	**58.5**	57.4	93.6%
SearchV	27.5	34%	**1536.3**	66.2	35.9	**39.2**	**70.6**	61.8	55.1	**57.9**	**94.5%**
Qwen2.5-VL-7B-Instruct											
Vanilla	8.5	100%	1646.5	82.8	50.2	60.8	74.3	59.2	33.0	63.2	100%
FastV (K=2, R=90%)	2.1	25%	1058.3	44.8	42.0	36.0	47.9	41.1	15.7	40.1	63.4%
VTW (K=5)	2.2	26%	795.5	29.1	43.1	28.7	42.8	36.8	0.8	31.6	50.0%
ShortV (N=24)	2.0	24%	1444.6	69.9	44.3	42.6	67.7	51.1	**24.7**	53.2	84.1%
SearchV	2.0	24%	**1487.6**	**72.5**	**45.6**	**43.7**	**67.8**	**52.4**	23.8	**54.3**	**85.9%**

The robustness of **SearchV** is particularly evident in the extreme compression scenarios for Qwen2.5-VL-7B-Instruct. With merely 24% remaining computational budget, **SearchV** retains 85.9% of the original performance, whereas VTW experiences a dramatic collapse to 50.0%. Notably, our method achieves consistent gains over ShortV across nearly all benchmarks in this high-compression regime, especially on perception-heavy tasks (e.g., MMBench) and reasoning-focused tasks (e.g., MMStar). These findings underscore that the terminal distribution alignment objective provides a robust and universal signal for submodule importance, allowing **SearchV** to adapt to diverse attention mechanisms and token interaction patterns without the need for task-specific tuning.

B Analysis of Performance-Efficiency Frontier

To further investigate the robustness of **SearchV**, we conduct a detailed analysis of the performance-efficiency frontier by varying the target FLOPs ratio from 91% down to 34%. Table 10 reports the results across various reasoning and perception categories on the MMStar benchmark.

Table 10: Analysis of performance under varying FLOPs ratios for **SearchV** and ShortV on MMStar. **SearchV** demonstrates superior robustness in low-resource regimes.

TFLOPs	Ratio	Avg. (Ours/ShortV)	Coarse Per.	Fine Per.	Instance Reas.	Logic Reas.	Math	Sci. & Tech.
8.5	100%	33.7	63.6	25.6	38.9	28.9	27.3	17.8
7.7	91%	**33.6** / 33.5	63.6	25.6	38.9	28.9	26.6	17.8
6.9	81%	**33.8** / **33.8**	63.7	25.5	38.5	28.6	27.7	18.7
6.1	72%	33.6 / **34.0**	63.2	25.2	39.4	28.6	27.5	17.5
5.3	62%	33.2 / **33.3**	62.6	22.4	38.9	28.6	26.7	20.0
4.7	55%	**33.9** / 33.3	62.6	25.9	38.8	28.3	28.4	20.0
4.5	53%	**33.8** / 33.7	63.7	22.8	38.6	29.2	27.3	21.2
3.7	44%	**33.7** / 33.4	60.6	24.9	37.0	26.7	26.5	23.5
2.9	34%	**33.7** / 30.2	62.3	24.0	37.8	27.4	27.0	23.8

The results reveal that **SearchV** exhibits exceptional stability in extremely resource-constrained settings. Notably, at a 34% FLOPs ratio, our method maintains an average score of 33.7, matching the vanilla model's performance, while the competitive baseline ShortV experiences a significant drop

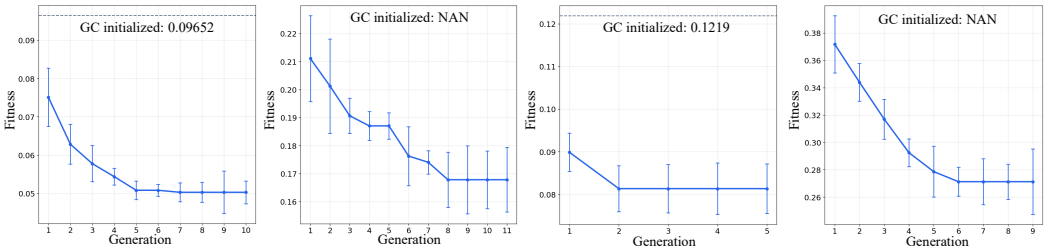
to 30.2. This performance gap underscores the effectiveness of evolutionary search in identifying optimal structural patterns that preserve critical multimodal information even when decoder FLOPs are reduced by about two-thirds.

In high-resource regimes (e.g., ratios above 60%), both **SearchV** and ShortV achieve performance levels comparable to the vanilla model with marginal fluctuations. We attribute this to performance saturation, where the abundance of parameters allows multiple structural configurations to achieve similar accuracy. While the search landscape becomes more complex with higher computational budgets, **SearchV** consistently remains within a narrow margin of optimal performance, demonstrating its reliability across the entire efficiency spectrum. These findings highlight that **SearchV**'s primary strength lies in its ability to navigate the trade-offs of extreme model compression where traditional heuristic-based methods often fail.

The counter-intuitive upward trend in the *Sci. & Tech.* category likely arises from varying degrees of visual dependency; while perception-heavy tasks require high visual fidelity, scientific tasks often rely more on the model's internal knowledge base, where excessive visual tokens can introduce "visual distraction." By strategically pruning submodules, **SearchV** effectively filters out redundant visual features, allowing the model to leverage its strong linguistic priors for knowledge-intensive reasoning.

C Visualization of Convergence Trajectory

Figure 4 illustrates the optimization trajectory of Global Contribution on both LLaVA-1.5-7B and LLaVA-NeXT-7B. We plot the best GC curve along with error bars representing the fitness distribution of each generation. Panels (a)–(b) correspond to LLaVA-1.5-7B, and panels (c)–(d) correspond to LLaVA-NeXT-7B. Figure 4(a) shows LLaVA-1.5-7B at a 44% FLOPs ratio initialized from the ShortV LC metric: the best GC drops quickly and stabilizes by generation 5. Figure 4(c) shows LLaVA-NeXT-7B with a 37% FLOPs ratio. Initialized with the ShortV LC metric, the search starts with a GC of 0.1219. After two generations, the GC rapidly drops and converges to 0.0813, reaching early stopping at the fifth generation. Figure 4(d) presents a more extreme scenario at a 27% FLOPs ratio, where the initial model GC is undefined and the model is collapsed. Despite this invalid starting point, **SearchV** quickly locates valid policies, with the GC decline stabilizing around the sixth generation, demonstrating robustness in high-compression regimes.



(a) LLaVA-1.5-7B (FLOPs ratio 44%) (b) LLaVA-1.5-7B (FLOPs ratio 34%) (c) LLaVA-NEXT-7B (FLOPs ratio 37%) (d) LLaVA-NEXT-7B (FLOPs ratio 27%)

Figure 4: Convergence of evolutionary search on LLaVA-1.5-7B (a,b) and LLaVA-NeXT-7B (c,d). The plots illustrate the optimization of Global Contribution over successive generations across different compression levels.

D Ablation Study on Fitness Objectives

The fitness function used to evaluate candidate configurations governs a critical trade-off between faithfulness to the dense model, computational cost, and generalization. We evaluated several alternatives before settling on the objective described in Section 3.2, and report a controlled comparison below.

Comparison with full-sequence alignment. We first investigated a fitness from EvoPress [31], which calculates the average token-level KL divergence across all token positions. However, in multimodal settings, this approach proved computationally prohibitive due to the memory overhead

of storing full-vocabulary distributions for long sequences. Furthermore, it served as a weak proxy for downstream reasoning, as vision and text tokens contribute disproportionately to the final loss.

Downstream task-specific fitness. We also considered using raw accuracy from a single evaluation task (e.g., MME) as the fitness metric. While intuitive, this approach consistently led to overfitting the specific calibration set and failed to transfer to other benchmarks. Similarly, multi-objective combinations of task metrics proved slow and unstable, as the search became sensitive to task-specific weighting and to variance in small-sample estimates rather than to model faithfulness.

Our choice: last-position distribution alignment. Our final choice adopts the last-position distribution alignment from ShortV [46]. By measuring KL divergence only at the last prompt time step, we capture the model's integrated reasoning result under full context while maintaining high efficiency. The robustness of this objective is reflected in our final results, where configurations optimized on a small calibration pool generalize seamlessly to benchmarks beyond the calibration tasks.

To isolate the fitness objective from the search procedure and the decoupled search space, we compare L1, MSE, and forward KL under identical budgets, mutation operators, and calibration data on LLaVA-1.5-7B at 34% FLOPs (Table 11). KL yields the highest retention of 88.8%. MSE is a close second at 88.5%, while L1 lags at 86.3%, particularly on MME and SEED. All three variants remain well above the ShortV baseline (81.2% in Table 6), indicating that evolutionary search does not rely exclusively on KL. We nevertheless adopt KL because it directly measures distributional alignment of the next-token predictive distribution, which is the quantity GC is designed to preserve.

Table 11: Fitness function comparison on LLaVA-1.5-7B at 34% FLOPs.

Fitness	Ratio	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
Vanilla	100%	1510.7	64.1	36.3	33.7	66.1	61.9	74.9	58.9	100%
L1	34%	1223.7	61.3	32.8	32.7	58.2	53.2	56.4	50.8	86.3%
MSE	34%	1298.1	**62.3**	**33.6**	32.8	59.8	55.0	**56.4**	52.1	88.5%
KL (GC, Ours)	34%	**1303.0**	61.8	33.3	**33.7**	**60.7**	**55.3**	56.2	**52.3**	**88.8%**

E Impact of Search Initialization Strategies

We evaluate the impact of different starting points for our evolutionary search to understand how initialization affects convergence speed and final performance. We compare three strategies on LLaVA-1.5-7B at a fixed budget of 3.7 TFLOPs: (i) ShortV Initialization, using the KL-divergence-based heuristic [46]; (ii) Inverted Order, following the observation from VTW [18] that deep-layer visual tokens are often redundant; and (iii) Random Initialization, starting from a stochastic skipping configuration. Detailed layer importance rankings for these settings are summarized in Table 12.

Table 12: Layer importance rankings for different initialization settings.

Initialization setting	Importance rank (Least → most important)
ShortV	31, 30, 29, 28, 26, 27, 0, 25, 24, 23, 22, 21, 2, 3, 20, 17, 19, 18, 12, 15, 16, 1, 13, 14, 4, 6, 10, 11, 7, 5, 8, 9
Inverted Order	31, 30, 29, 28, 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 17, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0
Random Init.	12, 24, 25, 1, 7, 2, 14, 27, 29, 21, 15, 31, 19, 9, 4, 26, 6, 22, 20, 30, 11, 16, 3, 5, 18, 0, 28, 8, 17, 13, 23, 10

As shown in Table 13, **SearchV** improves performance across all starting configurations. Leveraging either ShortV or Inverted Order as a warm-start accelerates convergence. While the Inverted Order provides a reasonable baseline (90.3% retention), the ShortV prior yields the most efficient path, achieving a peak average score of 57.0 and 96.8% performance retention in only 10 generations.

Table 13: Ablation of initialization strategies on LLaVA-1.5-7B. Bold indicates the best performance within each group.

Method	Gen.	TFLOPs	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
Vanilla	-	8.5	1510.7	64.1	36.3	33.7	66.1	61.9	74.9	58.9	100%
ShortV Init.	-	3.7	1341.7	60.7	34.1	33.4	62.5	58.3	64.2	54.3	92.2%
+ **SearchV**	10	3.7	**1502.7**	**64.7**	**36.3**	**33.7**	**65.6**	**59.1**	**64.6**	**57.0**	**96.8%**
Inverted Init.	-	3.7	1377.0	58.6	34.9	31.7	60.2	55.0	63.2	53.2	90.3%
+ **SearchV**	16	3.7	**1493.1**	**63.6**	**35.9**	**33.5**	**65.2**	**59.0**	**63.6**	**56.5**	**95.9%**
Random Init.	-	3.7	986.0	58.9	33.1	30.3	57.3	51.2	56.8	48.1	81.7%
+ **SearchV**	20	3.7	**1426.2**	**63.3**	**34.1**	**33.8**	**63.1**	**58.2**	**66.0**	**55.7**	**94.5%**

Notably, even starting from a stochastic random initialization where the initial population collapses to a mere 81.7% retention, **SearchV** consistently recovers performance to 94.5% within just 20 generations. This rapid convergence from a sub-optimal starting point underscores the robust capability of our mutation strategy to navigate the vast combinatorial landscape effectively, even without expert-designed priors or other heuristic guidance for the initial population.

F Sensitivity to Calibration Set Composition

We investigate the impact of calibration set diversity on search quality by removing one task at a time while maintaining a constant total sample size of 320. As shown in Table 14, the configuration using a balanced mix of all 5 tasks achieves the highest average performance (60.2). Removing any single task generally leads to degradation, even when the remaining tasks are oversampled to keep the volume constant. Specifically, removing SEED-Bench or Flickr30K results in the most noticeable drops. This suggests that the Global Contribution metric requires a diverse representational distribution to accurately estimate inter-layer dependencies. Interestingly, excluding MME slightly improves its own score but at the cost of overall generalization, reinforcing the necessity of a task-agnostic, multi-modal calibration set rather than a single-benchmark proxy.

Table 14: **Effect of calibration set composition on LLaVA-1.5-7B.** Fixed total samples (320) and FLOPs ratio (44%). Each row denotes the exclusion of a specific task with total volume kept constant.

Calibration Set	Total	S/T	MME	MMMU	SEED	GQA	Flickr	Avg.
Full Mix	320	64	1502.7	**36.3**	**65.6**	**59.1**	**64.6**	**60.2**
w/o MME	320	80	**1521.8**	36.2	65.1	58.7	63.6	59.9
w/o MMMU	320	80	1508.4	35.7	65.5	59.0	63.9	59.9
w/o SEED-Bench	320	80	1482.0	35.8	65.0	58.5	63.2	59.3
w/o GQA	320	80	1497.9	35.6	64.9	**59.1**	64.2	59.7
w/o Flickr30K	320	80	1486.9	35.4	65.0	58.6	63.6	59.4

G Multi-Seed Stability

Evolutionary search is stochastic, so single-seed tables can overstate the reliability of close margins. We therefore rerun the full search on LLaVA-1.5-7B at 34% FLOPs with three random seeds and evaluate every discovered policy on the complete benchmark suite. Table 15 reports the per-seed scores together with the mean and standard deviation.

Table 15: Multi-seed evaluation of **SearchV** on LLaVA-1.5-7B at 34% FLOPs. Mean and standard deviation are computed over three independent searches.

Seed	Ratio	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
Vanilla	100%	1510.7	64.1	36.3	33.7	66.1	61.9	74.9	58.9	100%
0	34%	1367.0	61.8	35.1	33.1	60.6	54.9	55.6	52.8	89.6%
1234	34%	1334.6	61.8	33.6	33.1	60.4	54.7	58.0	52.6	89.3%
42	34%	1303.0	61.8	33.3	33.7	60.7	55.3	56.2	52.3	88.8%
Mean ± Std	34%	1334.9±32.0	61.8±0.0	34.0±1.0	33.3±0.3	60.6±0.2	55.0±0.3	56.6±1.2	52.6±0.3	89.2±0.4%

Across seeds $\{0, 1234, 42\}$, **SearchV** attains an average score of 52.6 ± 0.3 and a performance retention of $89.2 \pm 0.4\%$. The spread is small relative to the gap over ShortV at the same budget (81.2% in Table 6): every seed remains well above the coarse baseline. Per-benchmark variance is largest on MME and Flickr30K, which are the most visually sensitive tasks, while MMBench is unchanged at 61.8. This stability follows from the elitist, budget-preserving evolutionary loop, which repeatedly re-evaluates survivors and therefore does not depend on a single lucky initialization.

H Extended Component Ablations

We expand the ShortV / ShortV-with-search / **SearchV** comparison to two budgets on LLaVA-1.5-13B. Table 16 isolates two effects: (i) replacing the LC heuristic with evolutionary search while keeping a layer-wise action space, and (ii) further unlocking the decoupled MHA/FFN space.

Table 16: Component ablation on LLaVA-1.5-13B at two matched FLOP budgets.

Method	Ratio	MME	MMB	MMMU	MMStar	SEED	GQA	Flickr	Avg.	Per.
Vanilla	100%	1531.3	68.9	35.4	36.1	68.2	63.3	79.6	61.2	100%
ShortV (N=28)	47%	1417.6	65.5	34.6	35.9	65.4	60.8	74.9	58.3	95.2%
ShortV (N=28) w/ Search	47%	1511.5	**68.0**	36.0	**36.2**	**67.7**	61.2	75.4	60.0	98.0%
SearchV	47%	**1515.9**	67.5	**36.7**	36.0	**67.7**	**62.2**	**75.8**	**60.2**	**98.4%**
ShortV (N=32)	40%	1298.8	64.5	33.6	36.0	63.2	59.3	68.4	55.7	91.0%
ShortV (N=32) w/ Search	40%	1430.0	67.7	34.9	35.7	66.9	60.9	70.0	58.2	95.2%
SearchV	40%	**1480.1**	**68.0**	**36.0**	**36.7**	**67.5**	**61.3**	**72.4**	**59.4**	**97.1%**

At 47% FLOPs, search lifts ShortV from 95.2% to 98.0% and decoupling reaches 98.4%; at 40% FLOPs the corresponding numbers are 91.0%, 95.2%, and 97.1%. Fine-grained search therefore delivers larger benefits precisely when the computational budget is tight, which is also the regime where coarse consecutive-layer skipping is most brittle under aggressive compression.

I Skipping Mechanics and Functional Heterogeneity

We clarify the physical meaning of a skipped submodule, which is the same mechanism as ShortV [46] but applied independently to MHA and FFN rather than to an entire layer.

MHA skipping. Visual tokens are excluded from the query and output projections, so they do not receive attention-based feature updates. They remain in the key/value cache, and therefore continue to provide spatial and semantic context to text tokens. Residual connections around MHA are preserved for text tokens; visual-token states are left unchanged when the MHA branch is skipped.

FFN skipping. Visual tokens bypass the feed-forward update and pass through the residual path without an additional nonlinear transform. Layer normalization is applied universally and is never skipped, which keeps activations on the original numerical scale used by the dense model.

KV cache and kernels. Because visual tokens stay in a contiguous K/V layout, the resulting policy is compatible with standard KV-cache implementations and FlashAttention-style kernels. Skipping changes only which projections and FFN blocks are executed, not the set of cached keys and values.

Where computation is retained. The searched policies consistently spend more of the budget on shallow layers, where low-level visual features are formed. Asymmetric configurations are common: early layers often keep MHA while skipping FFN (spatial grounding without redundant feature expansion), whereas middle layers more often keep FFN (semantic enrichment) and skip MHA. This depth-dependent asymmetry is precisely what Submodule Swap is designed to discover, and it explains why unified layer-wise operators leave a substantial gap in Table 8.

J Task-Level Failure Modes under Aggressive Skipping

Fine-grained perception is the most vulnerable category when visual computation is removed too aggressively. Table 17 reports the absolute change relative to the vanilla model at 34% FLOPs.

Table 17: Absolute change versus the vanilla LLaVA-1.5-7B model at 34% FLOPs. Perception-heavy MME and Flickr30K degrade the most under coarse skipping.

Task	**Vanilla**	**ShortV**	**SearchV**	**ShortV Δ**	**SearchV Δ**
MME	1510.7	1079.0	1303.0	−431.7	−207.7
MMB	64.1	57.9	61.8	−6.2	−2.3
MMMU	36.3	31.0	33.3	−5.3	−3.0
MMStar	33.7	30.2	33.7	−3.5	0.0
SEED	66.1	56.0	60.7	−10.1	−5.4
GQA	61.9	52.0	55.3	−9.9	−6.6
Flickr	74.9	53.6	56.2	−21.3	−18.7

ShortV loses 431.7 points on MME and 21.3 points on Flickr30K, whereas knowledge-leaning MMStar drops by only 3.5 points. **SearchV** cuts those perception losses roughly in half on MME (-207.7) and recovers MMStar completely, while still trailing vanilla on Flickr30K. The pattern matches the multi-image results in Table 3: tasks that require precise cross-image or localized visual evidence fail first under coarse skipping, and they are the tasks for which preserving selected MHA pathways yields the largest recovery relative to layer-wise skipping on these perception-heavy evaluations.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes].

   Justification: The abstract and introduction clearly outline our contributions: the Global Contribution (GC) metric, the fine-grained search space for vision-token processing, and the SearchV pipeline.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes].

   Justification: We discuss the limitations of our work in Section 5, specifically addressing the dependency on calibration set diversity, the computational cost of evolutionary search in current VLM research.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A].

Justification: Our work is primarily empirical and focuses on the introduction of a new optimization framework and search space; it does not present new theoretical theorems or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes].

Justification: We provide comprehensive details regarding our search space, evolutionary hyperparameters, and the composition of the calibration set.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes].

   Justification: Search scripts, configuration files, and pre-calculated layer contribution metrics are provided in the supplemental material to ensure faithful reproduction of our results.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes].

   Justification: Detailed experimental settings, including datasets for calibration, evolutionary population size, mutation window size (w), and the specific benchmarks used (MME, MMMU, etc.), are detailed in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes].

   Justification: Convergence plots report per-generation fitness error bars. We further report mean and standard deviation over three independent search seeds on LLaVA-1.5-7B (Appendix G). Full multi-seed sweeps on every backbone remain expensive, so other tables use a representative seed.

8. Experiments Compute Resources

   Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes].

   Justification: We report the GPU settings, memory consumption, and the total search time.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes].

   Justification: Our research adheres to the NeurIPS Code of Ethics. We use publicly available datasets and pre-trained models in accordance with their original licenses.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

Justification: We discuss the positive impact of reducing VLM inference costs on environmental sustainability and edge-device accessibility. Potential negative impacts are minimal as our work focuses on efficiency rather than altering model behavior or content generation.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A].

Justification: Our work focuses on structural optimization for inference efficiency and does not involve the release of new pre-trained weights or potentially sensitive datasets that require additional safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes].

Justification: All used benchmarks (MME, MMMU, SEED-Bench, etc.) and models (LLaVA) are properly cited, and we adhere to their licenses.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A].

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: This research does not involve crowdsourcing or any experiments with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

Justification: No human subject research was conducted; thus, IRB approval was not required.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A].

Justification: LLMs were only used for polishing the language and formatting the LaTeX tables, which did not impact the core methodology or scientific originality of the research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.