PARCEL: POOL-ANCHORED RESAMPLING WITH CONDITIONED ELASTIC QUERIES FOR EFFICIENT VISION-LANGUAGE UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Vision-Language Models (LVLMs) map visual inputs into dense token sequences, increasing inference and memory costs. Elastic compression addresses this by training a single model that can run at multiple visual token budgets. However, existing approaches struggle under aggressive compression. Nested spatial pooling behaves as an imperfect low-pass filter and induces spectral aliasing that obscures fine-grained detail, while nested query resampling replaces explicit grid-aligned tokens with non-local summaries and substantially degrades spatial grounding. To resolve this representational conflict, we introduce PARCEL (**P**ool-**A**nchored **R**esampling with **C**onditioned **EL**astic Queries), a visual tokenization architecture that dynamically partitions the labor of feature extraction. PARCEL establishes spatial pool tokens as low-frequency layout anchors and conditions elastic query tokens on these anchors through Pool-Conditioned Query Resampling. This coupling encourages queries to capture complementary visual details while strengthening the pooled pathway itself, including at budgets where no query tokens are leveraged at inference. Extensive evaluations across 30+ benchmarks and two LVLMs show that PARCEL improves the performance-efficiency Pareto frontier, consistently outperforming existing matryoshka baselines across visual token budgets while preserving the "train once, deploy anywhere" paradigm.

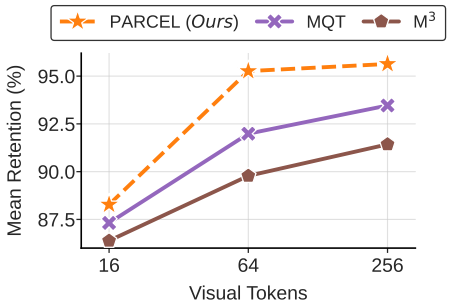
1 INTRODUCTION

Large Vision-Language Models (LVLMs) (Bai et al., 2023; Chen et al., 2025; Steiner et al., 2024) have achieved remarkable success across a wide range of multimodal tasks, spanning video understanding and dense recognition. Despite this success, LVLMs face an input-side bottleneck: images or videos are often represented with hundreds or thousands of visual tokens before being processed by the language decoder. This directly lengthens the language decoder's input, whose self-attention cost grows quadratically. Prior LVLMs (Steiner et al., 2024; An et al., 2025; Liu et al., 2024a) show that increasing the visual token budget often improves visual representation quality and downstream performance, but this comes at a steep compute and memory cost, hindering ubiquitous deployment.

To mitigate this computational burden, prior works explore visual token compression techniques, including dropping (Chen et al., 2024b; Xing et al., 2024), merging (Pippi et al., 2025; Shang et al., 2025), and projection (Cha et al., 2024; Liu et al., 2025). While effective at reducing costs, these approaches typically target fixed-length representations, forcing practitioners to choose a single operating point for deployment. This creates a strict trade-off: an efficiency-optimized model sacrifices fine-grained visual detail, whereas a high-resolution model remains computationally prohibitive for lightweight and latency-sensitive applications. In practice, available resources vary across devices and latency targets, especially when accommodating diverse input domains from images to videos. Elastic inference, a "train once, deploy anywhere" approach has therefore emerged as a valuable goal (Kusupati et al., 2022; Kudugunta et al., 2024).

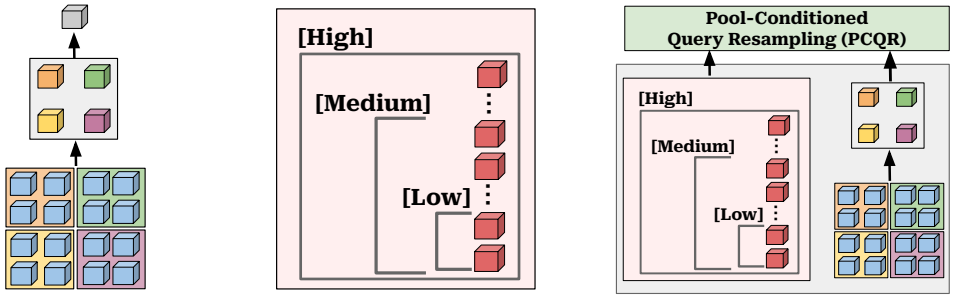
To achieve this elasticity, recent works adapt Matryoshka Representation Learning (Kusupati et al., 2022) to visual tokenization. These efforts primarily branch into two distinct architectural paradigms: rigid spatial downsampling (Cai et al., 2025) and non-local query resampling (Hu et al., 2024). Matryoshka Multimodal Models (M^3) (Cai et al., 2025) represent the former, constructing a nested token structure through successive multi-scale spatial average pooling. Conversely, the Matryoshka

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



Budget	Image TFLOPs	Video TFLOPs	Image KV	Video KV
16	1.0T	4.9T	15MB	33MB
64	1.2T	8.2T	20MB	111MB
256	2.0T	24.3T	39MB	423MB

Figure 1: **Aggregate retention–efficiency trade-off with PaliGemma-2.** *Left*: mean retention relative to Vanilla PG2 over 27 benchmarks and 3 seeds. *Right*: Prefill FLOPs and LLM KV-cache costs by visual token budget. KV-cache is identical across methods at matched budgets, FLOP differences are small since shared ViT/LLM terms dominate and only connector overhead differs by a small margin. PARCEL outperforms MQT and M^3 at matched budgets while lower visual token budgets reduce compute and KV-cache costs vs uncompressed PG2 for image and 16-frame video.



(a) M^3 (Cai et al., 2025) (b) MQT (Hu et al., 2024) (c) PARCEL *(Ours)*

Figure 2: **High-Level Overview of MQT, M^3 and PARCEL (Ours).** M^3 compresses visual features through rigid spatial pooling, MQT uses elastic query tokens, and PARCEL uses spatial anchors to condition elastic queries before they access the full-resolution visual features.

Query Transformer (MQT) (Hu et al., 2024) achieves elasticity using a query transformer paired with nested dropout (Rippel et al., 2014; Srivastava et al., 2014). While both successfully establish an elastic inference paradigm, they introduce opposing representational bottlenecks at constrained token budgets. As we formally analyze in Section 3, the rigid spatial downsampling in M^3 acts as an imperfect low-pass filter. This induces spectral aliasing that blurs the high-frequency semantic details required for resolution-sensitive tasks, such as chart reasoning and text-centric VQA. In contrast, query resampling employed by MQT sacrifices explicit spatial relationships in favor of non-local learned summaries, reducing spatial grounding and dense localization capabilities.

To resolve these representational conflicts, we propose **PARCEL** (**P**ool-**A**nchored **R**esampling with **C**onditioned **EL**astic Queries), a visual tokenization architecture that dynamically partitions the labor of feature extraction as shown in Figure 2. PARCEL encourages a division of labor between spatial anchors obtained by pooling features from the ViT (Dosovitskiy et al., 2021) and learnable query tokens. The spatial pool tokens anchor the low-frequency geometric layout. We then introduce nested query tokens that are explicitly conditioned on these spatial anchors. Operating as dedicated "semantic explorers", these pool-aware queries recover the complementary visual signal that average pooling discards. At anchor-only budgets, PARCEL outperforms M^3 at overall retention rates, showing that the pooled pathway is stronger under our architecture. Our contributions are as follows:

- **Analysis of Spectral Bottlenecks:** We formalize and empirically demonstrate the opposing representational bottlenecks in current elastic methods. Specifically, we show that rigid spatial average pooling in M^3 exhibits spectral signatures akin to aliasing, while non-local query-based resampling in MQT degrades dense spatial understanding under compression.
- **The PARCEL Architecture:** We introduce a visual tokenization architecture that dynamically partitions the labor of feature extraction. Through **Pool-Conditioned Query**

Resampling, we combine the geometric stability of spatial anchors with the high-frequency expressivity of semantic explorer queries. This coupling improves performance when both pathways are active and strengthens the spatial anchors at anchor-only budgets.

- **Budget-Aware Routing and Pareto Efficiency:** We design a budget-aware routing strategy that enables a single model to operate across varying inference budgets. This approach improves the performance-efficiency Pareto frontier across 30+ diverse vision-language benchmarks on two LVLMs. With 64 tokens, PARCEL achieves higher image and video retention than both M^3 and MQT at 256 tokens, using a quarter as many visual tokens.

2 Background and Related Work

Visual Token Compression in LVLMs. The quadratic complexity of attention has driven research into reducing LVLM visual tokens via dropping (Chen et al., 2024b; Xing et al., 2024; Zhang et al., 2026; Liu et al., 2024b; Arif et al., 2025; Alvar et al., 2025; Huang et al., 2025; He & Chen, 2026; Wu et al., 2026c;a), merging (Pippi et al., 2025; Shang et al., 2025), spatial reshaping and resolution adaptation (Chen et al., 2025; Zhu et al., 2025; Liao et al., 2026b; Chen et al., 2024a), projecting (Cha et al., 2024; Liu et al., 2025; Lou et al., 2025; Li et al., 2024; Zamini & Shukla, 2026), and query-based resampling (Bai et al., 2023; Li et al., 2025). As highlighted by recent surveys on efficient multimodal learning (Wang et al., 2025a), optimizing these architectural bottlenecks remains an active research frontier. Recent work (Endo et al., 2025; Huang et al., 2026; Yin et al., 2026; Wu et al., 2026b; Wen et al., 2025b; Baek et al., 2025; Li et al., 2026a) addresses positional biases and spatial distortions caused by aggressive dropping, often through training-free and post-hoc improvements. More broadly, Kong et al. (2026) argue for integrating token reduction into model training rather than treating it solely as a post-hoc efficiency step.

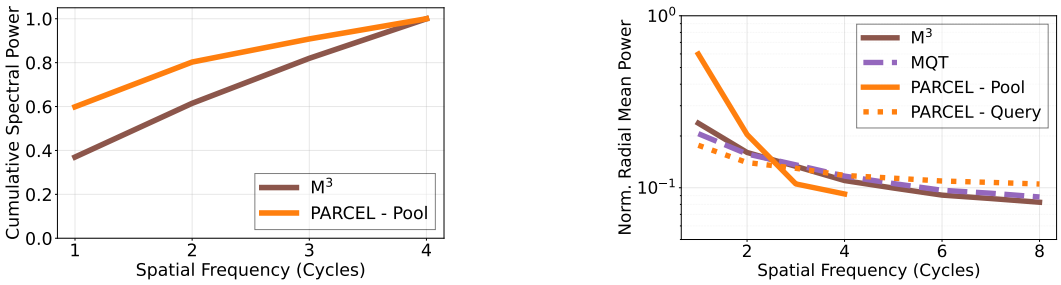
Matryoshka Representation Learning. To achieve inherent deployment elasticity without retraining, Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) encodes information at multiple granularities within a single nested structure. This "train once, deploy anywhere" paradigm spans diverse architectures, including nested transformers (Kudugunta et al., 2024) and Mixture-of-Experts routing (Wang et al., 2025b; Li et al., 2026b; Cappellazzo et al., 2025). Related image and video generation tokenizers use nested dropout or different token masking strategies to learn variable-length representations (Bachmann et al., 2025; Wen et al., 2025a; Yan et al., 2025).

Adaptive and Elastic Visual Tokenization. Following the matryoshka learning paradigm, recent works aim to equip visual token compression with the ability to perform inference at multiple visual token budgets after a single stage of training. Recent methods (Zhong et al., 2025; Ye et al., 2025a; Li & Song, 2025) introduce adaptive inference via dynamic, input-dependent token pruning. Other works realize this flexibility by allowing the user to specify the token count, such as Mask-LLaVA (Jahagirdar et al., 2026). Among these budget-aware architectures, Matryoshka Multimodal Models (M^3) (Cai et al., 2025) and the Matryoshka Query Transformer (MQT) (Hu et al., 2024) stand out as representative paradigms that achieve structural elasticity via two distinct mechanisms. M^3 utilizes multi-scale successive spatial average pooling to obtain elasticity across multiple visual token budgets. Conversely, MQT employs nested-dropout query resampling (Rippel et al., 2014; Srivastava et al., 2014), where the query sequence is truncated to random lengths during training for elasticity.

While both successfully establish elastic inference, they expose opposing bottlenecks. The rigid pooling in M^3 behaves like a spatial downsampling operator and is prone to spectral aliasing that weakens fine-grained detail. In contrast, MQT relies on non-local query resampling, which is suboptimal for spatial understanding. By explicitly dividing the labor of feature extraction, PARCEL conditions elastic queries on spatial anchors, preserving geometric layout while recovering detail beyond the pooled grid. This coupling also improves the anchor-only pathway, where no queries are used at inference. Furthermore, many VQA benchmarks are insensitive to visual token compression, offering limited insight into failures at low budgets (Liao et al., 2026a). Consequently, we validate PARCEL across a wide range of resolution-sensitive, dense reasoning and video understanding tasks.

3 Pool-Anchored Resampling with Conditioned Elastic Queries

While existing matryoshka visual token compression techniques successfully reduce the quadratic cost of visual tokens in LVLMs, relying strictly on either successive spatial average pooling (Cai et al., 2025) or non-local query resampling (Hu et al., 2024) degrades the visual representation in



(a) Baseband Concentration (64 Tokens)

(b) Spectral Disentanglement (256 Tokens)

Figure 3: **Spectral decoupling on ChartQA. (a)** At the 64-token anchor-only budget, cumulative spectral power shows that PARCEL concentrates more of the pooled grid's power at low frequencies than M^3, indicating a cleaner baseband for spatial anchoring. **(b)** Normalized radial mean power shows complementary token roles: pool tokens focus on low-frequency layout, while query-attended ViT feature footprints retain higher-frequency detail beyond the pooled grid. These spectral patterns align with ChartQA gains of $+\mathbf{4.7}$ and $+\mathbf{3.4}$ points over M^3 at 64 and 256 tokens, respectively.

complementary ways under aggressive compression. Rigid spatial downsampling induces spectral aliasing that blurs fine-grained visual details, whereas query-only resampling replaces explicit grid-aligned tokens with non-local summaries, weakening the grid-to-token correspondence needed for dense spatial grounding. To address this representational conflict, we introduce PARCEL, an architecture designed to dynamically partition the labor of feature extraction. First, spatial pool tokens serve as 2D anchors for the low-frequency geometric layout. Second, **Pool-Conditioned Query Resampling** provides a complementary pathway, where learnable query tokens are explicitly conditioned on the spatial anchors before interacting with the raw visual features.

3.1 Spectral Bottlenecks and Spatial Aliasing in Token Compression

To understand how elastic visual token compression changes the information carried by visual features, we analyze the compressed representations in the spatial-frequency domain. This provides a natural lens for separating coarse layout from fine-grained detail: lower spatial frequencies capture slowly varying global structure, whereas higher spatial frequencies correspond to localized changes and detail-sensitive visual features. We therefore use radial power spectral diagnostics to test how different frequency bands are suppressed, preserved, or emphasized by both our baselines and PARCEL. The mathematical details of this analysis are provided in Appendix A.

We first analyze the bottleneck induced by average pooling by evaluating the cumulative spectral concentration of the post-compression spatial grid. This diagnostic avoids the scale ambiguity of raw input-output spectral transfer ratios, testing whether the compressed grid concentrates its spectral mass in the low-frequency baseband. As shown in Figure 3a, PARCEL accumulates spectral power more rapidly at low spatial frequencies than M^3. This indicates stronger low-frequency concentration within the spatial pool tokens of PARCEL. Since both M^3 and PARCEL realize the 64-token budget through 2×2 average pooling, this difference is due to the learned visual features prior to pooling rather than the average pooling itself. Because average pooling does not fully remove frequencies above the downsampled Nyquist limit, this pattern is consistent with high-frequency content folding into the compressed grid (Oppenheim et al., 1999; Gonzalez & Woods, 1992; Zhang, 2019; Azulay & Weiss, 2019). This aligns with the division of labor in PARCEL, where query tokens capture complementary visual details and the spatial pool tokens concentrate on the low-frequency layout.

Query-only compression suffers from a complementary weakness. MQT replaces explicit, grid-aligned spatial tokens with non-local learned summaries, and a nested dropout strategy enforces elasticity over this query sequence (Rippel et al., 2014). While this elastic representation is highly flexible, it forces the queries to encode both the low-frequency layout and fine-grained semantic details without an underlying spatial anchor. As demonstrated in Figure 3b, MQT does not exhibit the same clear separation between low-frequency anchoring and higher-frequency features. This structural weakness is empirically reflected in dense spatial grounding tasks: across the RefCOCO suite (Table 2), PARCEL consistently outperforms MQT across all token budgets, achieving up to a $+5.9$ point retention advantage at 64 tokens. Together, these findings demonstrate the importance of retaining an explicit spatial anchor while conditioning query resampling on that anchor.

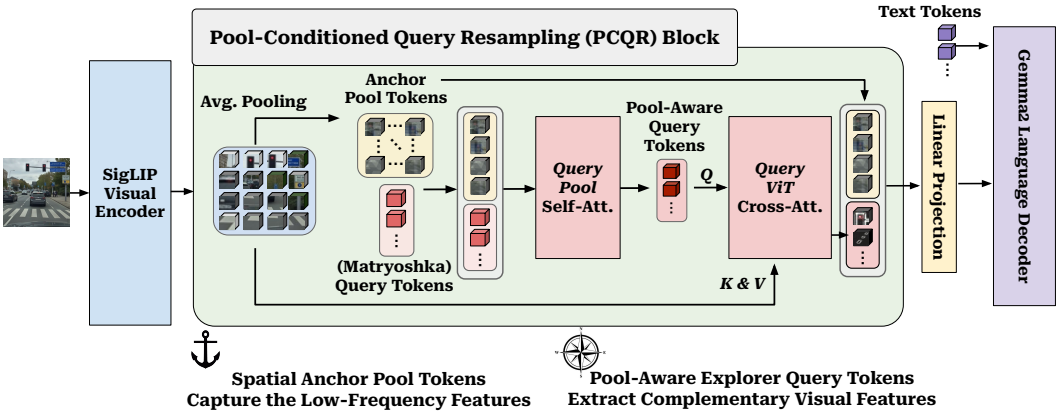


Figure 4: **High-Level Overview of the PARCEL Architecture.** PARCEL dynamically divides the labor of visual feature extraction into a unified pipeline. Uncompressed visual encoder features are first spatially pooled to create deterministic **2D Anchors** that secure the low-frequency geometric layout. A supporting set of query tokens then undergoes **Pool-Conditioned Query Resampling (PCQR)**. After interacting with the spatial anchors through PCQR, these queries act as **Semantic Explorers** that extract complementary information from the raw visual features.

3.2 Pool-Conditioned Query Resampling

We realize this design through Pool-Conditioned Query Resampling (PCQR), an efficient, sequential attention mechanism as outlined in Figure 4. Let $X_v \in \mathbb{R}^{S^2 \times D_v}$ denote visual encoder features on an $S \times S$ source grid. For each budget, we average-pool this grid to the anchor resolution specified by the routing rule in Section 3.3, producing grid-aligned **"2D Anchor" Pool Tokens** $P \in \mathbb{R}^{N_p \times D_v}$. In parallel, a base set of unconditioned, learnable query tokens Q_{IN} undergoes nested dropout to support elastic query budgets. With nested dropout, we sample a budget B, keep only the first $N_q = B - N_p$ query tokens after allocating anchors, and drop the rest. Since earlier queries are active across more budgets, they learn a nested prefix structure that enables query truncation at inference without retraining. To fuse the structural layout with the query pathway, we concatenate these sequences along the token dimension and process the joint representation through a unifying Query $\leftrightarrow$ Pool Self-Attention block. From the output of this block, we retain the updated query sequence as the **Pool-Aware Query Tokens** Q_{PA}, while keeping the original pooled anchors P for the decoder input. This step explicitly conditions the query tokens on the pooled spatial anchors *prior* to dense visual feature extraction, encouraging the query pathway to focus on complementary details absent from the spatial anchors.

After this conditioning step, the queries have access to the coarse spatial layout encoded by the anchors. We then let them cross-attend to the full-resolution ViT features so that they can retrieve complementary visual information not represented in the pooled anchor grid. Here, Q_{PA} serves as the queries (Q), while the raw, uncompressed features X_v act as the keys and values (K, V):

$$Q_{SE} = \text{CrossAttn.}(Q = Q_{PA}, K = X_v, V = X_v). \tag{1}$$

The resulting outputs are **"Semantic Explorer" Query Tokens** (Q_{SE}), which capture complementary features. Finally, we concatenate the original 2D anchors P and semantic explorer queries Q_{SE}, project the resulting tokens into the LLM's embedding space, and input them with the text tokens.

3.3 Budget-Aware Piecewise Routing and Nested Dropout

To realize PARCEL across variable inference constraints, we introduce a budget-aware piecewise routing strategy. Let $N_v = S^2$ be the number of visual-encoder tokens per image or video frame, and let $B_{\max} \leq N_v$ be the largest supported visual token budget. For a given visual encoder, let $\mathcal{A} = \{A_1 < \cdots < A_K\}$ denote the supported pooled-grid sizes, measured in tokens, with $A_K \leq B_{\max}$. We select these grids to retain a spatial layout at the lowest supported budget and to provide denser anchors as the budget increases. For a requested budget B, the router uses the largest anchor grid that fits and assigns the remaining tokens to queries:

$$j(B) = \max\{j : A_j \leq B\}, \qquad (N_p, N_q) = \big(A_{j(B)},\, B - A_{j(B)}\big), \quad A_1 \leq B \leq B_{\max}. \tag{2}$$

Table 1: **Overall Mean Performance with PG2.** Image represents an aggregation of the RefCOCO, Resolution-Sensitive, and General benchmarks. Video represents an aggregation of the video benchmarks. Absolute macro-average raw scores alongside mean retention rates (%) evaluated across 3 random seeds are reported. Aggregated retention rates are the mean of individual benchmark retention rates. Best results per budget are **bolded**, and second-best are underlined.

Modality	Visual Budget	M^3	MQT	PARCEL (Ours)
Image	256 Tokens	67.6 (91.1%)	69.1 (93.3%)	**70.4 (95.1%)**
	64 Tokens	66.5 (89.2%)	68.1 (91.6%)	**70.1 (94.7%)**
	16 Tokens	63.9 (85.2%)	64.3 (85.8%)	**64.8 (86.8%)**
Video	256 Tokens	50.6 (92.9%)	51.4 (94.4%)	**53.1 (98.0%)**
	64 Tokens	50.3 (92.5%)	50.9 (93.5%)	**53.1 (97.9%)**
	16 Tokens	49.8 (91.6%)	51.2 (94.0%)	**51.6 (95.0%)**

Table 2: **Detailed Performance on Video, Image Segmentation, and Resolution-Sensitive Benchmarks.** Raw scores are means over three seeds, with standard deviation shown in gray. Video and Resolution-Sensitive blocks show the top-3 compression-sensitive splits, selected by the largest 16-token retention drop and RefCOCO rows aggregate over splits. Mean Retention rows summarize each block, with RefCOCO computed over all RefCOCO splits. Vanilla PG2 is shaded as the uncompressed reference. Best are **bolded**, second-best are underlined.

Benchmark	PG2	256 Visual Tokens			64 Visual Tokens			16 Visual Tokens		
		M^3	MQT	PARCEL *(Ours)*	M^3	MQT	PARCEL *(Ours)*	M^3	MQT	PARCEL *(Ours)*
Video Understanding Benchmarks (Top 3 Most Compression-Sensitive)										
ActivityNet-CAP	43.7	36.1 ±1.6	37.2 ±1.0	**41.5** ±0.9	36.5 ±0.2	37.1 ±1.3	**40.5** ±1.5	36.5 ±1.0	37.4 ±0.9	**38.9** ±0.4
ActivityNet-QA	53.3	51.5 ±0.3	50.0 ±0.8	**52.4** ±0.7	51.2 ±0.1	49.8 ±0.8	**52.7** ±0.5	50.4 ±0.3	50.4 ±0.9	**50.6** ±0.1
MSRVTT-Cap	70.6	64.2 ±0.9	66.8 ±2.0	**68.6** ±1.6	63.4 ±2.7	64.5 ±2.2	**68.9** ±1.8	62.4 ±1.1	65.7 ±1.8	**66.7** ±0.8
Mean Retention	–	90.1%	91.2%	**96.8%**	89.7%	89.9%	**96.3%**	88.8%	91.1%	**92.8%**
Image Segmentation (RefCOCO)										
RefCOCO (Avg.)	68.0	57.8 ±0.3	61.0 ±0.3	**63.4** ±0.2	56.5 ±0.3	60.3 ±0.3	**63.6** ±0.3	53.0 ±0.2	56.2 ±0.4	**56.7** ±0.1
RefCOCO+ (Avg.)	65.4	52.6 ±0.5	55.5 ±0.3	**58.4** ±0.2	51.0 ±0.4	54.2 ±0.2	**58.4** ±0.2	47.3 ±0.3	50.0 ±0.3	**51.3** ±0.3
RefCOCO-g (Avg.)	65.2	51.2 ±0.4	54.6 ±0.2	**57.7** ±0.1	49.9 ±0.5	53.5 ±0.3	**57.8** ±0.2	46.6 ±0.2	50.2 ±0.1	**51.5** ±0.1
Mean Retention	–	81.7%	86.4%	**90.6%**	79.6%	84.9%	**90.8%**	74.2%	79.0%	**80.5%**
Resolution-Sensitive Tasks (Top 3 Most Compression-Sensitive)										
DocVQA (val)	36.6	30.5 ±0.3	**33.4** ±0.1	32.8 ±0.3	28.7 ±0.1	31.2 ±0.5	**32.1** ±0.4	25.9 ±0.2	24.4 ±0.1	**26.1** ±0.1
ChartQA (human)	40.7	33.8 ±0.3	35.6 ±0.8	**37.2** ±0.6	32.3 ±1.0	34.1 ±0.9	**37.0** ±0.3	29.7 ±0.7	30.7 ±0.3	**32.0** ±1.0
ChartQA (aug)	72.5	64.1 ±0.6	66.0 ±0.5	**66.2** ±0.8	63.0 ±1.3	64.8 ±0.2	**66.3** ±0.2	**59.9** ±0.8	58.2 ±0.9	58.8 ±0.9
Mean Retention	–	84.9%	90.0%	**90.8%**	81.6%	86.2%	**90.0%**	75.5%	74.2%	**77.1%**

Within each routing regime, the anchor grid remains fixed while the query allocation grows as a prefix of the learned query sequence. When the budget reaches the next anchor size, routing switches to the denser pooled grid, and the query allocation starts again from zero. At their default resolution, the anchor-only budgets are $B \in \{16, 64\}$ for PaliGemma-2 and $B \in \{36, 144\}$ for LLaVA 1.5 and no queries are used at inference at these budgets. Nevertheless, the anchor-only and query-active pathways are optimized within the same model during training, allowing the pooled pathway to benefit from the coupled design even when it operates alone.

4 Results and Discussions

We now discuss the experimental evaluation of PARCEL and the baselines (M^3 and MQT) on PaliGemma-2 and LLaVA 1.5. We first measure aggregate performance retention across both LVLMs, then examine benchmarks that stress the bottlenecks identified in Section 3.1, and finally ablate the routing and attention design to test whether the gains arise from the proposed division of labor rather than additional connector capacity alone.

Experimental Setup and Baselines. For the 27-benchmark evaluation, we implement PARCEL and both baselines using PaliGemma-2 (PG2) 3B (Steiner et al., 2024), consisting of a 2B Gemma-2

language decoder (Team et al., 2024) and a SigLIP-SO-400M vision encoder (Zhai et al., 2023). We choose PaliGemma-2 because it provides a compact yet capable open LVLM backbone with established support across the diverse task families needed for our study, including generic VQA, dense localization, resolution-sensitive document/chart reasoning, and video understanding. We also evaluate PARCEL and the baselines on LLaVA 1.5 (Liu et al., 2024a), which combines a CLIP ViT-L/14 vision encoder with a Vicuna language decoder (Chiang et al., 2023). As the model used in the original M^3 and MQT evaluations (Cai et al., 2025; Hu et al., 2024), LLaVA 1.5 provides a direct comparison in a widely used visual token compression setting. The uncompressed PG2 and LLaVA 1.5 models serve as the respective references for retention calculations.

We compare PARCEL with M^3 (Cai et al., 2025) and MQT (Hu et al., 2024) on both LVLMs, and with three training-free methods on LLaVA 1.5 (Appendix B.3). For M^3, we use the one-forward-pass-per-batch variant (Cai et al., 2025) to match training compute budgets. Beyond using different vision encoders and LLMs, the two LVLMs have distinct attention regimes inside the LLM. PG2 uses prefix-LM masking, allowing bidirectional attention across the visual tokens, whereas LLaVA 1.5 processes its concatenated visual and text tokens using the causal attention mask of Vicuna. Evaluating both models therefore also tests whether the benefits of PARCEL persist under both bidirectional prefix attention and causal visual token processing. For efficiency metrics, we follow prior works (Chen et al., 2024b; Xing et al., 2024), with PG2-specific adjustments detailed in Appendix C. We note that PARCEL adds negligible training overhead and the measured LLaVA 1.5 prefill add $+1.0\%$ to MQT, with PARCEL having 1.2–1.3% less latency at anchor-only budgets (Appendix C.2).

Benchmarks. For PG2, we evaluate 27 vision-centric benchmarks spanning video understanding, dense spatial grounding, resolution-sensitive reasoning, and general multimodal comprehension as we detail and describe in Appendix D. Table 2 highlights ActivityNet-Cap (Krishna et al., 2017), ActivityNet-QA (Yu et al., 2019), MSRVTT-Cap (Xu et al., 2016), RefCOCO, RefCOCO+, and RefCOCO-g (Kazemzadeh et al., 2014; Yu et al., 2016; Mao et al., 2016), DocVQA (Mathew et al., 2021), and the human and augmented splits of ChartQA (Masry et al., 2022). We report results on the remaining PG2 benchmarks in Appendix B.1 and high-resolution results on a subset of these benchmarks in Appendix B.2. We adhere to the training settings of PaliGemma-2 (Steiner et al., 2024) whenever applicable. For LLaVA 1.5, we evaluate seven image benchmarks spanning visual reasoning, perception, and text-rich understanding: GQA (Hudson & Manning, 2019), MME-Perception (Fu et al., 2025), POPE (Li et al., 2023c), ScienceQA-IMG (Lu et al., 2022), TextVQA (Singh et al., 2019), SEED-Bench-IMG (Li et al., 2023a), and OCRBench (Liu et al., 2024c). Dataset descriptions, evaluation metrics, and training hyperparameters are provided in Appendix D and Appendix E.

Evaluation Protocol and Design Choices. We report absolute scores and retention relative to each LVLM's uncompressed reference. For method m at budget b, benchmark retention is $100 \times s_{m,b}/s_{\text{vanilla}}$, where $s_{m,b}$ and s_{vanilla} are the compressed and corresponding uncompressed LVLM scores on that benchmark. Aggregate retention is the mean of per-benchmark retention rates within each evaluation suite. PG2 results are averaged over three random seeds. For Table 2, we report the top-3 compression-sensitive splits within the Video and Resolution-Sensitive groups, selected by the largest retention drop at 16 tokens across all methods. This focuses the analysis on settings where visual token compression is most stressful. For the remaining benchmarks, we refer to Appendix B.1.

4.1 Global Efficiency and Pareto Trade-offs

Figure 1 summarizes the aggregate accuracy–efficiency trade-off across all 27 benchmarks. Across all visual token budgets, PARCEL achieves the highest mean retention among compressed models, improving over both MQT and M^3. Table 1 shows that this advantage holds across both image and video domains. In image benchmarks, PARCEL preserves 95.1% and 94.7% retention at 256 and 64 tokens, respectively, and remains best at 16 tokens with 86.8%. In video benchmarks, PARCEL preserves 98.0% and 97.9% retention at 256 and 64 tokens, and 95.0% under the 16-token constraint. Since visual token count controls decoder prefill and KV-cache cost, these gains translate into a stronger accuracy–efficiency trade-off relative to Vanilla PG2.

4.2 Detailed Benchmark Analysis

In this section, we examine tasks that directly probe the bottlenecks identified in Section 3.1.

Video Understanding. The video block of Table 2 shows that the same pattern extends to multi-frame inputs. Averaging the top-3 compression-sensitive video tasks, PARCEL retains 92.8% of

Table 3: **Detailed Performance on LLaVA 1.5.** Raw scores are from one run per configuration. Mean Retention averages retention across the seven benchmarks relative to the shaded uncompressed LLaVA 1.5 column. M^3 results are available at 36 and 144 tokens. Best scores among evaluated methods at each budget are **bolded**, and second-best are underlined.

		256 Visual Tokens		144 Visual Tokens			64 Visual Tokens		36 Visual Tokens		
Benchmark	**LLaVA 1.5**	**MQT**	**PARCEL *(Ours)***	**M^3**	**MQT**	**PARCEL *(Ours)***	**MQT**	**PARCEL *(Ours)***	**M^3**	**MQT**	**PARCEL *(Ours)***
GQA	61.9	61.1	**62.4**	61.6	61.0	**62.1**	60.1	**60.8**	60.2	59.5	**60.7**
MME-P	1505.7	1406.0	**1437.6**	1424.3	1409.3	**1433.0**	1392.7	**1404.7**	1398.0	1341.8	**1424.7**
POPE	85.9	84.0	**84.9**	84.2	83.9	**85.0**	82.6	**83.1**	82.6	**83.0**	82.8
SQA^{IMG}	69.5	**68.6**	67.7	**69.2**	68.6	68.2	**68.1**	67.3	**67.9**	67.7	67.0
TextVQA	58.2	52.8	**54.1**	53.6	52.5	**54.3**	50.7	**51.7**	50.8	50.3	**51.4**
$SEED^{IMG}$	66.2	62.1	**62.6**	62.1	61.1	**62.5**	59.5	**60.7**	59.9	58.7	**60.3**
OCRBench	315	270	**292**	**288**	276	286	250	**315**	260	245	**272**
Mean Retention	–	94.1%	**96.1%**	95.6%	94.1%	**95.9%**	91.4%	**95.1%**	92.0%	90.3%	**93.0%**

Vanilla PG2 performance at 16 tokens, outperforming MQT (91.1%) and M^3 (88.8%). These results further highlight the role of separating spatial anchoring from complementary feature extraction while compressing temporal visual evidence.

Image Segmentation. The RefCOCO block of Table 2 directly tests the spatial grounding weakness of query-only compression. Across the full RefCOCO suite, MQT drops to 79.0% mean retention at 16 tokens, whereas PARCEL retains explicit spatial anchors and achieves 80.5%. At the 64-token anchor-only budget, PARCEL reaches 90.8% mean retention, compared with 84.9% for MQT and 79.6% for M^3, showing that its pooled representation remains stronger without query tokens at inference. The advantage persists at 256 tokens where PARCEL reaches 90.6% mean retention, outperforming MQT by $+\mathbf{4.2}$ points and M^3 by $+\mathbf{8.9}$ points.

Resolution-Sensitive Benchmarks. The resolution-sensitive block of Table 2 stresses fine-grained visual evidence through DocVQA and ChartQA. Averaging the top-3 resolution-sensitive tasks at 16 tokens, M^3 reaches 75.5% mean retention and MQT drops to 74.2%, while PARCEL achieves the best retention at 77.1%. At this boundary budget, PARCEL operates through its spatial-anchor representation, showing that the anchor branch itself provides a stronger compressed visual base. At larger budgets, PARCEL reaches 90.0% retention at 64 tokens from the expanded spatial base and 90.8% at 256 tokens once the queries become active.

LLaVA 1.5 Evaluation. Table 3 compares PARCEL, M^3, and MQT on LLaVA 1.5. At 36 and 144 tokens, where all three methods are available, PARCEL achieves the highest mean retention: 93.0% versus 92.0% for M^3 and 90.3% for MQT at 36 tokens, and 95.9% versus 95.6% and 94.1% at 144 tokens. Both are anchor-only budgets for PARCEL, extending the evidence for a stronger pooled pathway beyond PaliGemma-2. At 64 and 256 tokens, PARCEL retains 95.1% and 96.1%, compared with 91.4% and 94.1% for MQT. The gains at 36 tokens are particularly clear on OCRBench, TextVQA, and $SEED\text{-}Bench^{IMG}$: PARCEL scores 272, 51.4, and 60.3, respectively, exceeding both baselines on each benchmark. At 64 tokens, PARCEL matches uncompressed OCRBench performance (315), compared with 250 for MQT. PARCEL also outperforms all three training-free baselines, exceeding the nearest baseline by $+\mathbf{5.9}$ points (Appendix B.3).

4.3 Ablations on Design Choices

In this section, we quantify the effects of (i) the budget-aware piecewise routing strategy from Section 3.3, (ii) the Pool-Conditioned Query Resampling (PCQR) mechanism from Section 3.2, and (iii) enhanced baseline configurations to isolate the impact of our "division of labor" strategy. All models are evaluated by mean retention across 27 benchmarks.

Impact of Budget-Aware Routing. We hypothesize that spatial anchors must scale relative to the overall token budget. The Budget Routing block of Table 4 tests this across anchor-only and query-active settings. At the anchor-only budgets, PARCEL retains 88.3% and 95.3% of Vanilla PG2 performance at 16 and 64 tokens, compared with 83.1% and 92.8% for the corresponding average-pooling baselines. The pooled pathway therefore performs better under PARCEL even when no query tokens are used at inference. Conversely, fixing the spatial anchor size regardless of the total budget restricts performance. Maintaining a highly compressed 16-token anchor across all budgets causes retention to stagnate at 90.2%, even at 256 tokens. In this regime, the queries must

Table 4: **Ablation Studies on Architecture and Routing.** Overall Mean Retention Rate (%) across the 27-benchmark main evaluation suite. The first row is the full PARCEL configuration, whereas subsequent blocks isolate budget routing, attention design, and baseline-capacity controls. Unsupported budgets are marked as N/A. Best or tied-best displayed values per budget are **bolded**.

Ablation	Model Configuration	256	64	16
Full	**PARCEL *(Ours)***	**95.6%**	**95.3%**	**88.3%**
Budget Routing	Average Pooling (4×4)	N/A	N/A	83.1%
	Average Pooling (2×2)	N/A	92.8%	N/A
	PARCEL w/ Fixed 4×4 Routing	90.2%	89.6%	**88.3%**
	PARCEL w/ Fixed 2×2 Routing	**95.6%**	95.2%	N/A
Attention Design	ViT-Only Cross-Attention	95.2%	**95.3%**	87.9%
	Dual Cross-Attention (ViT + Pool)	95.4%	95.2%	88.0%
Baseline Fairness	MQT w/ Self-Attention	93.3%	92.5%	87.8%
	M^3 w/ Self-Attention	92.2%	90.4%	86.8%

compensate for structural detail that a denser anchor would have preserved. A fixed 64-token anchor reaches 95.2% and 95.6% retention at 64 and 256 tokens, respectively, but cannot accommodate the 16-token setting. Through budget-aware routing, PARCEL uses the 16-token anchor under severe compression and scales to the 64-token anchor when the budget permits, within the same trained model. It achieves 95.6%, 95.3%, and 88.3% retention at 256, 64, and 16 tokens, respectively.

Efficacy of PCQR. The Attention Design block of Table 4 studies the architectural flow of information between the pool tokens, query tokens, and raw visual features. We compare our sequential PCQR module against two alternatives: a "ViT-Only" mechanism where queries cross-attend to the raw visual features without attending to spatial anchors, and a "Dual Cross-Attention" mechanism where queries first cross-attend to the pool tokens and then the ViT features. Our final design first performs a full self-attention between the query tokens and the spatial anchors, and these pool-aware query tokens then cross-attend to the raw visual features. This sequential design (Pool Self-Attn $\rightarrow$ ViT Cross-Attn) achieves the best or tied-best retention across budgets. At the 256-token budget, PCQR reaches 95.6% retention, compared to 95.2% for ViT-only cross-attention and 95.4% for dual cross-attention. The performance delta supports our design choice: for the division of labor to effectively work, the queries must be *pool-aware*. By conditioning the queries on the spatial anchors prior to visual feature extraction, the model guides them away from redundant low-frequency features, reserving their capacity for complementary visual information, as also reflected in Figure 3.

Isolating the Division of Labor (Baseline Fairness). A natural concern is whether the observed gains arise from added learnable parameters, *i.e.*, the extra attention blocks, rather than the structural design. To ensure baseline fairness, we upgrade both MQT and M^3 by adding comparable self-attention blocks, matching the parameter count and depth of PARCEL. As shown in the Baseline Fairness block of Table 4, merely scaling capacity does not resolve the foundational bottlenecks of the baselines. Upgraded MQT reaches 93.3% at 256 tokens, but still falls short of PARCEL (95.6%) because it inherently lacks spatial anchors. Similarly, upgraded M^3 reaches 92.2% at 256 tokens and remains below PARCEL across all budgets, indicating that added self-attention does not overcome the bottlenecks caused by rigid spatial downsampling. These results support that the gains of PARCEL are not a product of added complexity alone, but rather from PARCEL's dynamic division of labor.

5 Conclusion

LVLMs face severe computational bottlenecks during inference, and existing elastic compression paradigms force a trade-off between spectral aliasing and degraded spatial grounding. To resolve this, we introduced PARCEL, a novel method that dynamically partitions the labor of visual feature extraction. By coupling spatial anchors with pool-conditioned queries, PARCEL disentangles low-frequency geometric layout from complementary high-frequency visual details. Extensive evaluations across 30+ benchmarks and two LVLMs show that PARCEL improves the performance–efficiency Pareto frontier across visual token budgets and PARCEL's gains at anchor-only budgets further show that the coupled design strengthens the pooled pathway itself. PARCEL therefore preserves the highly desirable "train once, deploy anywhere" paradigm while consistently outperforming existing elastic baselines, providing a flexible and efficient foundation for ubiquitous LVLM deployment.

AI USE STATEMENT

During manuscript preparation and revision, we only used generative AI tools under author supervision to improve readability (*e.g.* to spot and fix typos or basic grammar errors) and clarify potential narrative ambiguities. Generative AI tools were not used to develop the proposed architecture, implement the experimental methods, or generate experimental or theoretical data or results. The authors reviewed all AI-assisted material and no AI-assisted material was included in the work without author supervision. As authors, we take full responsibility for the final content of this work, including its text, claims, and reported results.

ETHICS STATEMENT

This work studies efficient visual token compression using existing large vision-language models and datasets. Reducing inference costs can make these models more accessible, but does not eliminate biases, hallucinations, or privacy concerns inherited from their training data and base models. Deployment should therefore account for the fairness requirements of the intended application and the limitations of the underlying base models, particularly where incorrect outputs could have negative, undesirable effects.

REPRODUCIBILITY STATEMENT

We describe PARCEL's Pool-Conditioned Query Resampling mechanism in the methodology section and formalize budget-aware routing in Section 3.3. Appendix E documents the PaliGemma-2 and LLaVA 1.5 implementations, including connector architectures, budget sampling, training stages, trainable components, training data and references to the utilized hyperparamaters to ensure full reproducibility of our results. Appendix D describes the datasets and evaluation protocols, while Appendix B provides detailed benchmark results. The spectral analysis protocol and FLOP and KV-cache calculations are specified in Appendices A and C, respectively.

REFERENCES

Manoj Acharya, Kushal Kafle, and Christopher Kanan. Tallyqa: Answering complex counting questions. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 8076–8084, 2019.

Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 8948–8957, 2019.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401, June 2025.

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, Chunsheng Wu, Huajie Tan, Chunyuan Li, Jing Yang, Jie Yu, Xiyao Wang, Bin Qin, Yumeng Wang, Zizhen Yan, Ziyong Feng, Ziwei Liu, Bo Li, and Jiankang Deng. Llava-onevision-1.5: Fully open framework for democratized multimodal training, 2025. URL https://arxiv.org/abs/2509.23661.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2): 1773–1781, Apr. 2025. doi: 10.1609/aaai.v39i2.32171. URL https://ojs.aaai.org/index.php/AAAI/article/view/32171.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *Journal of Machine Learning Research*, 20(184):1–25, 2019.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. Flextok: Resampling images into 1d token

sequences of flexible length. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DgdOkUUBzf.

Changwoo Baek, Jouwon Song, Sohyeon Kim, and Kyeongbo Kong. An empirical study of attention and diversity for adaptive visual token pruning in multimodal reasoning models. In *NeurIPS 2025 Workshop on Efficient Reasoning*, 2025. URL https://openreview.net/forum?id=j2NkINd3pw.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL https://arxiv.org/abs/2308.12966.

Lichen Bai, Zixuan Xiong, Hai Lin, Guangwei Xu, Xiangjin Xie, Ruijie Guo, Zhanhui Kang, Hai-Tao Zheng, and Hong-Gee Kim. Frozen language models are gradient coherence rectifiers in vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 1817–1825, 2025.

Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sagnak Tasırlar. Introducing our multimodal models, 2023. *URL https://www. adept. ai/blog/fuyu-8b*, 2, 2023.

Irwan Bello, Hieu Pham, Quoc V Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2016.

Lucas Beyer, Xiaohua Zhai, and Alexander Kolesnikov. Better plain vit baselines for imagenet-1k. *arXiv preprint arXiv:2205.01580*, 2022.

Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024.

Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluis Gomez, Marçal Rusinol, Ernest Valveny, CV Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4291–4301, 2019.

Ronald N Bracewell. The fourier transform. *Scientific American*, 260(6):86–95, 1989.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Yash Katariya, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax.

Adrian Bulat, Yassine Ouali, and Georgios Tzimiropoulos. Fwd2bot: Lvlm visual token compression with double forward bottleneck, 2025. URL https://arxiv.org/abs/2503.21757.

Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Uhj5OxAz7I.

Umberto Cappellazzo, Minsu Kim, Pingchuan Ma, Honglie Chen, Xubo Liu, Stavros Petridis, and Maja Pantic. Mome: Mixture of matryoshka experts for audio-visual speech recognition, 2025. URL https://arxiv.org/abs/2510.04136.

Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13817–13827, June 2024.

Soravit Changpinyo, Doron Kukliansy, Idan Szpektor, Xi Chen, Nan Ding, and Radu Soricut. All you may need for vqa are image captions. In *Proceedings of the 2022 conference of the north american chapter of the association for computational linguistics: human language technologies*, pp. 1947–1963, 2022.

David Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pp. 190–200, 2011.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. Efficient large multi-modal models via visual context compression. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 73986–74007. Curran Associates, Inc., 2024a. doi: 10.52202/079017-2353.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024b.

Ting Chen, Saurabh Saxena, Lala Li, David J Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. *arXiv preprint arXiv:2109.10852*, 2021.

Xi Chen, Xiao Wang, Soravit Changpinyo, Anthony J Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*, 2022.

Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023a.

Xi Chen, Xiao Wang, Lucas Beyer, Alexander Kolesnikov, Jialin Wu, Paul Voigtlaender, Basil Mustafa, Sebastian Goodman, Ibrahim Alabdulmohsin, Piotr Padlewski, et al. Pali-3 vision language models: Smaller, faster, stronger. *arXiv preprint arXiv:2310.09199*, 2023b.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

Yangyi Chen, Xingyao Wang, Hao Peng, and Heng Ji. A single transformer for scalable vision-language modeling. *arXiv preprint arXiv:2407.06438*, 2024c.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024d.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, Lixin Gu, Xuehui Wang, Qingyun Li, Yiming Ren, Zixuan Chen, Jiapeng Luo, Jiahao Wang, Tan Jiang, Bo Wang, Conghui He, Botian Shi, Xingcheng Zhang, Han Lv, Yi Wang, Wenqi Shao, Pei Chu, Zhongying Tu, Tong He, Zhiyong Wu, Huipeng Deng, Jiaye Ge, Kai Chen, Kaipeng Zhang, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling, 2025. URL `https://arxiv.org/abs/2412.05271`.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Haiwen Diao, Yufeng Cui, Xiaotong Li, Yueze Wang, Huchuan Lu, and Xinlong Wang. Unveiling encoder-free vision-language models. *arXiv preprint arXiv:2406.11832*, 2024.

Haiwen Diao, Xiaotong Li, Yufeng Cui, Yueze Wang, Haoge Deng, Ting Pan, Wenxuan Wang, Huchuan Lu, and Xinlong Wang. Evev2: Improved baselines for encoder-free vision-language models. *arXiv preprint arXiv:2502.06788*, 2025.

Haiwen Diao, Jiahao Wang, Chenjing Ding, Hanming Deng, Jiangnan Chen, Ruixi Zhang, Ruohui Wang, Wenwen Tong, Xiangyu Fan, Yubo Wang, et al. Sensenova-u1. 5: Towards native unified visual intelligence. *arXiv preprint arXiv:2609.11929*, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22826–22835, 2025.

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. *Advances in Neural Information Processing Systems*, 38, 2025.

Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

Google Cloud. Introduction to cloud tpu. https://cloud.google.com/tpu/docs/intro-to-tpu, 2026. Accessed: 2026-05-06.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Xiaoyang Guo, Jusheng Zhang, and Keze Wang. Adaptive-voco: Complexity-aware visual token compression for vision-language models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2396–2400, 2026. doi: 10.1109/ICASSP55912.2026.11465113.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3608–3617, 2018.

Jialuo He and Huangxun Chen. Energy-driven adaptive visual token pruning for efficient vision-language models, 2026. URL https://arxiv.org/abs/2603.05950.

Ting-Yao Hsu, C Lee Giles, and Ting-Hao Huang. Scicap: Generating captions for scientific figures. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3258–3264, 2021.

Wenbo Hu, Zi-Yi Dou, Liunian H Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. *Advances in Neural Information Processing Systems*, 37:50168–50188, 2024.

Xiaohu Huang, Hao Zhou, and Kai Han. PruneVid: Visual token pruning for efficient video large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1024. URL https://aclanthology.org/2025.findings-acl.1024/.

Yihong Huang, Fei Ma, Yihua Shao, Jingcai Guo, Zitong YU, Laizhong Cui, and Qi Tian. Nüwa: Mending the spatial integrity torn by VLM token pruning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=C9yclwdquU.

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

Soumya Jahagirdar, Walid Bousselham, Anna Kukleva, and Hilde Kuehne. When llava meets objects: Token composition for vision-language-models. *arXiv preprint arXiv:2602.04864*, 2026.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 787–798, 2014.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Zhenglun Kong, Yize Li, Fanhu Zeng, Lei Xin, Shvat Messica, Xue Lin, Pu Zhao, Manolis Kellis, Hao Tang, and Marinka Zitnik. Token reduction should go beyond efficiency in generative models – from vision, language to multimodality, 2026. URL https://arxiv.org/abs/2505.18227.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pp. 706–715, 2017.

Sneha Kudugunta, Aditya Kusupati, Tim Dettmers, Kaifeng Chen, Inderjit Dhillon, Yulia Tsvetkov, Hannaneh Hajishirzi, Sham Kakade, Ali Farhadi, and Prateek Jain. Matformer: Nested transformer for elastic inference. *Advances in Neural Information Processing Systems*, 37:140535–140564, 2024.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, et al. Matryoshka representation learning. *Advances in Neural Information Processing Systems*, 35:30233–30249, 2022.

Selim Kuzucu, Muhammad Ferjad Naeem, Anna Kukleva, Federico Tombari, and Bernt Schiele. Enhancing self-supervised visual representation learning via low-rank adapted LLMs. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=s2T8Kgj6Rd.

Zhixin Lai, Jing Wu, Suiyao Chen, Yucheng Zhou, and Naira Hovakimyan. Residual-based language models are free boosters for biomedical imaging tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5086–5096, 2024.

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

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

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

Kevin Li, Sachin Goyal, João D. Semedo, and J Zico Kolter. Inference optimal VLMs need fewer visual tokens and more parameters. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=6VhDQP7WGX.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm, 2024. URL https://arxiv.org/abs/2407.02392.

Xue Li and Xiaonan Song. Efficient vision-language reasoning via adaptive token pruning. In *1st Workshop on VLM4RWD @ NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=Vbqemx4YCC.

Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: Generating natural language description for mobile user interface elements. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 5495–5510, 2020.

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

Yueying Li, Fengxiang Wang, Yan Li, Mingshuo Chen, Mengying Zhao, and Long Lan. Semantic-geometric dual compression: Training-free visual token reduction for ultra-high-resolution remote sensing understanding, 2026a. URL https://arxiv.org/abs/2604.11122.

Zhongyang Li, Yaqian Li, Faming Fang, Rinyoichi Takezoe, Zi-Hao Bo, Cheng Qian, Mo Guang, Guixu Zhang, and Kaiwen Long. Qmop: Query guided mixture-of-projector for efficient visual token compression, 2026b. URL https://arxiv.org/abs/2603.21232.

Chenfei Liao, Wensong Wang, Zichen Wen, Xu Zheng, Yiyu Wang, Haocong He, Yuanhuiyi Lyu, Lutao Jiang, Xin Zou, Yuqian Fu, Bin Ren, Linfeng Zhang, and Xuming Hu. Are we using the right benchmark: An evaluation framework for visual token compression methods. *arXiv preprint arXiv:2510.07143*, 2026a.

Huanxuan Liao, Zhongtao Jiang, Yupu Hao, Yuqiao Tan, Shizhu He, Ben Wang, Jun Zhao, Kun Xu, and Kang Liu. Resadapt: Adaptive resolution for efficient multimodal reasoning, 2026b. URL https://arxiv.org/abs/2603.28610.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Fangyu Liu, Emanuele Bugliarello, Edoardo Maria Ponti, Siva Reddy, Nigel Collier, and Desmond Elliott. Visually grounded reasoning across languages and cultures. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 10467–10485, 2021.

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

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 (CVPR)*, pp. 26296–26306, June 2024a.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. Multi-stage vision token dropping: Towards efficient multimodal large language model. *arXiv preprint arXiv:2411.10803*, 2024b.

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

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Haotian Tang, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Jinyi Hu, Sifei Liu, Ranjay Krishna, Pavlo Molchanov, Jan Kautz, Hongxu Yin, Song Han, and Yao Lu. Nvila: Efficient frontier visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4122–4134, June 2025.

Sylvain Lobry, Diego Marcos, Jesse Murray, and Devis Tuia. Rsvqa: Visual question answering for remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12):8555–8566, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Haoran Lou, Chunxiao Fan, Ziyan Liu, Yuexin Wu, and Xinliang Wang. Llava-sp: Enhancing visual representation with visual spatial tokens for mllms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22014–22024, October 2025.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521, 2022.

Gen Luo, Xue Yang, Wenhan Dou, Zhaokai Wang, Jiawen Liu, Jifeng Dai, Yu Qiao, and Xizhou Zhu. Mono-internvl: Pushing the boundaries of monolithic multimodal large language models with endogenous visual pre-training. *arXiv preprint arXiv:2410.08202*, 2024.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 11–20, 2016.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 3195–3204, 2019.

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

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 947–952. IEEE, 2019.

Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. Prentice-Hall, 2nd edition, 1999.

Ziqi Pang, Ziyang Xie, Yunze Man, and Yu-Xiong Wang. Frozen transformers in language models are effective visual encoder layers. *arXiv preprint arXiv:2310.12973*, 2023.

Jonas Pfeiffer, Gregor Geigle, Aishwarya Kamath, Jan-Martin O Steitz, Stefan Roth, Ivan Vulić, and Iryna Gurevych. xgqa: Cross-lingual visual question answering. In *Findings of the association for computational linguistics: ACL 2022*, pp. 2497–2511, 2022.

Vittorio Pippi, Matthieu Guillaumin, Silvia Cascianelli, Rita Cucchiara, Maximilian Jaritz, and Loris Bazzani. Tofu: Visual tokens reduction via fusion for multi-modal, multi-patch, multi-image task, 2025. URL https://arxiv.org/abs/2503.04444.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: international conference for high performance computing, networking, storage and analysis*, pp. 1–16. IEEE, 2020.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 3505–3506, 2020.

Oren Rippel, Michael Gelbart, and Ryan Adams. Learning ordered representations with nested dropout. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1746–1754, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/rippel14.html.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. In *European conference on computer vision*, pp. 146–162. Springer, 2022.

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

Claude E Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949. doi: 10.1109/JRPROC.1949.232969.

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2556–2565, 2018.

Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *European conference on computer vision*, pp. 742–758. Springer, 2020.

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

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014. URL `http://jmlr.org/papers/v15/srivastava14a.html`.

Andreas Steiner, André Susano Pinto, Michael Tschannen, Daniel Keysers, Xiao Wang, Yonatan Bitton, Alexey Gritsenko, Matthias Minderer, Anthony Sherbondy, Shangbang Long, et al. Paligemma 2: A family of versatile vlms for transfer. *arXiv preprint arXiv:2412.03555*, 2024.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 6418–6428, 2019.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Ashish V Thapliyal, Jordi Pont Tuset, Xi Chen, and Radu Soricut. Crossmodal-3600: A massively multilingual multimodal evaluation dataset. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 715–729, 2022.

Bo Wan, Michael Tschannen, Yongqin Xian, Filip Pavetic, Ibrahim Alabdulmohsin, Xiao Wang, André S Pinto, Andreas Steiner, Lucas Beyer, and Xiaohua Zhai. Locca: Visual pretraining with location-aware captioners. *Advances in Neural Information Processing Systems*, 37:116355–116387, 2024.

Bryan Wang, Gang Li, Xin Zhou, Zhourong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, pp. 498–510, 2021.

Pan Wang, Siwei Song, Hui Ji, Siqi Cao, Heng Yu, Zhijian Liu, Huanrui Yang, Yingyan (Celine) Lin, Beidi Chen, Mohit Bansal, Xiaoming Liu, Pengfei Zhou, Ming-Hsuan Yang, Tianlong Chen, and Jingtong Hu. From models to systems: A comprehensive survey of efficient multimodal learning. *Authorea Preprints*, 2025a.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4581–4591, 2019.

Yaoxiang Wang, Qingguo Hu, Yucheng Ding, Ruizhe Wang, Yeyun Gong, Jian Jiao, Yelong Shen, Peng Cheng, and Jinsong Su. Training matryoshka mixture-of-experts for elastic inference-time expert utilization, 2025b. URL https://arxiv.org/abs/2509.26520.

Xin Wen, Bingchen Zhao, Ismail Elezi, Jiankang Deng, and Xiaojuan Qi. "principal components" enable a new language of images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 16641–16651, October 2025a.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 15537–15549, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.802. URL https://aclanthology.org/2025.findings-acl.802/.

Guangxin Wu, Zumeng Zhang, Hao Li, Yangfan He, Yue Jiang, Zhaolu Kang, Weilin Zhou, and Hao Zhang. Vision meets language: Adaptive joint pruning for efficient multimodal models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 19727–19731, 2026a. doi: 10.1109/ICASSP55912.2026.11461792.

Hao Wu, Ke Lu, Xiuyuan Zhu, Yuqiu Li, Jian Xue, and Yi Liu. Laprune: Layout-aware pruning for efficient multimodal large language models. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 12922–12926, 2026b. doi: 10.1109/ICASSP55912.2026.11464292.

Zhenkai Wu, Xiaowen Ma, Zhenliang Ni, Dengming Zhang, Han Shu, Xin Jiang, and Xinghao Chen. Vlm-pruner: Buffering for spatial sparsity in an efficient vlm centrifugal token pruning paradigm, 2026c. URL https://arxiv.org/abs/2512.02700.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *arXiv preprint arXiv:2410.17247*, 2024.

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1645–1653, 2017.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5288–5296, 2016.

Yuanzhong Xu, HyoukJoong Lee, Dehao Chen, Blake Hechtman, Yanping Huang, Rahul Joshi, Maxim Krikun, Dmitry Lepikhin, Andy Ly, Marcello Maggioni, et al. Gspmd: general and scalable parallelization for ml computation graphs. *arXiv preprint arXiv:2105.04663*, 2021.

Wilson Yan, Volodymyr Mnih, Aleksandra Faust, Matei Zaharia, Pieter Abbeel, and Hao Liu. Elastictok: Adaptive tokenization for image and video. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tFV5GrWOGm.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24972–24982, June 2025a.

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 29836–29846, June 2025b.

Shukang Yin, Sirui Zhao, Hanchao Wang, Baozhi Jia, Xianquan Wang, Chaoyou Fu, and Enhong Chen. Tango: Taming visual signals for efficient video large language models, 2026. URL https://arxiv.org/abs/2604.09547.

Licheng Yu, Patrick Poirson, Shan Yang, Alexander C Berg, and Tamara L Berg. Modeling context in referring expressions. In *European conference on computer vision*, pp. 69–85. Springer, 2016.

Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 9127–9134, 2019.

Mohamad Zamini and Diksha Shukla. Delta-llava: Base-then-specialize alignment for token-efficient vision-language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3648–3657, March 2026.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Haitao Mi, and Dong Yu. VScan: Rethinking visual token reduction for efficient large vision-language models. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=KZYhyilFnt.

Jusheng Zhang, Xiaoyang Guo, Kaitong Cai, Qinhan Lv, Yijia Fan, Wenhao Chai, Jian Wang, and Keze Wang. Hybridtoken-vlm: Hybrid token compression for vision-language models, 2025. URL https://arxiv.org/abs/2512.08240.

Richard Zhang. Making convolutional networks shift-invariant again. In *International Conference on Machine Learning*, pp. 7324–7334. PMLR, 2019.

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. Aim: Adaptive inference of multi-modal llms via token merging and pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20180–20192, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

CONTENTS

1 Introduction **1**

2 Background and Related Work **3**

3 Pool-Anchored Resampling with Conditioned Elastic Queries **3**

3.1 Spectral Bottlenecks and Spatial Aliasing in Token Compression 4

3.2 Pool-Conditioned Query Resampling 5

3.3 Budget-Aware Piecewise Routing and Nested Dropout 5

4 Results and Discussions **6**

4.1 Global Efficiency and Pareto Trade-offs 7

4.2 Detailed Benchmark Analysis 7

4.3 Ablations on Design Choices 8

5 Conclusion **9**

A Spectral Analysis Protocol **21**

B Additional Experimental Results and Discussions **23**

B.1 Detailed Results Across All PaliGemma-2 Benchmarks 23

B.2 Detailed Results for Benchmarks at High Resolution 25

B.3 Comparisons Against Training-Free Baselines with LLaVA 1.5 26

C Efficiency Measures and Calculations **28**

C.1 FLOP and KV-Cache Calculations 28

C.2 Measured Prefill Latency and Memory 31

D Benchmark Details **31**

D.1 PaliGemma-2 Benchmark Suite 31

D.2 LLaVA 1.5 Benchmark Suite 33

E Implementation Details **33**

E.1 Implementation and Evaluation Details for PaliGemma-2 Experiments 33

E.2 Implementation and Evaluation Details for LLaVA 1.5 Experiments 35

F Further Discussion of Related Work **36**

F.1 Visual Token Compression at Different Stages of LVLM Pipeline 36

F.2 Visual Token Compression Through Visual Bottlenecks and Register Tokens 36

G Limitations **36**

A SPECTRAL ANALYSIS PROTOCOL

We use spectral diagnostics to analyze how different visual token compression mechanisms allocate spatial-frequency power. Spatial pooling can be viewed as a downsampling operation: reducing a feature grid from $H \times W$ to $H' \times W'$ lowers the maximum representable spatial frequency, and components above the reduced Nyquist limit may alias into lower frequencies if they are not attenuated before decimation (Shannon, 1949; Oppenheim et al., 1999; Zhang, 2019). Our goal is not to reconstruct the original feature map, but to characterize the extent to which compressed spatial tokens capture low-frequency features and query tokens emphasize higher-frequency source-grid detail beyond what is retained by the pooled spatial anchors.

Feature grids. Let $\mathbf{M} \in \mathbb{R}^{H_M \times W_M \times C}$ denote a visual feature grid, where $H_M \times W_M$ is the native spatial resolution and C is the channel dimension. All spectra are computed on native feature grids before projection into the language-model embedding space. Pooled grids are not upsampled before Fourier analysis.

We first remove the spatially constant component of each channel:

$$\tilde{\mathbf{M}}_{h,w,c} = \mathbf{M}_{h,w,c} - \frac{1}{H_M W_M} \sum_{h'=0}^{H_M-1} \sum_{w'=0}^{W_M-1} \mathbf{M}_{h',w',c}. \tag{3}$$

We then compute the forward normalized 2D Discrete Fourier Transform. For notational brevity, we first define the complex exponential basis term $\mathcal{E}_{h,w}(u,v)$ as:

$$\mathcal{E}_{h,w}(u,v) = \exp\left[-2\pi i \left(\frac{uh}{H_M} + \frac{vw}{W_M}\right)\right]. \tag{4}$$

The transform is then compactly given by:

$$\widehat{\mathbf{M}}_c(u,v) = \frac{1}{H_M W_M} \sum_{h=0}^{H_M-1} \sum_{w=0}^{W_M-1} \tilde{\mathbf{M}}_{h,w,c} \mathcal{E}_{h,w}(u,v). \tag{5}$$

This normalization prevents larger spatial grids from producing larger Fourier magnitudes solely because they contain more samples. Under this convention, Parseval's relation gives:

$$\sum_{u,v} \left|\widehat{\mathbf{M}}_c(u,v)\right|^2 = \frac{1}{H_M W_M} \sum_{h,w} \left|\tilde{\mathbf{M}}_{h,w,c}\right|^2, \tag{6}$$

so the summed spectral power corresponds to the average spatial AC energy per feature location. This makes spectra scale-consistent across grids of different spatial resolutions (Oppenheim et al., 1999; Bracewell, 1989; Gonzalez & Woods, 1992).

Finally, we compute the channel-averaged power spectrum, which yields power spectral density (PSD):

$$S_{\mathbf{M}}(u,v) = \frac{1}{C} \sum_{c=1}^{C} \left|\widehat{\mathbf{M}}_c(u,v)\right|^2. \tag{7}$$

Radial mean power. To obtain one-dimensional frequency profiles for our visualizations, we collapse the 2D Fourier spectrum into a radial profile. After applying the standard FFT shift, the DC component is placed at the frequency origin, and each Fourier coefficient can be indexed by zero-centered spatial-frequency coordinates:

$$\begin{aligned} u &\in \left\{-\left\lfloor \frac{H_M}{2} \right\rfloor, \ldots, \left\lceil \frac{H_M}{2} \right\rceil - 1\right\}, \\ v &\in \left\{-\left\lfloor \frac{W_M}{2} \right\rfloor, \ldots, \left\lceil \frac{W_M}{2} \right\rceil - 1\right\}. \end{aligned} \tag{8}$$

The radial frequency of a coefficient is its Euclidean distance from the origin in this frequency plane:

$$\rho(u, v) = \sqrt{u^2 + v^2}. \tag{9}$$

Radial averaging is reliable only for frequency rings that are fully represented inside the finite 2D Fourier grid. We therefore restrict comparisons to the largest centered circle that fits inside this grid, *i.e.,* the inscribed Nyquist radius:

$$r_{\max}(\mathbf{M}) = \frac{1}{2} \min(H_M, W_M). \tag{10}$$

This native-grid cutoff is the reason for different x-axis ranges in Figure 3. To exemplify, a 16×16 source grid has an inscribed radial Nyquist limit of $r_{\max} = 8$, whereas an 8×8 pooled grid has $r_{\max} = 4$. Since pooled tokens are analyzed on their native compressed grid without upsampling, their radial profiles terminate at the pooled-grid limit. Query-attention weighted ViT features (denoted by PARCEL–Query and MQT) are computed on the original source grid and therefore retain the higher source-grid frequency support up to the source-grid Nyquist limit.

To form discrete radial bins, we group Fourier coefficients into unit-width rings around the origin. The radial bin at frequency radius r is given by:

$$\mathcal{R}_r = \left\{ (u, v) : \begin{matrix} r - \frac{1}{2} \le \rho(u, v) < r + \frac{1}{2}, \\ \rho(u, v) \le r_{\max}(\mathbf{M}) \end{matrix} \right\}. \tag{11}$$

The radial mean power is then the average PSD value within each ring:

$$P_{\mathbf{M}}(r) = \frac{1}{|\mathcal{R}_r|} \sum_{(u,v) \in \mathcal{R}_r} S_{\mathbf{M}}(u, v). \tag{12}$$

This measures the average spectral power per Fourier coefficient at radius r. Unlike annular energy, it does not give extra weight to high-frequency rings merely because they contain more Fourier coefficients.

For dataset-level curves, we compute $P_{\mathbf{M}}(r)$ per sample and then average the resulting radial profiles:

$$\bar{P}_{\mathbf{M}}(r) = \mathbb{E}_{x \sim \mathcal{D}} \left[P_{\mathbf{M}(x)}(r) \right]. \tag{13}$$

Cumulative radial mean-power concentration. Figure 3(a) analyzes the extent to which the compressed spatial tokens capture the low-frequency baseband. For a compressed spatial grid $\mathbf{Y} \in \mathbb{R}^{H_{\text{out}} \times W_{\text{out}} \times C}$, we first normalize its dataset-averaged radial mean-power profile:

$$\hat{P}_{\mathbf{Y}}(r) = \frac{\bar{P}_{\mathbf{Y}}(r)}{\sum_{k=1}^{r_{\max}(\mathbf{Y})} \bar{P}_{\mathbf{Y}}(k) + \epsilon}, \tag{14}$$

where $\epsilon = 10^{-15}$ is used for numerical stability. We then define the cumulative spectral concentration as:

$$C_{\mathbf{Y}}(r) = \sum_{k=1}^{r} \hat{P}_{\mathbf{Y}}(k). \tag{15}$$

A steeply rising $C_{\mathbf{Y}}(r)$ indicates that most normalized spectral power is concentrated in low spatial frequencies, implying a heavier focus on capturing low-frequency visual features. On the other hand, a slower-rising curve indicates that spectral power is distributed more broadly over the available frequency range, implying that the compressed tokens capture a more spread-out band of feature frequencies.

Normalized radial mean power. Figure 3(b) visualizes the normalized radial mean-power directly:

$$\hat{P}_{\mathbf{M}}(r) = \frac{\bar{P}_{\mathbf{M}}(r)}{\sum_{k=1}^{r_{\max}(\mathbf{M})} \bar{P}_{\mathbf{M}}(k) + \epsilon}. \tag{16}$$

This profile measures relative per-mode frequency concentration independent of total feature magnitude. When the target feature map is the pooled spatial grid, we set $\mathbf{M} = \mathbf{Y}$ in the definition above and obtain $\hat{P}_{\mathbf{Y}}(r)$. For spatial pool tokens, $\hat{P}_{\mathbf{Y}}(r)$ is evaluated only up to the pooled-grid Nyquist radius. For query-based compressors such as MQT (Hu et al., 2024) and PARCEL, query tokens do not possess a native 2D structure because they are learned non-spatial sequence tokens where nested dropout enforces elasticity over this query sequence (Rippel et al., 2014). We therefore analyze their attention-weighted footprint on the ViT output features.

Let $\mathbf{A} \in \mathbb{R}^{N_q \times HW}$ denote the post-softmax query-to-visual attention matrix from N_q query tokens to the ViT output tokens, averaged over attention heads. For query q, the vector $\mathbf{A}_{q,:} \in \mathbb{R}^{HW}$ contains its attention weights over all spatial ViT tokens. We reshape this vector into a 2D attention map $\mathbf{A}_q \in \mathbb{R}^{H \times W}$ and rescale it to unit spatial mean:

$$\tilde{\mathbf{A}}_{q,h,w} = \frac{\mathbf{A}_{q,h,w}}{\frac{1}{HW}\sum_{h'=0}^{H-1}\sum_{w'=0}^{W-1}\mathbf{A}_{q,h',w'} + \epsilon}. \tag{17}$$

The query-attended source feature map is then:

$$\mathbf{Z}^{(q)}_{h,w,c} = \mathbf{X}_{h,w,c}\tilde{\mathbf{A}}_{q,h,w}, \tag{18}$$

where $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ is the source visual grid. We compute the PSD of each $\mathbf{Z}^{(q)}$ and average across queries:

$$S_{\mathbf{Z}}(u,v) = \frac{1}{N_q}\sum_{q=1}^{N_q} S_{\mathbf{Z}^{(q)}}(u,v). \tag{19}$$

We then apply the same radial mean-power computation defined above to $S_{\mathbf{Z}}(u,v)$, followed by the same normalization, to obtain $\hat{P}_{\mathbf{Z}}(r)$. The resulting $\hat{P}_{\mathbf{Z}}(r)$ is interpreted as the normalized radial mean-power profile of the ViT output feature regions weighted by query-token attention, rather than as a direct Fourier transform of the query-token sequence.

Together, these two diagnostics test the spectral role of each token family across both our baselines, M^3 (Cai et al., 2025) and MQT (Hu et al., 2024), and our proposed method, PARCEL. The cumulative curve in Figure 3(a) measures how quickly compressed spatial tokens concentrate their power into low frequencies. The normalized profiles in Figure 3(b) compare pool tokens and query-attended ViT output tokens, allowing us to assess whether spatial tokens serve as low-frequency anchors while query tokens retain access to higher-frequency features.

B Additional Experimental Results and Discussions

This section provides the benchmark-level breakdown supporting the aggregate results in the main paper. In Appendix B.2, we further evaluate the compressed models after 448×448 pretraining following the Stage-2 recipe of PaliGemma and PaliGemma-2 (Beyer et al., 2024; Steiner et al., 2024), providing an additional stress test for settings where spatial detail is especially important. Together, these results offer a more complete view of where PARCEL provides the largest gains and where different compression strategies behave similarly.

B.1 Detailed Results Across All PaliGemma-2 Benchmarks

Tables 5–8 provide the 224×224 PaliGemma-2 evaluation results across video understanding, spatial grounding, resolution-sensitive understanding, and general vision-language benchmarks. All scores

Table 5: **Detailed Performance on Video Understanding Benchmarks.** Raw scores are reported as the mean over three random seeds, with standard deviation shown in gray. Per-benchmark retention values are omitted for readability and aggregate retention relative to Vanilla PG2 is reported in the final row. Vanilla PG2 is shaded in gray and serves as the uncompressed reference. Best results per budget are **bolded**, and second-best are underlined.

		256 Visual Tokens			64 Visual Tokens			16 Visual Tokens		
Benchmark	**PG2**	**M^3**	**MQT**	**PARCEL *(Ours)***	**M^3**	**MQT**	**PARCEL *(Ours)***	**M^3**	**MQT**	**PARCEL *(Ours)***
ActivityNet-CAP	43.7	36.1 $\pm$1.6	37.2 $\pm$1.0	**41.5** $\pm$0.9	36.5 $\pm$0.2	37.1 $\pm$1.3	**40.5** $\pm$1.5	36.5 $\pm$1.0	37.4 $\pm$0.9	**38.9** $\pm$0.4
ActivityNet-QA	53.3	51.5 $\pm$0.3	50.0 $\pm$0.8	**52.4** $\pm$0.7	51.2 $\pm$0.1	49.8 $\pm$0.8	**52.7** $\pm$0.5	50.4 $\pm$0.3	50.4 $\pm$0.9	**50.6** $\pm$0.1
MSRVTT-Cap	70.6	64.2 $\pm$0.9	66.8 $\pm$2.0	**68.6** $\pm$1.6	63.4 $\pm$2.7	64.5 $\pm$2.2	**68.9** $\pm$1.8	62.4 $\pm$1.1	65.7 $\pm$1.8	**66.7** $\pm$0.8
MSRVTT-QA	41.5	40.2 $\pm$0.1	41.5 $\pm$0.6	**42.8** $\pm$0.6	39.9 $\pm$0.2	40.9 $\pm$0.4	**43.0** $\pm$0.6	39.5 $\pm$0.1	40.1 $\pm$0.6	**41.9** $\pm$0.1
MSVD-QA	62.7	60.8 $\pm$1.2	**61.4** $\pm$1.0	60.5 $\pm$1.8	60.7 $\pm$1.2	**62.2** $\pm$0.8	60.6 $\pm$1.6	60.4 $\pm$0.3	**62.6** $\pm$0.7	59.8 $\pm$0.3
Mean Retention	–	92.9%	94.4%	**98.0%**	92.5%	93.5%	**97.9%**	91.6%	94.0%	**95.0%**

are reported as the mean over three random seeds, with standard deviations shown in gray. To keep the tables readable, per-benchmark retention values are omitted, while aggregate retention relative to Vanilla PG2 is reported in the final row of each table.

Overall, PARCEL provides the strongest aggregate retention on the task groups most affected by visual token compression. On video understanding benchmarks, PARCEL achieves the highest mean retention at all token budgets, reaching 98.0%, 97.9%, and 95.0% retention at 256, 64, and 16 tokens, respectively. These results indicate that combining explicit spatial anchors with conditioned queries preserves temporal visual evidence more effectively than either rigid spatial pooling or query-only resampling alone.

The advantage is even more pronounced on spatial grounding benchmarks. Across the RefCOCO suite, PARCEL consistently outperforms both baselines at every budget, reaching 90.6%, 90.8%, and 80.5% retention at 256, 64, and 16 tokens, respectively. At the 64-token anchor-only budget, this exceeds MQT and M^3 by 5.9 and 11.2 points, showing that the pooled representation remains substantially stronger even without query tokens at inference. Furthermore, these results support the role of explicit 2D spatial anchors in preserving layout-sensitive information.

On resolution-sensitive benchmarks, PARCEL achieves the strongest aggregate retention at 256 and 64 tokens. At 256 tokens, PARCEL reaches 96.7% mean retention, improving over MQT (96.3%) and M^3 (94.8%). At 64 tokens, the gap becomes larger, with PARCEL retaining 95.7%, compared to 94.1% for MQT and 92.3% for M^3. At the extreme 16-token budget, M^3 obtains the highest aggregate retention on this group, while PARCEL remains competitive and outperforms MQT.

For general vision-language benchmarks, all compressed models retain a large fraction of Vanilla PG2 performance, suggesting that many of these tasks are less sensitive to aggressive visual token compression. Even in this saturated regime, PARCEL achieves the best aggregate retention at 256 and 64 tokens, reaching 99.4% and 99.2%, respectively, compared with 99.2% and 98.8% for M^3, and 99.1% and 98.6% for MQT. At 16 tokens, M^3 leads with 97.5% retention, while PARCEL retains 96.8% and remains slightly ahead of MQT at 96.7%. Together, these detailed results show that the gains of PARCEL are concentrated where compression is most challenging: video understanding, spatial grounding, and resolution-sensitive reasoning, while remaining competitive or better on broader vision-language benchmarks.

VATEX validation results. Finally, we evaluate both PARCEL and the baselines on VATEX (Wang et al., 2019), but exclude it from aggregate summaries because we observe very high variance on the validation split across all methods and the official test set is not publicly available. For completeness, at 256 tokens, M^3, MQT, and PARCEL obtain $78.7_{\pm 1.1}$, $77.9_{\pm 3.6}$, and $78.4_{\pm 1.8}$, respectively, compared to the Vanilla PG2 reference of 80.5. At 64 tokens, the corresponding scores are $78.8_{\pm 1.5}$, $79.6_{\pm 1.9}$, and $79.5_{\pm 2.6}$, while at 16 tokens they are $79.6_{\pm 1.9}$, $77.2_{\pm 1.1}$, and $77.8_{\pm 2.7}$. Because the observed method differences are comparable to the seed-level variation on this validation-only benchmark, we report these numbers for transparency but exclude VATEX from aggregate retention calculations.

Table 6: **Detailed Performance on Image Segmentation (RefCOCO) Benchmarks.** Raw scores are reported as the mean over three random seeds, with standard deviation shown in gray. Per-benchmark retention values are omitted for readability and aggregate retention relative to Vanilla PG2 is reported in the final row. Vanilla PG2 is shaded in gray and serves as the uncompressed reference. Best results per budget are **bolded**, and second-best are underlined.

		256 Visual Tokens			64 Visual Tokens			16 Visual Tokens		
Benchmark	**PG2**	**M^3**	**MQT**	**PARCEL** ***(Ours)***	**M^3**	**MQT**	**PARCEL** ***(Ours)***	**M^3**	**MQT**	**PARCEL** ***(Ours)***
RefCOCO (testA)	71.8	59.6 ±0.4	63.4 ±0.2	**65.2** ±0.2	58.2 ±0.3	62.6 ±0.2	**65.4** ±0.2	54.5 ±0.2	**57.7** ±0.4	57.6 ±0.1
RefCOCO (testB)	65.2	56.1 ±0.2	58.6 ±0.4	**61.7** ±0.4	55.0 ±0.4	58.1 ±0.5	**61.9** ±0.4	51.9 ±0.2	55.0 ±0.5	**56.1** ±0.1
RefCOCO (val)	67.2	57.7 ±0.4	61.0 ±0.2	**63.3** ±0.1	56.4 ±0.3	60.2 ±0.2	**63.5** ±0.3	52.7 ±0.3	56.1 ±0.4	**56.3** ±0.1
RefCOCO+ (testA)	69.5	55.7 ±0.5	59.2 ±0.2	**61.9** ±0.3	54.3 ±0.3	58.0 ±0.3	**61.9** ±0.3	50.0 ±0.1	52.6 ±0.5	**53.6** ±0.3
RefCOCO+ (testB)	61.4	49.3 ±0.6	51.5 ±0.5	**54.7** ±0.1	47.4 ±0.6	50.1 ±0.4	**54.8** ±0.2	44.4 ±0.5	47.2 ±0.0	**49.0** ±0.4
RefCOCO+ (val)	65.3	52.8 ±0.4	55.9 ±0.2	**58.7** ±0.3	51.3 ±0.3	54.5 ±0.1	**58.7** ±0.3	47.4 ±0.2	50.3 ±0.4	**51.5** ±0.1
RefCOCO-g (test)	65.4	51.3 ±0.3	55.0 ±0.1	**57.8** ±0.1	50.1 ±0.4	53.7 ±0.3	**57.9** ±0.1	46.8 ±0.2	50.4 ±0.0	**51.5** ±0.1
RefCOCO-g (val)	64.9	51.1 ±0.6	54.3 ±0.3	**57.6** ±0.1	49.8 ±0.5	53.3 ±0.4	**57.7** ±0.3	46.4 ±0.1	49.9 ±0.2	**51.4** ±0.1
Mean Retention	–	81.7%	86.4%	**90.6%**	79.6%	84.9%	**90.8%**	74.2%	79.0%	**80.5%**

Table 7: **Detailed Performance on Resolution-Sensitive Benchmarks.** Raw scores are reported as the mean over three random seeds, with standard deviation shown in gray. Per-benchmark retention values are omitted for readability and aggregate retention relative to Vanilla PG2 is reported in the final row. Vanilla PG2 is shaded in gray and serves as the uncompressed reference. Best results per budget are **bolded**, and second-best are underlined.

		256 Visual Tokens			64 Visual Tokens			16 Visual Tokens		
Benchmark	**PG2**	**M^3**	**MQT**	**PARCEL** ***(Ours)***	**M^3**	**MQT**	**PARCEL** ***(Ours)***	**M^3**	**MQT**	**PARCEL** ***(Ours)***
ChartQA (aug)	72.5	64.1 ±0.6	66.0 ±0.5	**66.2** ±0.8	63.0 ±1.3	64.8 ±0.2	**66.3** ±0.2	**59.9** ±0.8	58.2 ±0.9	58.8 ±0.9
ChartQA (human)	40.7	33.8 ±0.3	35.6 ±0.8	**37.2** ±0.6	32.3 ±1.0	34.1 ±0.9	**37.0** ±0.3	29.7 ±0.7	30.7 ±0.3	**32.0** ±1.0
DocVQA (val)	36.6	30.5 ±0.3	**33.4** ±0.1	32.8 ±0.3	28.7 ±0.1	31.2 ±0.5	**32.1** ±0.4	25.9 ±0.2	24.4 ±0.1	**26.1** ±0.1
InfoVQA (val)	24.8	**25.0** ±0.2	24.2 ±0.3	24.4 ±0.3	23.4 ±0.3	23.6 ±0.2	**23.9** ±0.5	**22.9** ±0.2	22.3 ±0.3	22.2 ±0.2
SciCap	163.5	159.6 ±0.7	163.8 ±0.5	**164.4** ±2.0	159.9 ±0.7	**164.2** ±0.2	163.6 ±2.7	159.1 ±0.1	**160.4** ±0.5	159.7 ±2.8
ST-VQA (val)	59.9	**60.4** ±0.2	59.7 ±0.3	59.1 ±0.1	**59.3** ±0.1	58.1 ±0.5	58.9 ±0.1	**56.8** ±0.3	54.9 ±0.1	54.5 ±0.4
TextCaps	124.6	123.2 ±1.0	125.0 ±0.2	**125.8** ±0.9	121.9 ±0.5	123.4 ±0.8	**123.8** ±0.5	**115.8** ±0.5	114.1 ±0.3	113.0 ±0.7
TextVQA (val)	57.7	**57.9** ±0.2	57.5 ±0.1	57.2 ±0.2	**56.6** ±0.1	56.5 ±0.3	56.6 ±0.1	**54.3** ±0.5	52.8 ±0.3	52.1 ±0.2
WidgetCap	133.8	133.9 ±0.2	133.5 ±1.1	**134.4** ±0.9	132.9 ±0.2	131.9 ±0.6	**133.4** ±1.1	**130.9** ±1.1	129.4 ±0.4	128.1 ±1.3
Mean Retention	–	94.8%	96.3%	**96.7%**	92.3%	94.1%	**95.7%**	**88.4%**	86.8%	87.3%

B.2 Detailed Results for Benchmarks at High Resolution

In this section, we present additional results after pretraining PARCEL and the baselines (Hu et al., 2024; Cai et al., 2025) with the PaliGemma-2 high-resolution Stage-2 recipe at 448×448 resolution (Beyer et al., 2024; Steiner et al., 2024). Due to the substantially higher cost of high-resolution pretraining and evaluation, these results are reported from a single seed and exclude video benchmarks.

Table 9 reports detailed results for high-resolution 448×448 PaliGemma-2 evaluations across 1024 (full budget), 256, and 64 visual token budgets. We therefore use this analysis as a high-resolution stress test of compression behavior rather than as a replacement for the three-seed default-resolution evaluation.

Overall, PARCEL retains the strongest aggregate performance across all high-resolution token budgets. At 1024 visual tokens, PARCEL reaches 98.2% mean retention relative to the 448×448 Vanilla PG2 reference, compared to 96.0% for M^3 and 95.4% for MQT. At 256 tokens, PARCEL again achieves 98.2% mean retention, outperforming MQT (95.8%) and M^3 (95.4%). Even under the more constrained 64-token budget, PARCEL remains the strongest compressed model with 95.4% mean retention, compared to 93.5% for MQT and 93.4% for M^3. Notably, the 256- and 64-token settings are both anchor-only budgets for PARCEL at this resolution. At both budgets, PARCEL

Table 8: **Detailed Performance on General Vision-Language Benchmarks.** Raw scores are reported as the mean over three random seeds, with standard deviation shown in gray. Per-benchmark retention values are omitted for readability and aggregate retention relative to Vanilla PG2 is reported in the final row. Vanilla PG2 is shaded in gray and serves as the uncompressed reference. Standard deviations are rounded to one decimal and 0.0 denotes a value below 0.05. Best results per budget are **bolded**, and second-best are underlined.

Benchmark	PG2	256 Visual Tokens			64 Visual Tokens			16 Visual Tokens		
		M^3	MQT	PARCEL *(Ours)*	M^3	MQT	PARCEL *(Ours)*	M^3	MQT	PARCEL *(Ours)*
AI2D	74.4	71.6 ±0.5	72.9 ±0.8	**74.2** ±0.4	71.1 ±0.5	72.3 ±0.6	**74.1** ±0.3	69.7 ±0.3	69.8 ±0.6	**71.4** ±0.7
AOKVQA-DA (val)	62.4	61.6 ±0.5	**62.6** ±0.3	61.7 ±0.7	61.7 ±0.4	**62.2** ±0.2	61.7 ±0.7	**61.2** ±0.3	60.9 ±0.9	60.3 ±0.4
AOKVQA-MC (val)	79.7	77.6 ±0.6	**78.6** ±0.6	78.2 ±0.1	77.4 ±0.7	**78.5** ±0.5	77.6 ±0.3	75.0 ±1.9	**76.6** ±0.6	74.9 ±0.7
COCO-35L (en)	138.1	138.6 ±0.4	**139.0** ±0.5	138.8 ±0.5	**138.4** ±0.3	138.3 ±0.3	138.2 ±0.5	**138.1** ±0.5	136.7 ±0.7	134.9 ±0.3
COCO-Captions	141.1	140.2 ±0.1	**140.5** ±0.7	**140.5** ±0.2	139.6 ±0.4	139.9 ±0.5	**140.5** ±0.3	**138.2** ±0.3	137.2 ±0.6	136.0 ±0.4
CountBenchQA	80.2	**81.0** ±1.1	79.7 ±2.4	80.1 ±0.7	80.3 ±1.3	79.1 ±2.1	**80.5** ±0.9	77.6 ±0.6	**79.1** ±0.9	78.9 ±1.7
GQA	65.6	**65.1** ±0.1	64.7 ±0.3	65.0 ±0.2	**64.7** ±0.4	64.3 ±0.5	**64.7** ±0.1	**63.5** ±0.4	62.2 ±0.4	62.7 ±0.1
NLVR2	90.8	89.9 ±0.2	89.2 ±0.4	**90.1** ±0.3	89.8 ±0.1	88.9 ±0.3	**90.3** ±0.5	88.7 ±0.0	86.8 ±0.2	**88.8** ±0.3
NoCaps	122.2	121.9 ±0.3	121.2 ±0.2	**122.0** ±0.3	120.8 ±0.3	120.9 ±0.2	**121.6** ±0.6	**119.0** ±0.3	118.7 ±0.2	117.6 ±0.7
OCR-VQA	72.2	**72.1** ±0.0	**72.1** ±0.0	**72.1** ±0.1	71.6 ±0.0	71.5 ±0.0	**71.7** ±0.0	**69.5** ±0.0	67.7 ±0.1	67.3 ±0.2
OKVQA	63.4	62.9 ±0.3	62.6 ±0.4	**63.3** ±0.1	62.2 ±0.2	62.1 ±0.4	**62.8** ±0.4	**61.5** ±0.1	61.3 ±0.3	61.2 ±0.4
RSVQA-hr (test)	92.6	**92.7** ±0.1	**92.7** ±0.0	**92.7** ±0.1	**92.7** ±0.1	**92.7** ±0.0	**92.7** ±0.1	**92.7** ±0.1	**92.7** ±0.0	92.6 ±0.0
RSVQA-hr (test2)	90.6	**90.8** ±0.2	90.5 ±0.1	90.7 ±0.0	**90.8** ±0.2	90.5 ±0.2	**90.8** ±0.1	**90.7** ±0.1	90.4 ±0.2	90.5 ±0.2
RSVQA-lr	92.7	92.8 ±1.0	92.5 ±0.3	**93.0** ±0.5	93.2 ±0.9	92.7 ±0.7	**93.3** ±0.2	**93.4** ±0.9	92.5 ±0.5	92.6 ±0.5
Screen2Words	113.5	112.3 ±1.4	112.5 ±1.1	**113.2** ±0.7	112.2 ±1.5	111.6 ±0.5	**112.8** ±1.0	**111.6** ±0.7	109.7 ±0.5	111.4 ±0.8
TallyQA (complex)	69.9	**68.5** ±0.0	67.5 ±0.1	68.1 ±0.2	67.4 ±0.4	66.0 ±0.3	**67.9** ±0.4	**65.7** ±0.4	64.5 ±0.6	65.0 ±0.3
TallyQA (simple)	81.0	**80.5** ±0.1	80.0 ±0.2	80.4 ±0.0	80.0 ±0.0	79.6 ±0.1	**80.4** ±0.1	**79.3** ±0.0	78.2 ±0.1	78.7 ±0.2
ScienceQA	96.5	94.7 ±1.1	**95.7** ±0.6	**95.7** ±0.3	94.4 ±1.3	95.5 ±0.6	**95.7** ±0.2	94.2 ±1.3	**94.5** ±0.6	94.3 ±0.4
VQAv2 (minival)	82.6	**81.8** ±0.2	81.7 ±0.2	81.6 ±0.3	81.1 ±0.1	81.2 ±0.1	**81.6** ±0.2	**79.8** ±0.1	79.3 ±0.1	78.6 ±0.1
VizWizVQA (val)	75.9	**76.0** ±0.2	75.6 ±0.7	**76.0** ±0.1	75.6 ±0.4	75.5 ±0.4	**75.7** ±0.3	**75.1** ±0.4	74.6 ±0.3	74.3 ±0.3
XM3600 (en)	80.3	79.6 ±0.2	**80.1** ±0.4	78.5 ±0.4	**79.7** ±0.3	79.3 ±0.5	78.6 ±0.3	**79.4** ±1.0	77.7 ±0.2	77.5 ±0.1
xGQA	57.4	**57.3** ±0.4	**57.3** ±0.2	56.9 ±0.2	**56.9** ±0.3	56.8 ±0.1	56.6 ±0.2	**55.6** ±0.4	54.8 ±0.3	54.4 ±0.1
MARVL-5	81.9	**81.4** ±0.3	79.1 ±0.5	**81.4** ±0.2	80.5 ±0.1	79.1 ±0.7	**81.1** ±0.5	79.7 ±0.2	77.6 ±0.5	**79.9** ±0.4
Mean Retention	–	99.2%	99.1%	**99.4%**	98.8%	98.6%	**99.2%**	**97.5%**	96.7%	96.8%

outperforms M^3 and MQT, showing that the benefit of the coupled design persists even when the pooled pathway operates alone.

The detailed breakdown shows that the advantage of PARCEL is most pronounced on the task families targeted by our design. On image segmentation benchmarks, PARCEL consistently improves over both baselines across RefCOCO, RefCOCO+, and RefCOCO-g splits, supporting the role of explicit spatial anchors for preserving layout-sensitive evidence. On resolution-sensitive tasks such as ChartQA, DocVQA, InfoVQA, and TextCaps, PARCEL also achieves strong retention, indicating that the hybrid decomposition remains effective when visual inputs are processed at higher spatial resolution. For general multimodal benchmarks, the gaps are smaller because many tasks already saturate near the Vanilla PG2 reference, but PARCEL still provides the best aggregate trade-off. These results show that the gains from spatial anchoring and Pool-Conditioned Query Resampling extend to high-resolution visual encoding.

B.3 Comparisons Against Training-Free Baselines with LLaVA 1.5

We further compare PARCEL with three representative training-free visual token compression methods on LLaVA 1.5. LLaVA-PruMerge performs adaptive token selection and merging before the language model, FastV prunes visual tokens inside the language model, and AIM combines token merging before the language model with progressive pruning across its layers. Together, these methods cover complementary forms of training-free compression operating at different stages of the LVLM pipeline.

We evaluate the low-compute configurations considered in Table 3 of Zhong et al. (2025) alongside PARCEL at 36 tokens. Since the training-free methods may retain different numbers of tokens across inputs or language-model layers, their nominal visual token budgets do not directly determine their prefill cost. We therefore report the published prefill costs of these configurations alongside

Table 9: **Detailed Performance on High-Resolution** 448×448 **Benchmarks.** Raw scores are reported for single-seed high-resolution fine-tuning runs due to computational cost. Per-benchmark retention values are omitted for readability and aggregate retention relative to the 448×448 Vanilla PG2 reference is reported in the final row. Vanilla PG2 is shaded in gray and serves as the uncompressed reference. Best results per budget are **bolded**, and second-best are underlined.

		1024 Visual Tokens			256 Visual Tokens			64 Visual Tokens		
Benchmark	**PG2**	**M^3**	**MQT**	**PARCEL**	**M^3**	**MQT**	**PARCEL**	**M^3**	**MQT**	**PARCEL**
Resolution-Sensitive Benchmarks										
ChartQA (aug)	88.3	85.0	83.5	**85.8**	83.4	84.3	**85.6**	82.3	81.8	**82.7**
ChartQA (human)	53.2	48.2	48.1	**50.4**	48.2	48.7	**50.2**	45.6	46.2	**47.5**
DocVQA (val)	69.8	62.9	61.8	**63.7**	57.8	60.4	**63.7**	50.3	49.4	**52.5**
InfoVQA (val)	35.3	**35.1**	31.8	34.0	33.2	30.9	**34.2**	29.5	27.5	**29.5**
SciCap	182.3	177.9	179.6	**183.5**	177.2	181.1	**183.4**	177.2	181.2	**181.8**
ST-VQA	78.6	**79.9**	77.0	78.8	78.9	77.9	**78.9**	76.7	75.7	**76.9**
TextCaps	146.2	145.7	146.2	**149.1**	146.1	146.7	**149.2**	144.1	142.1	**145.7**
TextVQA	72.8	**73.2**	72.0	72.6	**72.9**	71.9	72.7	**71.0**	70.0	69.9
WidgetCap	148.8	147.1	146.1	**147.9**	146.1	147.4	**148.1**	144.2	**145.6**	145.1
Image Segmentation (RefCOCO)										
RefCOCO (testA)	76.4	67.1	67.8	**72.7**	66.8	68.6	**72.8**	64.8	67.2	**69.2**
RefCOCO (testB)	71.5	62.1	63.4	**67.4**	62.1	64.3	**67.6**	60.6	63.4	**64.7**
RefCOCO (val)	69.6	64.9	65.7	**69.9**	64.6	66.5	**69.9**	62.9	64.9	**66.8**
RefCOCO+ (testA)	74.0	64.2	64.1	**69.8**	63.3	65.3	**69.5**	60.7	62.8	**65.5**
RefCOCO+ (testB)	65.0	56.0	56.2	**61.1**	55.3	56.4	**61.0**	53.8	54.4	**58.0**
RefCOCO+ (val)	69.8	60.4	59.9	**65.6**	59.7	60.9	**65.8**	58.0	58.7	**62.4**
RefCOCO-g (test)	70.0	59.1	60.1	**64.8**	58.6	61.1	**64.7**	56.7	59.1	**61.7**
RefCOCO-g (val)	69.6	59.1	59.3	**64.7**	58.6	60.0	**64.4**	56.5	58.0	**61.1**
General Vision-Language Understanding										
AI2D	75.1	74.9	73.7	**75.5**	75.5	74.1	**75.8**	**75.0**	73.0	74.6
AOKVQA-DA	65.2	64.6	64.7	**65.5**	64.9	65.0	**65.5**	63.3	63.5	**64.7**
AOKVQA-MC	80.8	80.1	80.8	**81.7**	80.2	**81.3**	81.2	79.5	79.9	**81.0**
COCO-35L	140.4	**141.4**	140.2	141.1	**141.9**	140.8	141.0	141.2	139.3	**141.3**
COCO-Captions	142.1	142.2	142.3	**143.5**	142.1	142.1	**142.9**	141.9	141.4	**142.0**
GQA	67.7	67.3	66.0	**67.4**	**67.3**	65.9	67.0	**66.4**	65.6	66.2
NLVR2	91.1	89.8	88.4	**89.8**	**90.0**	89.2	89.8	**89.8**	88.5	89.3
NoCaps	123.5	**121.8**	121.1	121.5	122.4	**122.5**	121.7	121.1	**121.6**	121.5
OCR-VQA	74.9	74.7	74.4	**74.9**	74.7	74.7	**74.8**	**74.2**	73.8	74.0
OKVQA	63.7	**63.0**	62.0	63.0	62.6	62.1	**63.0**	61.5	61.7	**62.5**
RSVQA-hr	92.9	92.8	92.8	**92.8**	**92.9**	92.8	92.9	**92.9**	92.8	92.8
RSVQA-hr (test2)	90.8	90.7	**90.8**	90.6	90.7	**90.8**	90.6	**90.8**	**90.8**	90.6
ScienceQA	96.0	**95.7**	95.5	95.6	**95.8**	95.5	95.7	**95.4**	94.8	95.2
Screen2Words	114.2	115.1	113.1	**115.7**	115.6	114.5	**116.7**	115.1	113.8	**116.1**
VQAv2 (val)	84.2	83.4	82.9	**84.0**	83.3	82.9	**84.0**	82.7	82.4	**83.4**
VizWizVQA	77.1	76.2	76.8	**77.0**	76.8	76.5	**77.2**	76.6	76.0	**76.7**
XM3600	80.0	81.0	80.1	**81.0**	80.7	79.8	**81.2**	79.6	78.2	**80.8**
Mean Retention	–	96.0%	95.4%	**98.2%**	95.4%	95.8%	**98.2%**	93.4%	93.5%	**95.4%**

our reproduced accuracy results, distinguishing these costs from measurements of the reproduced implementations. We reproduce the training-free results using the released implementations and evaluate them on GQA, MME-P, TextVQA, SQA$^{\text{IMG}}$, and POPE. The PARCEL scores are taken from Table 3. For consistency with the main paper, all retention values in this comparison use the same uncompressed LLaVA 1.5 reference as Table 3.

Table 10 shows that PARCEL achieves the highest mean retention among the evaluated configurations. At its 36-token anchor-only budget, PARCEL retains 94.7% of the uncompressed reference performance, compared with 88.8% for AIM, 87.0% for LLaVA-PruMerge, and 71.7% for FastV. This corresponds to mean-retention improvements of 5.9, 7.7, and 23.0 points, respectively. Across the individual benchmarks, PARCEL achieves the highest scores on GQA, MME-P, and POPE and the second-highest scores on TextVQA and SQA$^{\text{IMG}}$. We also note that the advantage over PruMerge

Table 10: **Comparison with Training-Free Compression on LLaVA 1.5.** We compare the 36-token anchor-only configuration of PARCEL with the low-compute configurations of AIM, FastV, and PruMerge. Training-free accuracy results are reproduced using the released implementations. The baseline prefill FLOPs are reported by Zhong et al. (2025), while the PARCEL value was calculated using the same prefill TFLOP convention as Zhong et al. (2025). Mean Retention averages per-benchmark retention across the five benchmarks relative to the shaded uncompressed reference from Table 3. Retentions and rankings are computed before rounding. Best results among all compressed models, including PARCEL, are **bolded**, and second-best results are underlined.

Method	Prefill TFLOPs	GQA	MME-P	TextVQA	SQA^{IMG}	POPE	Mean Retention
LLaVA 1.5	8.18	61.9	1505.7	58.2	69.5	85.9	100.0%
AIM (Zhong et al., 2025)	1.00	54.5	1252.1	48.8	66.9	79.5	88.8%
FastV (Chen et al., 2024b)	1.12	45.6	946.0	51.3	66.1	33.3	71.7%
PruMerge (Shang et al., 2025)	1.04	51.2	1249.4	**54.0**	**68.5**	67.0	87.0%
PARCEL @ 36 *(Ours)*	0.99	**60.7**	**1424.7**	51.4	67.0	**82.8**	**94.7%**

persists even after substituting its paper-reported POPE score[1], which increases its mean retention to 89.1%, still below PARCEL's 94.7%. These results extend the comparison beyond elastic matryoshka baselines and show that the jointly trained pooled representation remains effective under aggressive compression, even when no query tokens are used at inference.

C Efficiency Measures and Calculations

In this section, we describe the specifics of both the theoretical FLOP and KV-cache calculations we report in our work (Appendix C.1) and also the latency implications of PARCEL compared to the vanilla LVLMs and MQT where we show that PARCEL brings negligible overhead while bringing similar or less computational overhead with respect to MQT (Appendix C.2).

C.1 FLOP and KV-Cache Calculations

This section details the theoretical FLOP and KV-cache calculations used for the efficiency analysis in Figure 1. As we utilized PaliGemma-2 for Figure 1, all calculations below use the PaliGemma-2 3B configuration. We estimate inference-time compute for the visual encoder, visual connector, cross-modal projection, language decoder, and output head. Following standard transformer FLOP accounting, one multiply-add is counted as two FLOPs. Lower-order operations such as normalization, activation functions, positional operations, and softmax normalization are omitted. Finally, our calculations also account for the text-prefix tokens rather than omitting them, providing a more realistic estimate of the full forward-pass compute for both Vanilla PaliGemma-2 and PARCEL.

Architectural constants. We use the PaliGemma-2 3B configuration (Steiner et al., 2024), which consists of a SigLIP-So400M vision encoder (Zhai et al., 2023) and a Gemma-2 2B language decoder (Team et al., 2024). We note that the vision encoder has $L_v = 27$ transformer layers, hidden dimension $D_v = 1152$, and MLP dimension $M_v = 4304$. Furthermore, the language decoder has $L_l = 26$ layers, hidden dimension $D_l = 2304$, MLP dimension $M_l = 9216$, query heads $H_q = 8$, key-value heads $H_{kv} = 4$, head dimension $d_h = 256$, and vocabulary size $V = 257152$.

Token-count notation. For a single image, $T = 1$ and for the video setting, $T = 16$. At the default 224×224 resolution, the SigLIP encoder produces $N_v = 256$ visual tokens per frame. Let B denote the compressed visual token budget per frame. For PARCEL, the budget is decomposed into N_p spatial anchor tokens and N_q query tokens:

$$B = N_{\text{vis}} = N_p + N_q. \tag{20}$$

[1]Our reproduction obtains 66.98 mean POPE-F1. We found a similar issue in the official LLaVA-PruMerge repository, where an independent user also reported $\approx$ 67 POPE-F1: https://github.com/42Shawn/LLaVA-PruMerge/issues/15.

For the default-resolution PaliGemma-2 configuration used in Figure 1, the general routing rule in Section 3.3 boils down to:

$$(N_p, N_q) = \begin{cases} (16, B-16), & 16 \le B < 64, \\ (64, B-64), & 64 \le B \le 256. \end{cases} \tag{21}$$

Then, for T frames, the number of compressed visual tokens entering the language decoder is:

$$N_{\text{img}} = TB. \tag{22}$$

Furthermore, let N_t denote the number of text-prefix tokens. These then result in the following full prefill sequence length:

$$N_{\text{tot}} = N_{\text{img}} + N_t. \tag{23}$$

For the values reported in Figure 1, we use $N_t = 128 + 1$ for image inputs and $N_t = 64 + 1$ for 16-frame video inputs, following the official PaliGemma parameters for the image and video evals (Beyer et al., 2024; Steiner et al., 2024).

Vision encoder FLOPs. Each frame is independently encoded by the SigLIP vision encoder. For one ViT layer, the Q, K, V projections and output projection cost $8N_vD_v^2$, the attention matrix products cost $4N_v^2D_v$, and the two-layer MLP costs $4N_vD_vM_v$. Accordingly, the per-frame vision-encoder cost is:

$$C_{\text{ViT}}^{\text{frame}} = L_v \left(8N_vD_v^2 + 4N_v^2D_v + 4N_vD_vM_v\right), \tag{24}$$

Thus, the total vision-encoder cost is given by:

$$C_{\text{ViT}} = TC_{\text{ViT}}^{\text{frame}}. \tag{25}$$

Visual connector FLOPs. For the compact efficiency table, we report the PARCEL connector cost. At matched visual token budgets, M^3, MQT, and PARCEL share the same dominant ViT and LLM costs; their FLOPs differ only in the relatively small connector terms. For PARCEL, the connector consists of Query $\leftrightarrow$ Pool self-attention followed by Query $\rightarrow$ ViT cross-attention whenever $N_q > 0$. When $N_q = 0$, the routing naturally reduces to a spatial-anchor-only representation and the query pathway is inactive.

For $N_q > 0$, the Query $\leftrightarrow$ Pool self-attention over B compressed tokens costs:

$$C_{\text{QP}} = 8BD_v^2 + 4B^2D_v. \tag{26}$$

The Query $\rightarrow$ ViT cross-attention uses N_q query tokens to attend to the N_v original ViT tokens:

$$C_{\text{Q}\rightarrow\text{V}} = 4(N_q + N_v)D_v^2 + 4N_qN_vD_v. \tag{27}$$

Furthermore, query-token MLP cost is given by:

$$C_{\text{Q-MLP}} = 4N_qD_vM_v. \tag{28}$$

Following from these, the total connector cost is thus given by:

$$C_{\text{conn}} = T \cdot \mathbb{1}[N_q > 0] \left(C_{\text{QP}} + C_{\text{Q}\rightarrow\text{V}} + C_{\text{Q-MLP}}\right), \tag{29}$$

where $\mathbb{1}[N_q > 0]$ indicates that the query pathway is active only when query tokens are allocated.

Cross-modal projection FLOPs. After compression, the B visual tokens per frame are projected from the vision dimension D_v to the language dimension D_l:

$$C_{\text{proj}} = 2TBD_vD_l. \tag{30}$$

Language decoder FLOPs. The visual tokens and text prefix are processed by the Gemma-2 language decoder during prefill. We use the full prefix-attention cost:

$$A_{\text{prefill}} = N_{\text{tot}}^2. \tag{31}$$

For one Gemma-2 decoder layer with grouped-query attention, the projection cost is:

$$C_{\text{GQA-proj}} = 4N_{\text{tot}}D_l d_h(H_q + H_{kv}), \tag{32}$$

and the attention matrix cost is:

$$C_{\text{GQA-attn}} = 4A_{\text{prefill}}H_q d_h. \tag{33}$$

The gated feed-forward network cost is:

$$C_{\text{FFN}} = 6N_{\text{tot}}D_l M_l. \tag{34}$$

The total language-decoder cost is:

$$C_{\text{LLM}} = L_l \left(C_{\text{GQA-proj}} + C_{\text{GQA-attn}} + C_{\text{FFN}}\right). \tag{35}$$

Output head FLOPs. For the reported FLOP values, we evaluate vocabulary logits over the text-prefix positions. Thus, with $N_{\text{logit}} = N_t$, the output projection cost is:

$$C_{\text{head}} = 2N_t D_l V. \tag{36}$$

Total theoretical FLOPs. The total theoretical prefill compute is:

$$C_{\text{total}} = C_{\text{ViT}} + C_{\text{conn}} + C_{\text{proj}} + C_{\text{LLM}} + C_{\text{head}}. \tag{37}$$

Substituting the constants above and rounding to one decimal place gives the TFLOP values reported in Figure 1.

KV-cache memory. We compute KV-cache memory for the language decoder, which is the dominant autoregressive memory term during generation. Gemma-2 uses grouped-query attention with $H_{kv} = 4$ key-value heads. Assuming `bfloat16` cache storage, each scalar requires 2 bytes. The number of bytes required to store the key and value cache for one token across all decoder layers is:

$$B_{\text{token}} = 2 \times 2 \times L_l \times H_{kv} \times d_h, \tag{38}$$

where the first factor of 2 is the number of bytes per `bfloat16` scalar and the second factor of 2 accounts for keys and values. Substituting the Gemma-2 constants gives:

$$B_{\text{token}} = 2 \times 2 \times 26 \times 4 \times 256 = 106{,}496 \text{ bytes}. \tag{39}$$

Thus, the prefill KV-cache memory in MB is:

$$M_{\text{KV}} = \frac{N_{\text{tot}}B_{\text{token}}}{1024^2}. \tag{40}$$

For image inputs, $N_{\text{tot}} = B + 129$ and for 16-frame video inputs, $N_{\text{tot}} = 16B + 65$. Rounding to the nearest MB gives the KV-cache values reported in Figure 1.

C.2 Measured Prefill Latency and Memory

In addition to the theoretical FLOP and KV-cache calculations above, we measure wall-clock prefill latency and peak memory on a single NVIDIA A100 GPU. Measurements use bfloat16 precision and a batch size of one at visual token budgets of 36, 64, 144, and 256.

At a matched visual token budget, PARCEL and MQT pass the same number of tokens to the LLM. Thus, LLM prefill shapes and KV-cache sizes are identical, and the difference in inference cost comes from the visual connector.

Across the evaluated budgets, the PARCEL connector accounts for only 0.4–2.6% (depending on the budget, whether query pathway is utilized *etc.*) of full-model prefill latency and requires at most 0.12 GB of peak memory. At the 36- and 144-token anchor-only budgets, the query pathway is inactive, and PARCEL reduces full-model prefill latency by 1.3% and 1.2% relative to MQT, respectively. At the 64- and 256-token query-active budgets, PCQR increases full-model prefill latency by only 1.0% and 0.8%, respectively. These measurements show that the PCQR overhead is small relative to the full LVLM and that PARCEL has comparable or lower inference cost than MQT, depending on the budget.

D Benchmark Details

In this section, we describe and detail the benchmarks used for evaluating PARCEL and baselines on the PaliGemma-2 benchmark suite (Appendix D.1) and LLaVA 1.5 benchmark suite (Appendix D.2).

D.1 PaliGemma-2 Benchmark Suite

For PG2 experiments, we follow the broad transfer-evaluation protocol used in PaliGemma and PaliGemma-2 (Beyer et al., 2024; Steiner et al., 2024), covering video understanding, dense spatial grounding, resolution-sensitive reasoning, captioning, and general multimodal comprehension. Below, we briefly describe the role of each benchmark in our evaluation suite.

27 Benchmark Suite. We evaluate on 27 vision-centric benchmarks spanning video understanding, dense spatial grounding, resolution-sensitive reasoning, and general multimodal comprehension. Video tasks include ActivityNet-Cap (Krishna et al., 2017), ActivityNet-QA (Yu et al., 2019), MSRVTT-Cap (Xu et al., 2016), MSRVTT-QA (Xu et al., 2017), and MSVD-QA (Xu et al., 2017; Chen & Dolan, 2011). Dense spatial grounding is evaluated on RefCOCO, RefCOCO+, and RefCOCO-g (Kazemzadeh et al., 2014; Yu et al., 2016; Mao et al., 2016). Following the PaliGemma evaluation protocol (Beyer et al., 2024), we evaluate fine-grained visual reasoning using nine resolution-sensitive splits, including ChartQA human/augmented splits (Masry et al., 2022), DocVQA (Mathew et al., 2021), InfoVQA (Mathew et al., 2022), SciCap (Hsu et al., 2021), ST-VQA (Biten et al., 2019), TextCaps (Sidorov et al., 2020), TextVQA (Singh et al., 2019), and Widget-Cap (Li et al., 2020). To increase benchmark diversity, we additionally include GQA/xGQA (Hudson & Manning, 2019; Pfeiffer et al., 2022), NLVR2/MARVL5 (Suhr et al., 2019; Liu et al., 2021), and OCR-VQA (Mishra et al., 2019). Aggregate retention in Figure 1, Table 1 and Table 4 is computed over this 27-benchmark main evaluation suite.

Video understanding. ActivityNet-CAP (Krishna et al., 2017) evaluates dense video captioning, requiring the model to summarize human activities and events from short video clips. ActivityNet-QA (Yu et al., 2019) evaluates question answering over video content, stressing temporal understanding and action-level reasoning. MSRVTT-CAP (Xu et al., 2016) measures video captioning on diverse web videos, testing whether compressed visual tokens preserve enough temporal and semantic context for generation. MSRVTT-QA (Xu et al., 2017) evaluates open-ended question answering on MSRVTT videos. MSVD-QA (Xu et al., 2017; Chen & Dolan, 2011) similarly targets video question answering, with a focus on short clips and object/action recognition. VATEX (Wang et al., 2019) is a multilingual video captioning benchmark built around human-annotated video descriptions. Since the official test set is not publicly available and we observe high validation-set variance, we report VATEX only in Section B. Overall, for the video benchmarks, there were slight changes in data splits with respect to the official PaliGemma works (Beyer et al., 2024; Steiner et al., 2024) due to data wipeouts associated with these datasets.

Dense spatial grounding and segmentation. The RefCOCO suite evaluates referring expression segmentation, where the model must localize the image region described by a natural-language expression. RefCOCO and RefCOCO+ (Kazemzadeh et al., 2014; Yu et al., 2016) differ in the style of referring expressions, with RefCOCO+ reducing reliance on absolute location words and therefore requiring stronger visual grounding. RefCOCO-g (Mao et al., 2016) contains longer and more descriptive referring expressions, making it a stronger test of language-conditioned dense localization. Together, these splits directly probe whether visual token compression preserves explicit spatial layout and fine-grained object boundaries.

Resolution-sensitive document and chart understanding and OCR tasks. ChartQA (Masry et al., 2022) evaluates question answering over charts, with separate augmented and human-written splits. DocVQA (Mathew et al., 2021) evaluates visual question answering over document images, stressing text recognition, layout understanding, and fine-grained evidence retrieval. InfoVQA (Mathew et al., 2022) extends document VQA to infographic-style inputs, where information is often distributed across text, icons, tables, and visual layouts. ST-VQA (Biten et al., 2019) and TextVQA (Singh et al., 2019) evaluate scene-text understanding, requiring the model to read and reason over text embedded in natural images. TextCaps (Sidorov et al., 2020) evaluates caption generation that must incorporate scene text, testing whether the model can preserve text-sensitive visual evidence under compression. WidgetCap (Li et al., 2020) requires captioning a specific user-interface element, making it sensitive to localized UI structure and fine-grained visual details. SciCap (Hsu et al., 2021) evaluates scientific figure captioning, where the model must describe structured visual content such as plots, diagrams, and scientific imagery.

General visual question answering and reasoning. VQAv2 (Goyal et al., 2017) is a broad visual question answering benchmark over natural images and serves as a general-purpose VQA test. GQA (Hudson & Manning, 2019) evaluates compositional visual reasoning over scene graphs, stressing object relations and structured reasoning. xGQA (Pfeiffer et al., 2022) extends GQA to multilingual settings, testing whether visual reasoning transfers across languages. OKVQA (Marino et al., 2019) requires outside knowledge in addition to visual understanding. AOKVQA (Schwenk et al., 2022) further targets knowledge-intensive visual question answering and we evaluate both direct-answer (AOKVQA-DA) and multiple-choice (AOKVQA-MC) variants. AI2D (Kembhavi et al., 2016) evaluates diagram understanding on science-style illustrations, requiring the model to interpret arrows, labels, and spatial relations. ScienceQA (Lu et al., 2022) evaluates multimodal science question answering, combining visual interpretation with textual and commonsense reasoning. TallyQA (Acharya et al., 2019) targets counting-based visual question answering, with simple and complex splits that differ in the difficulty of counting and relational reasoning. CountBenchQA (Beyer et al., 2024) similarly stresses counting and quantity-sensitive reasoning, though with improved and corrected annotations over TallyQA as described in PaliGemma (Beyer et al., 2024). NLVR2 (Suhr et al., 2019) evaluates reasoning over paired images and a natural-language statement, requiring the model to jointly inspect multiple visual inputs. MARVL-5 (Liu et al., 2021) extends this style of multi-image reasoning to multilingual and culturally diverse settings. VizWizVQA (Gurari et al., 2018) evaluates VQA on images captured by blind or low-vision users, which often contain blur, occlusion, unusual framing, or low visual quality. Screen2Words (Wang et al., 2021) evaluates mobile-screen summarization, requiring the model to produce a concise natural-language description of an interface screen. OCR-VQA (Mishra et al., 2019) focuses on recognizing and reasoning over text in images. RSVQA (Lobry et al., 2020) evaluates visual question answering over remote-sensing imagery, with low-resolution and high-resolution subsets that stress geospatial interpretation at different image scales.

Captioning and multilingual image understanding. COCO-CAP (Lin et al., 2014; Chen et al., 2015) evaluates standard image captioning on MSCOCO-style natural images. NoCaps (Agrawal et al., 2019) evaluates captioning on images containing novel objects beyond the standard COCO object categories, testing open-vocabulary generalization. COCO-35L (Thapliyal et al., 2022) evaluates multilingual image captioning using COCO captions translated across multiple languages, including an English split and multilingual averages. XM3600 (Thapliyal et al., 2022) evaluates cross-lingual image captioning over a diverse multilingual captioning set, further testing whether compressed visual representations remain useful across language settings.

D.2 LLaVA 1.5 Benchmark Suite

The LLaVA 1.5 evaluation also uses GQA, the image subset of ScienceQA, and TextVQA, described above. MME-Perception (MME-P) (Fu et al., 2025) aggregates visual perception tasks. POPE (Li et al., 2023c) probes object hallucination through questions about whether an object is present in an image. SEED-Bench-IMG (Li et al., 2023a) evaluates image understanding through multiple-choice questions. OCRBench (Liu et al., 2024c) tests text recognition and text-based reasoning across visual inputs; we report its score out of 1000.

E Implementation Details

In this section, we describe the implementation details for the PaliGemma-2 (Appendix E.1) and LLaVA 1.5 (Appendix E.2) experiments.

E.1 Implementation and Evaluation Details for PaliGemma-2 Experiments

Budget sampling and routing. For PARCEL and all baselines, we follow the architectural choices and budget-sampling strategies of the corresponding elastic compression methods as closely as possible. For MQT, the visual token budget is entirely allocated to query tokens. During training, we sample an even query-token budget from $\{2, 4, \ldots, 256\}$ for the default 224×224 setting. For PARCEL, the sampled visual token budget is decomposed into spatial anchor tokens and query tokens according to the routing strategy in Section 3.3. At 224×224, the SigLIP visual grid contains $16 \times 16 = 256$ tokens. We use two spatial anchor resolutions: a 4×4 anchor grid with $N_p = 16$ tokens, obtained by 4×4 average pooling, and an 8×8 anchor grid with $N_p = 64$ tokens, obtained by 2×2 average pooling. We sample an even total budget $B \in \{16, 18, \ldots, 256\}$. If $16 \leq B < 64$, we use the $N_p = 16$ anchor grid and allocate $N_q = B - 16$ query tokens. If $64 \leq B \leq 256$, we use the $N_p = 64$ anchor grid and allocate $N_q = B - 64$ query tokens. Thus, query tokens are used only to fill the remaining budget not explicitly covered by the spatial anchor grid. For M^3, we follow the square-only spatial pooling strategy of the original method (Cai et al., 2025). This gives supported token budgets $\{4, 16, 64, 256\}$ in the 224×224 setting. In the main experiments, we report the budgets shared with the other methods, namely 16, 64, and 256 visual tokens.

Connector architecture. For MQT, we use a single query-to-visual cross-attention block following the official MQT design (Hu et al., 2024). For PARCEL, we use exactly one Query $\leftrightarrow$ Pool self-attention block followed by one Query $\rightarrow$ ViT cross-attention block. No connector attention block is repeated. All connector attention blocks operate at the visual hidden width $D_v = 1152$ and use 12 attention heads. The query tokens form a single ordered nested list of tokens shared across routing regimes, with each budget retaining its first N_q queries. For example, the first 46 query embeddings are shared between the low-budget regime that fills budgets above the 16-token anchor and the higher-budget regime that starts from the 64-token anchor and adds query tokens. This shared prefix structure is the mechanism that allows nested dropout to support elastic query truncation across budgets.

Pretraining setup. For PaliGemma-2, the base initialization follows the unimodal pretraining of its constituent modules, including contrastive vision-language pretraining for SigLIP (Zhai et al., 2023) and autoregressive text-only pretraining for Gemma 2 (Team et al., 2024). Starting from these unimodal components, we train the Vanilla PaliGemma-2 model for 100M samples using the Stage-1 pretraining recipe described in PaliGemma-2 (Steiner et al., 2024). We use 100M samples because the PaliGemma and PaliGemma-2 studies show that many transfer benchmarks begin to saturate at this scale (Beyer et al., 2024; Steiner et al., 2024), and because 1B-sample pretraining is computationally prohibitive for our comparison across multiple compression methods. For this Stage-1 pretraining, we follow the PaliGemma-2 configuration without modifying the learning-rate multipliers, data mixture, pretraining task mixture, or Gemma 2 logit soft-capping (Team et al., 2024; Bello et al., 2016). The pretraining mixture includes captioning, grounded captioning (Wan et al., 2024), OCR, VQA (Changpinyo et al., 2022), detection, and instance segmentation tasks (Chen et al., 2021; 2023b), with data drawn from sources such as WebLI (Chen et al., 2022; 2023a) and CC3M (Sharma et al., 2018). For full details on the pretraining data, task definitions, and splits, we refer the reader to the PaliGemma (Beyer et al., 2024) and PaliGemma-2 (Steiner et al., 2024) works.

Training infrastructure and optimizer. All models are trained in the open-source `big_vision` codebase (Beyer et al., 2022) following the PaliGemma training setup (Beyer et al., 2024; Steiner et al., 2024), but using Cloud TPUv4 accelerators (Google Cloud, 2026). During pretraining, data, model parameters, and optimizer state are sharded across devices using the JAX/GSPMD (Bradbury et al., 2018; Xu et al., 2021) fully-sharded data-parallel strategy adopted by PaliGemma. We use 256 TPUv4 chips, a global batch size of 8192 and no gradient accumulation for pretraining. Following PaliGemma-2 (Steiner et al., 2024), we use Adam (Kingma & Ba, 2014; Loshchilov & Hutter, 2017) with default hyperparameters. For Stage-1 and Stage-2 pretraining of the PaliGemma-2 3B backbone, we use the default PaliGemma learning rate of 2×10^{-5} multiplied by 0.5, following the PaliGemma-2 scaling rule. We also follow PaliGemma-2 in applying Gemma-2 logit soft-capping (Team et al., 2024; Bello et al., 2016) during Stage-1 and Stage-2 pretraining, but not during transfer tuning.

Intermediary pretraining of compressed models. After Stage-1 pretraining, we integrate the method-specific connector components for MQT, M^3, PARCEL, and the ablations. Weights shared with PaliGemma-2, including the vision encoder, language decoder, and cross-modal projection, are initialized from the Stage-1 model, while newly introduced method-specific connector parameters are randomly initialized. During intermediary pretraining, the SigLIP vision encoder, Gemma-2 language decoder, cross-modal projection, and connector parameters are all trainable, following the fully trainable PaliGemma-2 setup. The learning-rate schedule for newly introduced connector parameters follows the schedule used for the PaliGemma cross-modal projection. We then perform an additional 100M-sample intermediary pretraining stage for each compressed model with the full method-specific architecture active. During this stage, nested dropout is enabled for both MQT and PARCEL. Because the newly introduced connector parameters must be learned from scratch, we restart the learning-rate schedule for intermediary pretraining rather than resuming the Stage-1 schedule. For a fair comparison against the uncompressed reference, we also continue Vanilla PaliGemma-2 training for an additional 100M samples.

Transfer tuning and evaluation. After pretraining, we perform transfer tuning on the benchmarks described in Appendix D. Budget sampling remains active during transfer tuning for all elastic compression methods, using the same budget ranges and random sampling strategy as in pretraining. For video transfer tasks, visual token budgets are applied per frame, and frame sampling follows the PaliGemma/PaliGemma-2 evaluation setup (Beyer et al., 2024; Steiner et al., 2024). We do not perform additional hyperparameter tuning for the proposed method or baselines. For model selection, we follow the PaliGemma and PaliGemma-2 transfer protocols and rely on the corresponding evaluation or validation splits whenever available. The exact transfer-tuning hyperparameters are benchmark-dependent, so we follow the corresponding PaliGemma and PaliGemma-2 transfer recipes. All three-seed experiments use the same set of randomly sampled seeds across methods, sampled from the range $[10{,}000, 100{,}000]$.

High-resolution 448×448 setting. For the high-resolution experiments, we follow the Stage-2 high-resolution pretraining strategy of PaliGemma-2 (Steiner et al., 2024). This stage uses 448×448 image inputs and is run for 10M samples. At this resolution, the SigLIP visual grid contains $32 \times 32 = 1024$ visual tokens. For MQT, we expand the nested list of query tokens and sample even budgets from $\{2, 4, \ldots, 1024\}$. For M^3, square-only spatial pooling gives supported budgets $\{4, 16, 64, 256, 1024\}$.

For PARCEL, we extend the default-resolution routing by introducing a third spatial anchor scale. Specifically, we use anchor sizes $N_p \in \{16, 64, 256\}$, corresponding to 4×4, 8×8, and 16×16 anchor grids on the 32×32 source grid. Equivalently, these are obtained by 8×8, 4×4, and 2×2 average pooling, respectively. Given a sampled high-resolution budget B, we allocate query tokens to fill the gap above the selected anchor size:

$$(N_p, N_q) = \begin{cases} (16, B-16), & 16 \le B < 64, \\ (64, B-64), & 64 \le B < 256, \\ (256, B-256), & 256 \le B \le 1024. \end{cases} \tag{41}$$

After the 10M-sample high-resolution pretraining stage, we perform high-resolution transfer tuning and evaluation using the PaliGemma-2 high-resolution protocol. Due to the substantially higher

computational cost of 448×448 pretraining and evaluation, high-resolution benchmark results are reported from a single seed.

E.2 Implementation and Evaluation Details for LLaVA 1.5 Experiments

Backbone and connector architecture. We follow the MQT (Hu et al., 2024) and M^3 (Cai et al., 2025) codebases using LLaVA 1.5-7B (Liu et al., 2024a). [2] The model combines a CLIP ViT-L/14 vision encoder at 336×336 resolution with a Vicuna-7B v1.5 language decoder (Chiang et al., 2023). We use patch features from the penultimate vision-encoder layer, yielding a 24×24 grid of 576 tokens with hidden dimension $D_v = 1024$ and use a total number of 256 learnable queries following our PaliGemma-2 experiments and Hu et al. (2024). The vision encoder remains frozen throughout both training stages, following (Hu et al., 2024; Liu et al., 2024a). For PARCEL, Pool-Conditioned Query Resampling operates at $D_v = 1024$ with 16 attention heads, using one Query $\leftrightarrow$ Pool self-attention block followed by one Query $\rightarrow$ ViT cross-attention block. The original spatial anchors and the resampled query features are concatenated, normalized, and projected to the 4096-dimensional language-decoder space. MQT uses its single-layer query resampler (Hu et al., 2024). For M^3, we retain the two-layer GELU projector and apply spatial average pooling after projection, following the original architecture (Cai et al., 2025).

Budget sampling and routing. At the default resolution, PARCEL uses 6×6 and 12×12 spatial anchor grids, obtained through 4×4 and 2×2 average pooling, respectively. The general routing rule in Section 3.3 therefore specializes to:

$$(N_p, N_q) = \begin{cases} (36, B - 36), & 36 \le B < 144, \\ (144, B - 144), & 144 \le B \le 256. \end{cases} \tag{42}$$

During elastic instruction tuning, PARCEL samples even total budgets from $\{36, 38, \ldots, 256\}$. MQT samples even query budgets from $\{2, 4, \ldots, 256\}$, while M^3 samples from its original scale set $\{1, 9, 36, 144, 576\}$ (Cai et al., 2025; Hu et al., 2024). We sample one budget per minibatch for each method, using the one-forward-pass variant of M^3. We evaluate PARCEL and MQT at 36, 64, 144, and 256 tokens. M^3 is evaluated at the two overlapping budgets with MQT and PARCEL in its training scale set, 36 and 144 tokens. For PARCEL, both of these budgets are anchor-only with no query tokens used at inference.

Two-stage training. Each method undergoes its own connector-alignment stage followed by elastic instruction tuning, following Hu et al. (2024); Cai et al. (2025). During alignment, we train only the connector on the 558K image-text pairs of the LLaVA pretraining mixture (Liu et al., 2023; 2024a) for one epoch, keeping both the vision encoder and language decoder frozen. Alignment uses a fixed budget of 256 tokens for PARCEL and MQT, and the full 576-token grid for M^3. We then initialize each method from its corresponding alignment checkpoint and perform instruction tuning on the LLaVA 1.5 665K mixture (Liu et al., 2023; 2024a). Following the MQT training recipe (Hu et al., 2024), we use two instruction-tuning epochs for all three methods. Optimizer, learning rate, and batch size settings directly follow MQT (Hu et al., 2024), based on LLaVA 1.5 (Liu et al., 2024a). The global batch sizes are 256 during alignment and 128 during instruction tuning for every method, following Liu et al. (2023); Hu et al. (2024); Cai et al. (2025). Both stages are run on eight NVIDIA A100 GPUs with `bfloat16` training and DeepSpeed (Rasley et al., 2020) ZeRO-2 and ZeRO-3 (Rajbhandari et al., 2020) for alignment and instruction tuning stages, respectively. Instruction tuning uses the Vicuna conversation template, padded image aspect ratios, and a maximum sequence length of 2048 tokens, following (Chiang et al., 2023).

Evaluation. We evaluate all methods using the same LLaVA evaluation pipeline and benchmark splits as Hu et al. (2024); Cai et al. (2025). We report GQA test-dev balanced accuracy (Hudson & Manning, 2019), MME-Perception scores (Fu et al., 2025), mean POPE F1 across all splits (Li et al., 2023c), TextVQA validation accuracy (Singh et al., 2019), and results on the image subsets of ScienceQA (Lu et al., 2022) and SEED-Bench (Li et al., 2023a). OCRBench (Liu et al., 2024c) is reported on its original 1000-point scale. Retention is computed relative to our uncompressed LLaVA 1.5 reference with 576 visual tokens. The reported comparison uses a single random seed shared across methods, with the training seed also controlling budget sampling.

[2] The baseline implementations follow the official `MQT-LLaVA` and `matryoshka-mm` repositories.

F Further Discussion of Related Work

In this section, we provide further discussion on related work, namely on which part of the pipeline to compress visual tokens (Appendix F.1) and a different family of efficient inference methods focusing on compressing visual tokens through visual bottlenecks and register tokens (Appendix F.2).

F.1 Visual Token Compression at Different Stages of LVLM Pipeline

Many LVLMs connect a visual encoder to a language decoder through a projection or learned resampling module, such as a linear projection or a Q-Former (Liu et al., 2024a; Steiner et al., 2024; Li et al., 2023b; Bai et al., 2023; Chen et al., 2025; Li et al., 2022; An et al., 2025; Beyer et al., 2024; Zhu et al., 2025; Chen et al., 2024d). For models that pass visual tokens into the decoder sequence, reducing their number at the bridge layer between the visual encoder and the language decoder also reduces the tokens processed by every subsequent heavyweight LLM layer. Many other approaches select tokens after they enter the decoder: FastV (Chen et al., 2024b) uses early-layer attention to prune visual tokens, PyramidDrop (Xing et al., 2024) progressively removes tokens across layers, and FEATHER (Endo et al., 2025) revisits the selection criteria and timing of pruning to improve performance on spatial tasks such as visual grounding. Nüwa combines aggregation after the visual encoder with a later, text-guided pruning stage inside the LLM (Huang et al., 2026). These methods can use information that emerges when visual and text tokens interact, but their pruning decisions depend on intermediate decoder states and the layers at which tokens are removed. PARCEL instead forms its spatial anchors and pool-conditioned queries before the visual tokens enter the LLM decoder. The requested budget determines the tokens passed to the decoder, without modifying its layers or relying on their attention states. Keeping compression at the connector thus avoids the layer-specific dependencies of decoder-side pruning when adapting the method to different LVLMs, which improves the practicality of integrating PARCEL for a broader range of applications.

Finally, a growing body of work brings visual and language processing closer together, either within a shared transformer (Bavishi et al., 2023; Diao et al., 2024; 2025; Luo et al., 2024; Chen et al., 2024c; Diao et al., 2026) or by using pretrained blocks from a particular modality to improve representations of the other (Pang et al., 2023; Lai et al., 2024; Bai et al., 2025; Kuzucu et al., 2026). EVE removes the pretrained visual encoder, EVEv2 retains some modality-specific operations within its shared architecture, and Mono-InternVL incorporates visual experts into a language model (Diao et al., 2024; 2025; Luo et al., 2024). Our experiments study LVLMs with a distinct visual encoder and an LLM decoder along with a bridging mechanism in between. Following from the efficacy of PARCEL on a wide range of applications in Section 4, we believe that applying PARCEL to monolithic VLMs poses an interesting future work to explore.

F.2 Visual Token Compression Through Visual Bottlenecks and Register Tokens

VoCo-LLaMA (Ye et al., 2025b) uses attention distillation to train vision-compression tokens within the language model, while Adaptive-VoCo (Guo et al., 2026) predicts their number from input complexity. Fwd2Bot (Bulat et al., 2025) forms a visual bottleneck in one LLM forward pass and uses it in a second pass with the instruction. HTC-VLM (Zhang et al., 2025) combines continuous visual features with discrete semantic anchors before compressing them into a compact bottleneck. By explicitly dividing the labor of feature extraction, PARCEL retains grid-aligned pool tokens as spatial anchors and conditions elastic queries on them to recover detail beyond the pooled grid. This coupling improves both mixed pool–query budgets and anchor-only budgets, where no query tokens are used at inference.

G Limitations

While PARCEL improves the accuracy–efficiency trade-off for visual token compression, several limitations remain. First, our method inherits the limitations of the underlying PaliGemma-2 and LLaVA 1.5 backbones and their pretraining data. As with other large vision-language models, the models may reflect biases present in web-scale image-text data, including demographic, cultural, geographic, and linguistic data imbalance. Second, although token compression reduces inference

cost, training and evaluating large multimodal models still requires substantial compute. This limits accessibility and motivates future work on more compute-efficient training recipes, lightweight ablations, and better low-cost evaluation protocols.

Our method also uses a list of budgets that is specified by the practitioner rather than predicted from the input. Accordingly, developing an input-adaptive budget predictor that allocates visual tokens dynamically is therefore an important direction for future work.