FROM VISUAL TOKEN REDUCTION TO REALIZED EDGE EFFICIENCY: A STAGE-WISE STUDY WITH CIVIC

Anonymous authors
Paper under double-blind review

ABSTRACT

Reducing visual tokens is a common strategy for accelerating vision–language models, yet token counts and operation counts do not establish faster batch-one inference on edge hardware. Compression after visual encoding leaves expensive backbone layers untouched, while early compression may be undone at the multimodal interface or offset by aggregation overhead. This paper presents **C**ompact **I**nference for **V**ision-Language **I**ntegrated **C**ompression (CIVIC), which aggregates visual states at an intermediate encoder layer and carries the compact representation through the remaining visual blocks, projector, and language-model prefill. A configurable visual budget controls aggregation, and text-aligned distillation transfers predictions across unequal visual-token sequences. The evaluation adapts CIVIC to Qwen3-VL-4B-Instruct and profiles batch-one inference on Jetson AGX Orin, tracing interface lengths and timing aggregation, the encoder, the projector, and prefill separately. The median final prefix falls from 816 to 470 tokens; derived response-length-standardized latency decreases by 26.6%, with lower measured encoder and prefill stage times. In a fixed-configuration control, restoring dense downstream tokens increases derived latency, whereas optional visual key–value pooling adds device cost. CIVIC reduces derived latency on the four scored benchmarks and within each Video-MME duration group. Task scores show benchmark-dependent quality trade-offs. Together, the stage traces and fixed-budget controls connect persistent compactness with realized edge acceleration despite the measured aggregation overhead.

1 INTRODUCTION

Vision–language models (VLMs) combine visual perception with language generation in systems such as Flamingo (Alayrac et al., 2022), BLIP-2 (Li et al., 2023), and LLaVA (Liu et al., 2023). Deploying this capability on robots, vehicles, and other edge platforms makes latency and memory use as important as the ability to answer a visual question. At batch size one, each request traverses visual encoding, multimodal projection, language-model prefill, and autoregressive generation under a limited device budget. Higher image resolutions and longer clips increase the visual sequence presented to these stages. Reducing that sequence can relieve computation and memory pressure, provided that the savings persist to the stages that dominate device execution. On memory-limited devices, unusually long intermediate token spans can also exceed the capacity of individual layers. The deployment challenge is therefore to shorten the actual sequence processed by costly layers while preserving the visual evidence needed for the task.

The timing of compression within the model is therefore consequential. Methods that prune or merge visual tokens within an encoder can leave earlier layers operating on the original sequence. Compressing only after visual encoding can shorten the language-model input but cannot recover the work already spent on the visual backbone. Conversely, an early reduction may cease to matter if a projector, spatial merger, or input-construction step reinstates a longer visual span. Selection, assignment, and pooling also introduce work of their own. Attention kernels (Dao et al., 2022) and decoder cache management (Kwon et al., 2023) further affect the cost of a given sequence length. Token counts and FLOP counts therefore describe potential savings; on-device stage times determine whether a compact pathway realizes them. The encoder's patch states, projected groups, and language placeholders have different structural requirements. Preserving their native interfaces while reducing their lengths is essential to obtaining downstream savings.

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.

Prior work offers effective reductions at different interfaces: DynamicViT (Rao et al., 2021), ToMe (Bolya et al., 2023), and DiffRate (Chen et al., 2023) modify vision-transformer sequences; TokenPacker (Li et al., 2025b) compresses the projector output; and LLaVA-Mini (Zhang et al., 2025), Dynamic-LLaVA (Huang et al., 2025), and iLLaVA (Hu et al., 2026) study compact multimodal or generation paths. FastVLM (Vasu et al., 2025) explicitly improves visual encoding, and EarlyTom (Wang et al., 2026) compresses video tokens early. Prior studies establish that visual-token reduction and device latency both matter. For a compact pathway to deliver edge acceleration, the reduction must survive the encoder, projector, and language-model interface, and the cost saved at those stages must exceed the work introduced by compression. Resolving that relationship requires interface lengths, device-stage timings, and task quality in the same deployment setting.

Compact **I**nference for **V**ision-Language **I**ntegrated **C**ompression (CIVIC) addresses this problem through learned aggregation after a dense visual prefix and before the remaining visual blocks. The compact tokens pass through the projector and language-model input construction without reconstructing the dense visual span. Configurable retention controls the aggregation budget, and text-aligned distillation transfers teacher behavior without imposing a one-to-one visual-token mapping. The resulting notion of *persistent compactness* can be tested at the encoder output, projected sequence, and prefill context. Adapting CIVIC to Qwen3-VL (Bai et al., 2025) additionally requires consistent positions, spatial-merger groups, language placeholders, and intermediate visual features. These interface requirements provide a concrete basis for adapting the compact pathway to future backbone designs.

The study traces encoder, projector, and prefill lengths and times during batch-one inference on an NVIDIA Jetson AGX Orin. A response-length-standardized latency statistic measures net device behavior; fixed-budget restoration and attention controls locate the savings within the path. Together, these measurements test whether structural compression survives the backbone's interfaces and becomes realized acceleration. Figure 1 previews the relationship between encoder work, retained sequence lengths, and device latency.

The contributions are threefold:

- A stage-wise framework traces compactness across the visual encoder, projector, and language-model prefill while measuring batch-one device cost.
- CIVIC combines intermediate-layer aggregation, configurable retention, and text-aligned distillation with an explicit contract for backbone interfaces.
- Batch-one stage timing, task-level latency, and fixed-budget controls show how persistent compression translates into measured edge savings.

2 Related Work

2.1 Visual-Token Compression and Compact VLM Interfaces

Visual token selection and aggregation. Patch-based vision transformers (Dosovitskiy et al., 2021) offer a natural granularity for compression. Perceiver (Jaegle et al., 2021) and TokenLearner (Ryoo et al., 2021) form learned compact representations. DynamicViT (Rao et al., 2021), EViT (Liang et al., 2022), ToMe (Bolya et al., 2023), and DiffRate (Chen et al., 2023) change token populations or retention rates within vision backbones. Recent VLM methods move the selection point or incorporate multimodal cues: FastV (Chen et al., 2024) and PyramidDrop (Xing et al., 2025) prune inside the language model, while VisionZip (Yang et al., 2025), DyMU (Wang et al., 2025b), and DivPrune (Alvar et al., 2025) preserve visual information under compression. VisionTrim (Yu et al., 2026) studies both vision and language-stage efficiency; EarlyTom (Wang et al., 2026) moves video-token compression early. CIVIC maintains its reduced sequence through the remaining vision blocks and multimodal interface and measures each stage on edge hardware.

Multimodal bottlenecks and language context. Flamingo (Alayrac et al., 2022) and BLIP-2 (Li et al., 2023) illustrate different couplings of pretrained visual and language components; BLIP-2 uses a querying bridge. TokenPacker (Li et al., 2025b) compresses the visual projector. DeCo (Yao et al., 2024) uses projector-level patch compression, while MQT-LLaVA (Hu et al., 2024) learns an elastic set of visual queries. LLaVA-Mini (Zhang et al., 2025) combines modality pre-fusion with a short

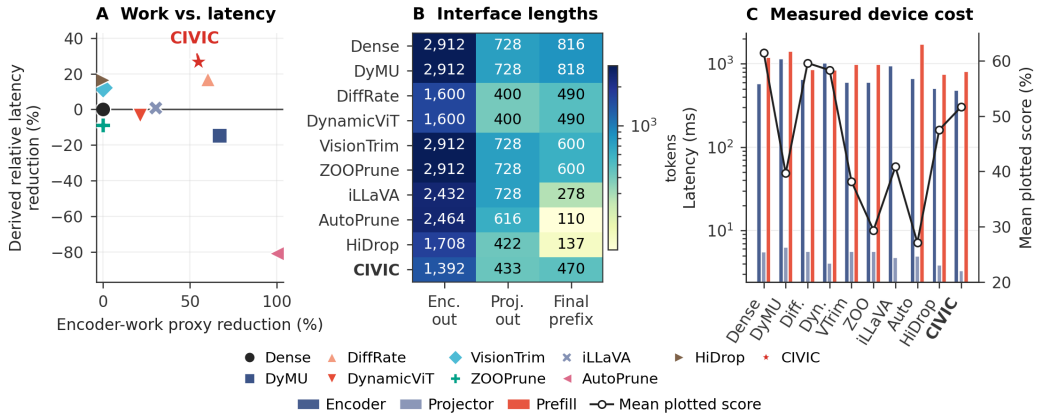

Figure 1: Nominal work and realized device behavior. (A) Encoder-work proxy versus derived standardized-latency reduction; the horizontal line marks no latency change. (B) Marginal median encoder-output, projector-output, and final-prefix lengths; DiffRate and DynamicViT projector lengths use the Dense merger ratio. (C) Median encoder, projector, and prefill times over seven device settings (left, log scale), alongside the mean plotted score over up to four scored benchmarks (right, percent); coverage differs by method. Stage bars are absent when no measurement is displayed.

visual prefix and measures response latency. Dynamic-LLaVA (Huang et al., 2025) reduces visual and generated-language contexts through prefill and decoding; iLLaVA (Hu et al., 2026) analyzes encoder and decoder costs jointly. Inference-optimal VLMs (Li et al., 2025a) examine compute allocation across model size and visual tokens. FastVLM (Vasu et al., 2025) co-designs the visual encoder for latency; MobileVLM (Chu et al., 2023) develops a mobile-oriented VLM. Together these studies motivate compact interfaces and device evaluation. CIVIC addresses a complementary setting: it adapts an existing Qwen3-VL backbone while retaining its pretrained visual tower, carries compact states through its native multimodal interfaces, and measures the encoder, projector, and prefill separately under batch-one edge inference. This distinction links the location of compression to the stages at which device cost is actually removed.

Memory and realized device efficiency. Reducing the final prefix also changes the decoder's cache footprint. H2O (Zhang et al., 2023), SnapKV (Li et al., 2024), and VL-Cache (Tu et al., 2025) select cache states at different points in inference; Compress & Cache (Bulat et al., 2025) amortizes visual computation over reuse. Kernel design matters independently of token count: FlashAttention (Dao et al., 2022) reduces attention memory traffic, and PagedAttention (Kwon et al., 2023) changes cache allocation and serving efficiency. These mechanisms motivate reporting both interface lengths and on-device stage times. Table 1 maps the intervention points of the evaluated methods.

3 Methodology

3.1 Stage-Wise Study Formulation

For a given input, let n_e and n_p denote observed encoder-output and projected visual lengths. Dense and CIVIC merge all their projected visual states into placeholders, so their prefix and initialized decoder cache have length $L + n_p$, for L nonvisual prefix tokens. Methods that prune or alter the sequence after projection have a separately observed final-prefix length. Below, Dense uses $n_e = T_e$, $n_p = T_p$ and CIVIC uses $n_e = M_e$, $n_p = M_p$; the latter is not the nominal local aggregation budget. Temporary vision-attention KV anchors are distinct from the decoder cache. At each measured boundary, reduced sequence length must translate into device-stage savings after aggregation overhead.

Table 1: Compression points for the evaluated implementations. Dense is the uncompressed reference; CIVIC (Ours) is the proposed pathway. Descriptions identify intervention locations rather than implying comparable rankings.

Method	Compression point	Evaluation focus
Dense	None	Uncompressed device reference
DynamicViT	Vision encoder	Dynamic token selection
DiffRate	Vision encoder	Differentiable retention rate
DyMU	Visual tokens	Multimodal-guided token merging
VisionTrim	Vision and language interfaces	Vision-side and downstream compression
ZOOPrune	Visual tokens	Visual-token pruning
iLLaVA	Encoder and decoder	Joint visual and generation cost
AutoPrune	Visual tokens	Input-dependent pruning policy
HiDrop	Visual and language interfaces	Late injection and hierarchical pruning
CIVIC (Ours)	Intermediate vision layer to prefill	Persistent compactness and measured edge-stage cost

3.2 Persistent Compactness

Let $\mathbf{X}_0 \in \mathbb{R}^{T_e \times d_v}$ be the patch embeddings for a visual input, and let $\mathbf{E}_x \in \mathbb{R}^{L \times d_\ell}$ be the text embeddings. A dense VLM maps the visual sequence through encoder E_v, projector Π, and multimodal merge $\mathcal{M}$:

$$\mathbf{H}_d = E_v(\mathbf{X}_0), \qquad \mathbf{V}_d = \Pi(\mathbf{H}_d), \qquad \mathbf{U}_d = \mathcal{M}(\mathbf{E}_x, \mathbf{V}_d). \tag{1}$$

For a fixed input i, the encoder length is $T_{e,i}$, projected visual length is $T_{p,i}$, and the language-model prefix has length $L_i + T_{p,i}$, where L_i counts the nonvisual tokens actually passed to the model. Holding this input, prompt, and tokenizer fixed is required when attributing a change in prefix length to visual compression.

CIVIC processes the first l_0 visual blocks at the original length, aggregates their states, and keeps the resulting sequence compact through the remaining $N_v - l_0$ blocks and downstream interfaces:

$$\begin{aligned} &\mathbf{H}_0 = E_v^{1:l_0}(\mathbf{X}_0), && \mathbf{Z}_0 = \mathcal{A}_\phi(\mathbf{H}_0), \\ &\mathbf{H}_c = E_v^{l_0+1:N_v}(\mathbf{Z}_0, \mathbf{P}_c), && \mathbf{V}_c = \Pi_c(\mathbf{H}_c), \\ &\mathbf{U}_c = \mathcal{M}(\mathbf{E}_x, \mathbf{V}_c). \end{aligned} \tag{2}$$

The evaluated Qwen implementation has $l_0 = 16$; its initial visual blocks therefore remain dense. Compaction is persistent from the aggregation boundary onward. The nominal budget applies at an aggregation invocation, whereas the reported encoder-output lengths summarize the complete input. Its corresponding lengths on the same input are M_e, M_p, and $L + M_p$. An input is *persistently compact* when its observed lengths remain strictly smaller at each listed interface:

$$\boldsymbol{n}_c = [M_e \quad M_p \quad L + M_p]^\top \prec [T_e \quad T_p \quad L + T_p]^\top = \boldsymbol{n}_d. \tag{3}$$

The ordering is element-wise; it becomes non-strict when Eq. 4 retains every patch on a short input. CIVIC propagates the selected compact tokens without restoring a dense downstream visual span. Device timing establishes whether this structural saving exceeds aggregation and kernel costs.

3.3 CIVIC Overview

Write ϕ for aggregation and adapter parameters. The Qwen general-task student is first supervised-fine-tuned with its vision backbone frozen and a selected set of language and multimodal weights trainable; a subsequent stage distills from a dense teacher. Figure 2 sketches the compact inference path; the training phases are described in Appendix A.

3.4 Compact Visual Aggregation

A fixed budget can be too aggressive when resolution or video duration grows. For each aggregation invocation on t incoming visual states, CIVIC selects a local compact length m using a nominal budget C_e and optional minimum retention rate $\rho_{\min}$:

$$m = \min\left\{t,\ \max\left\{C_e, \lceil \rho_{\min} t \rceil\right\}\right\}. \tag{4}$$

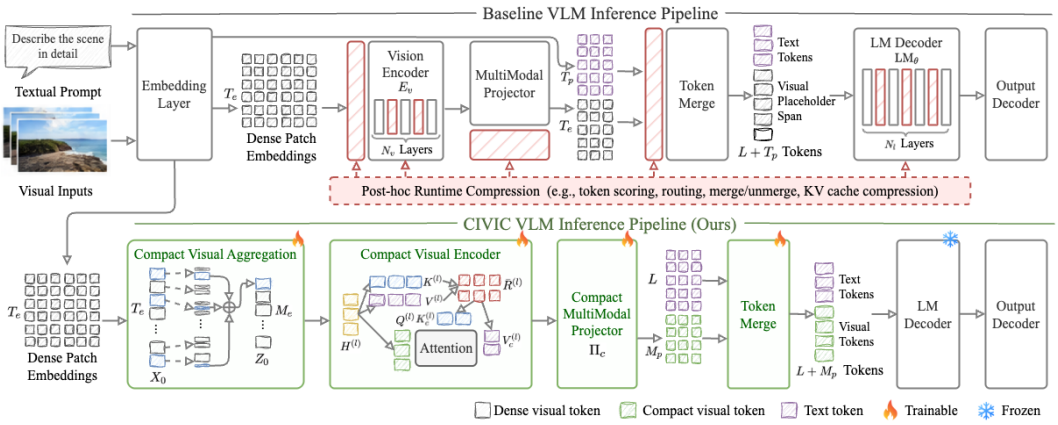

Figure 2: Conceptual dense (upper) and CIVIC (lower) inference paths. Compact states traverse the spatial merger and projector and enter language-model prefill without restoring the dense visual span. The drawn visual-KV circuit is an optional component control, disabled in the headline configuration. The training icons are illustrative, not the trainable-parameter map of the evaluated Qwen model; see Appendix A.

Here $\mathbf{H}_0 \in \mathbb{R}^{t \times d_v}$ denotes the segment of dense visual states entering one invocation. Across visual units the measured interface lengths refer to the whole request, and can exceed the local budget. The headline Qwen setting uses $C_e = 256$ and $\rho_{\min} = 0$; the compact-128 controls use a separate setting. Let $\mathbf{A} \in \mathbb{R}^{m \times d_v}$ be learned anchors and $\mathbf{P}_0 \in \mathbb{R}^{t \times q}$ the dense spatial coordinates. Here $q = 2$ for images and $q = 3$ for videos, $\tau_A > 0$ is a temperature, LN is layer normalization, and norm denotes row-wise ℓ_2 normalization. CIVIC computes assignments over the dense-token dimension:

$$\overline{\mathbf{A}} = \mathrm{norm}(\mathrm{LN}(\mathbf{A}))\,, \qquad \overline{\mathbf{H}}_0 = \mathrm{norm}(\mathrm{LN}(\mathbf{H}_0))\,,$$
$$\mathbf{W}_A = \mathrm{softmax}_t\left(\overline{\mathbf{A}}\overline{\mathbf{H}}_0^{\top}/\tau_A\right), \qquad \mathbf{Z}_0 = \mathbf{W}_A\mathbf{H}_0, \qquad \mathbf{P}_c = \mathbf{W}_A\mathbf{P}_0. \tag{5}$$

Each row of $\mathbf{W}_A$ sums to one, so every compact token and coordinate is a weighted summary of the same dense grid. The assignment itself requires mt similarities and data movement. A weighted coordinate need not be an integer location on a regular image grid; it cannot be substituted into a grid-indexed positional pathway without specifying its interpolation rule.

3.5 Compact Encoding and Interface Propagation

Only the visual blocks following aggregation operate on compact states. Optional visual-key–value pooling shortens the post-aggregation attention-score matrix from $M_e \times M_e$ to $M_e \times S$, at the cost of assignments and pooling. Headline Qwen runs disable this operator; Appendix A specifies it for the separate compact-128 component control. Neither form re-expands the downstream visual sequence.

The intended merger/interface acts on $\mathbf{H}_c$ and yields $\mathbf{V}_c \in \mathbb{R}^{M_p \times d_\ell}$ without restoring dense visual placeholders. Backbone portability depends on an explicit *interface adapter*: for a backbone B, define $\mathcal{G}_B(\mathbf{H}_c, \mathbf{P}_c)$ to return valid native-grid, position, merger-group, placeholder, and intermediate-feature tensors as well as $\mathbf{V}_c$. Qwen3-VL couples grid-dependent rotary positions and a spatial merger to image/video token layout, and supplies DeepStack visual features to early language layers. Consequently changing the patch tensor alone cannot establish a working Qwen adaptation. The backbone-specific interface contract keeps projected visual counts, inserted placeholder counts, post-merger group sizes, and DeepStack feature indices mutually consistent without expanding the language prefix. Appendix A states the corresponding interface checks. Porting to a newer backbone requires implementing $\mathcal{G}_B$ for its native interfaces and remeasuring the complete inference path.

Ignoring non-attention work, let j index separately attended visual units with lengths t_j before and m_j after aggregation; a jointly attended request is a single unit. Under the attention-score model,

ratios for the split encoder, language prefill, and initialized cache are

$$\frac{\mathcal{C}^c_{\text{attn}}}{\mathcal{C}^d_{\text{attn}}} \approx \frac{\sum_j \left[l_0 t_j^2 + (N_v - l_0) m_j^2\right]}{N_v \sum_j t_j^2}, \quad \frac{\mathcal{C}^c_{\text{prefill,attn}}}{\mathcal{C}^d_{\text{prefill,attn}}} \approx \frac{(L + M_p)^2}{(L + T_p)^2}, \qquad \frac{\mathcal{M}^c_{\text{cache}}}{\mathcal{M}^d_{\text{cache}}} \approx \frac{L + M_p}{L + T_p}. \tag{6}$$

If optional compact visual KV is active, replace m_j^2 in the first numerator by $m_j S_j$ for affected blocks. These expressions describe attention and cache scaling only. Figure 1A reports a separate shape-based encoder-work proxy; its plotted value must not be read as work removed from the first l_0 blocks. Whole-encoder stage time independently captures the net effect of aggregation and device execution.

3.6 Text-Aligned Distillation

Dense and compact visual positions have no one-to-one correspondence, whereas text order is preserved. Distillation therefore uses aligned nonvisual text positions. Let $\boldsymbol{z}_t^i$ and $\boldsymbol{z}_s^{\pi(i)}$ be teacher and student logits, with π mapping the same text token across the two prefixes. Let $\mathcal{I}_t$ be the aligned text-position set, $T_D > 0$ the distillation temperature, and $\lambda \geq 0$ its weight. The objective is

$$\mathcal{L}_{\text{KD}} = \frac{\lambda T_D^2}{|\mathcal{I}_t|} \sum_{i \in \mathcal{I}_t} \text{KL}\Big[\text{softmax}(\boldsymbol{z}_t^i / T_D) \parallel \text{softmax}(\boldsymbol{z}_s^{\pi(i)} / T_D)\Big], \qquad \omega^* = \arg\min_{\omega} \mathcal{L}_{\text{KD}}. \tag{7}$$

Following knowledge distillation (Hinton et al., 2015), the objective aligns predictions at shared text positions while allowing different visual-token counts. Here ω denotes the trainable student parameter subset in the distillation stage, distinct from the broader supervised fine-tuning stage; the dense teacher is held fixed.

4 Experimental Setup

Platform and Protocol. The evaluation adapts Qwen3-VL-4B-Instruct (Bai et al., 2025) for batch-one inference on an NVIDIA Jetson AGX Orin 64GB. The recorded protocol uses deterministic decoding and synchronized device timing, with the checkpoint, prompts, precision, and generation API held fixed within each condition. For CIVIC, dense-teacher fine-tuning precedes compact-student fine-tuning and benchmark-specific distillation; the supervised compact student is selected by held-out validation loss. The evaluated Qwen variant compresses after visual layer 16, with a nominal local budget of 128 and no visual KV pooling. The prespecified evaluation inputs are used without method-specific sample selection; out-of-memory attempts reduce the number of finite device timings, whose coverage is reported in Appendix Table 5. Some high-token dense runs encounter out-of-memory execution during intermediate layers; affected conditions are identified in Figure 3. Aggregate timing and task quality are interpreted as separate measurements.

Comparisons and Tasks. The comparison includes DyMU (Wang et al., 2025b), DiffRate (Chen et al., 2023), DynamicViT (Rao et al., 2021), VisionTrim (Yu et al., 2026), ZOOPrune (Kim et al., 2026), iLLaVA (Hu et al., 2026), AutoPrune (Wang et al., 2025a), and HiDrop (Wu et al., 2026) in the recorded benchmark comparisons. CIVIC uses task-specific trained checkpoints, whereas the learned DiffRate, DynamicViT, and HiDrop comparison checkpoints pool their supported general training tasks. Their timing comparisons therefore describe the evaluated implementations rather than a uniform training-data budget. The evaluation covers MMMU (Yue et al., 2024), MathVision (Wang et al., 2024), RealWorldQA (xAI, 2024), and Video-MME (Fu et al., 2025); short, medium, and long video settings are distinguished when timings are available. The cross-task device summary also includes ODinW-13 and separate Video-MME duration groups, whereas scored comparisons use the four benchmarks in Table 2. Figure 3 displays task-level latency alongside task scores on those benchmarks; the fixed-budget component study covers five settings. Incomplete device runs are distinguished from scorable model outputs.

Metrics. The evaluation records encoder, projector, and final-prefix lengths; stage and end-to-end time; decode time per generated token; and peak CUDA allocation. An encoder-work proxy is reported alongside measured device times. To account for variation in response length, latency is

standardized at a Dense-reference output length K_b for benchmark b:

$$T_{\mathrm{std},i,m}(K_b) = T_{i,m}^{\mathrm{nondec}} + K_b \frac{T_{i,m}^{\mathrm{decode}}}{Y_{i,m}}, \tag{8}$$

where $Y_{i,m} > 0$ is the observed generated length. This applies the observed average decode rate to a common length; because decode costs can vary with cache length, the result is termed *derived standardized latency* throughout. Timing coverage and the fixed-budget native-score contrasts are reported separately.

Statistics and Ablations. Paired bootstrap (Efron, 1979) intervals summarize fixed-configuration component timing contrasts (Appendix D). The main comparison uses benchmark-specific checkpoints with a 128-token nominal local budget. Each control is interpreted relative to its own reference; component effects are reported for the fixed configuration.

5 Results

5.1 Nominal Reduction Does Not Guarantee Realized Efficiency

Figure 1A and Appendix Table 5 contrast the encoder-work proxy with device behavior. A larger work-proxy reduction does not necessarily yield a larger reduction in derived latency: DyMU has a 67.0% proxy but -14.6% relative latency, while DiffRate has a 60.2% proxy and $+16.9\%$ relative latency. AutoPrune illustrates an even wider gap between structural reduction and device execution. CIVIC combines a substantial work-proxy reduction with the largest displayed derived-latency reduction on available executions. Method coverage differs, so this ordering is descriptive rather than a matched-input cross-method ranking; Appendix Table 5 gives the measurements and coverage.

For CIVIC, the plotted comparison reports a 54.8% encoder-work proxy reduction and a 26.6% *derived* standardized-latency reduction. The plotted medians are encoder/projector/prefix counts of 2,912/728/816 for Dense and 1,392/433/470 for CIVIC. The reduction appears at the encoder, projector, and final prefix, consistent with the proposed persistent path after layer 16. Whole-encoder measurements in Table 4 establish the device-stage effect. Figure 1C places stage costs beside the mean plotted task score across available settings.

Comparisons across the remaining methods clarify why a short final prefix alone is insufficient. iLLaVA records a shorter final prefix than CIVIC but retains more tokens at the encoder output and has a higher cross-task median derived latency. DiffRate produces fewer projector tokens than CIVIC yet has a higher device-latency median. Conversely, VisionTrim and ZOOPrune shorten their final prefixes without reducing the recorded encoder-output length. These observations motivate measuring the whole path: a single boundary count cannot locate saved work or account for the cost of compression.

Stage medians in Appendix Table 4 help locate the gain. Relative to the dense model, CIVIC has a shorter recorded visual-encoder time (479.3 versus 573.7 ms) and language-model prefill time (810.1 versus 1187.8 ms), while its aggregation adds 23.7 ms of overhead. The prefill difference is particularly relevant for batch-one inference because the entire multimodal prefix must be consumed before generation begins. Projector time also falls, although it is small compared with visual encoding and prefill. The result shows that the shorter visual sequence reaches two expensive stages despite the additional aggregation operator. Table 5 separately reports the response-length-standardized latency statistic.

5.2 Task Latency and Quality

The per-task timing coordinates in Figure 3 show that CIVIC is faster than Dense in each displayed setting. Derived latency falls from 2.212 to 1.362 s on MMMU and from 33.831 to 19.186 s on MathVision; it also falls on RealWorldQA and in the plotted Video-MME short-duration setting. Among the plotted alternatives, CIVIC has the lowest task latency on MathVision and RealWorldQA, whereas other methods attain lower latency on MMMU and Video-MME short. Table 2 reports an equal-duration average for Video-MME: Dense latency is 2.321 s and CIVIC latency is 2.060 s. Its task scores are measured over the full benchmark; the duration-specific device times are provided in

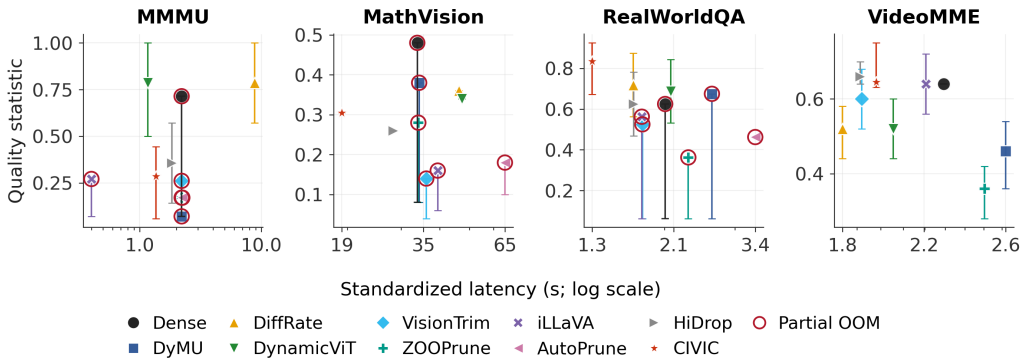

Figure 3: Task score versus derived standardized latency for MMMU, MathVision, RealWorldQA, and Video-MME. The Video-MME panel uses short-duration device latency; the Dense and CIVIC scores cover the full benchmark, and Table 2 summarizes all three durations. Vertical bars show the displayed score ranges. Red rings mark partial out-of-memory execution when high-token intermediate layers exceed Jetson memory capacity.

Table 2: Derived edge latency and task scores across image and video benchmarks. Bold marks lower latency or higher score.

Task	Derived latency (s)		Task score (0–1)	
	Dense	CIVIC (Ours)	Dense	CIVIC (Ours)
MMMU	2.212	**1.362**	**0.714**	0.285
MATH-Vision	33.831	**19.186**	**0.480**	0.305
RealWorldQA	2.007	**1.317**	0.625	**0.835**
Video-MME	2.321	**2.060**	0.640	**0.645**

Appendix Table 3. Encoder and prefill measurements in Appendix Table 4 locate the principal stage savings behind these task times.

CIVIC scores 0.285 on MMMU, 0.305 on MathVision, 0.835 on RealWorldQA, and 0.645 on Video-MME, compared with Dense scores of 0.714, 0.480, 0.625, and 0.640. Quality is lower on MMMU and MathVision, higher on RealWorldQA, and similar on Video-MME; derived latency falls in all four Table 2 settings. The fixed-budget component and retention sweeps in Figure 4 further identify how aggregation and downstream compactness affect task quality and device cost.

5.3 Component-Level Realization

Figure 4 uses five tasks: MATH-Vision, RealWorldQA, and Video-MME short/medium/long, all compared with the same fixed compact-128 reference. Average pooling lowers the median native score by 0.06 relative to learned aggregation while reducing derived latency by 3.0%. Restoring dense downstream tokens raises the median visual-prefill count by 48.4% and derived latency by 4.5%, with positive paired timing intervals on all five tasks. This control directly tests whether compactness at later interfaces contributes to device speed.

Removing optional visual-KV pooling reduces the median derived latency by 17.9% with zero median native-score difference; paired timing intervals are negative across the five tasks. This is a compact-128 visual-attention control, distinct from decoder caching and the 256-state headline configuration. Appendix D reports the task-level effects.

Figure 4C places the highest aggregate-aligned coordinate at 128 states (0.517, 1.451 s). The 64-state point is faster (0.497, 1.366 s), while both larger budgets cost more time and fall below the 128-state quality coordinate. This fixed-configuration sensitivity uses a separate setting from the benchmark-specific headline results. The plotted coordinates and additional retention controls appear in Appendix D.

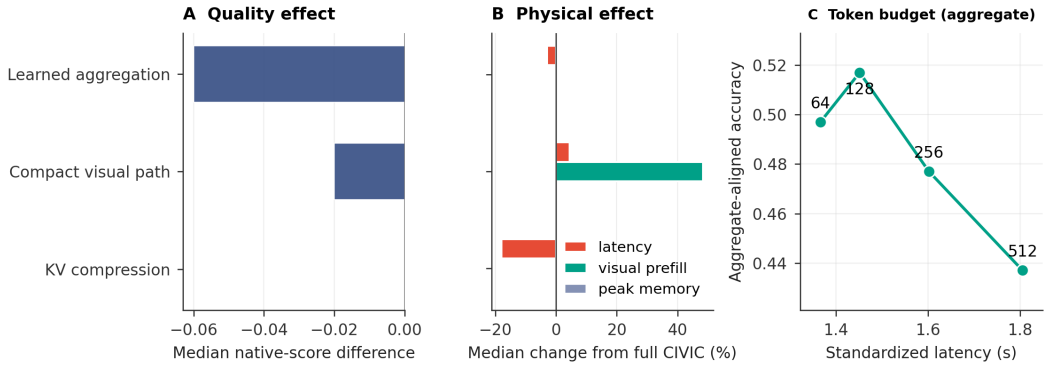

Figure 4: Five-task component controls around the fixed compact-128 reference (MATH-Vision, RealWorldQA, and Video-MME short/medium/long). Rows identify the component tested: from top to bottom, the plotted controls replace learned aggregation with average pooling, restore dense visual prefill, and disable compact visual KV. (A) Median control-minus-reference native-score difference. (B) Median changes in derived latency, visual-prefill count, and peak memory for those controls. (C) Aggregate-aligned score coordinate versus derived latency at four nominal compact-token budgets; Appendix Table 8 reports the plotted coordinates.

6 Discussion

Realization, controls, and portability. The first Qwen visual blocks remain dense, but later encoder blocks and language prefill process shorter sequences; measured times for both stages fall despite the cost of aggregation. Restoring dense visual placeholders in the fixed-budget control increases prefill length and derived latency, isolating the contribution of a compact multimodal interface. Removing optional visual-KV pooling further reduces device time, illustrating why an additional attention shortcut must repay its operator cost. These controls link the structural path to realized execution without assuming that token count alone predicts latency. Task scores are lower on MMMU and MathVision and higher or similar on RealWorldQA and Video-MME, respectively; deployments must consider task quality alongside stage-wise timing. Porting CIVIC to another backbone requires consistent grid positions, merger groups, placeholders, and intermediate features without recovering a dense prefix.

Video duration and deployment. CIVIC lowers Video-MME derived latency for short, medium, and long clips. This consistency across input lengths complements the dense-restoration control: retaining compact visual placeholders saves downstream work, while the full-benchmark score measures video-question quality.

7 Conclusion

CIVIC demonstrates how visual-token compression can yield batch-one edge acceleration when compact representations persist through costly model interfaces. Intermediate aggregation shortens the remaining visual encoder, projector, and language-model prefill without restoring a dense visual span. On Jetson AGX Orin, the median final prefix decreases from 816 to 470 tokens and derived response-length-standardized latency falls by 26.6%. Stage timings locate savings in encoding and prefill; the dense-restoration control confirms that downstream compactness contributes to device speed. Together, these results connect structural reduction to realized inference cost. Task-dependent quality motivates adaptive retention and distillation for more demanding benchmarks. Future work will test how consistently the compact path and its device benefits transfer to newer backbones and edge platforms.

AI USE STATEMENT

Generative AI assisted literature discovery and manuscript polishing. The authors are responsible for all reported measurements, citations, and claims.

REPRODUCIBILITY STATEMENT

Section 4 and Appendices A–D provide the training, device measures, coverage, and component controls.

ETHICS STATEMENT

The study uses public benchmarks. Safety-critical edge applications require validating the effects of lost visual evidence and monitoring deployments.

REFERENCES

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

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. DivPrune: Diversity-based visual token pruning for large multimodal models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9392–9401, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Adrian Bulat, Yassine Ouali, and Georgios Tzimiropoulos. Compress & Cache: Vision token compression for efficient generation and retrieval. In *Advances in Neural Information Processing Systems*, pages 36034–36059, 2025.

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

Mengzhao Chen, Wenqi Shao, Peng Xu, Mingbao Lin, Kaipeng Zhang, Fei Chao, Rongrong Ji, Yu Qiao, and Ping Luo. DiffRate: Differentiable compression rate for efficient vision transformers. In *IEEE/CVF International Conference on Computer Vision*, pages 17164–17174, 2023.

Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, and Chunhua Shen. MobileVLM: A fast, strong and open vision language assistant for mobile devices, 2023. URL `https://arxiv.org/abs/2312.16886`.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*, pages 16344–16359, 2022.

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. In *International Conference on Learning Representations*, 2021.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24108–24118, 2025.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Lianyu Hu, Liqing Gao, Fanhua Shang, Liang Wan, and Wei Feng. iLLaVA: An image is worth fewer than 1/3 input tokens in large multimodal models. In *International Conference on Learning Representations*, 2026.

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/59c147c7d4fdb732daea3064eab949bf-Abstract-Conference.html.

Wenxuan Huang, Zijie Zhai, Yunhang Shen, Shaosheng Cao, Fei Zhao, Xiangfeng Xu, Zheyu Ye, and Shaohui Lin. Dynamic-LLaVA: Efficient multimodal large language models via dynamic vision-language context sparsification. In *International Conference on Learning Representations*, 2025.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pages 4651–4664, 2021.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. ZOO-Prune: Training-free token pruning via zeroth-order gradient estimation in vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 39572–39582, 2026.

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

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, pages 19730–19742, 2023.

Kevin Y. Li, Sachin Goyal, João D. Semedo, and J. Zico Kolter. Inference optimal VLMs need fewer visual tokens and more parameters. In *International Conference on Learning Representations*, 2025a.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. TokenPacker: Efficient visual projector for multimodal LLM. *International Journal of Computer Vision*, 133(10):6794–6812, 2025b.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. SnapKV: LLM knows what you are looking for before generation. In *Advances in Neural Information Processing Systems*, pages 22947–22970, 2024.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, pages 34892–34916, 2023.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. DynamicViT: Efficient vision transformers with dynamic token sparsification. In *Advances in Neural Information Processing Systems*, pages 13937–13949, 2021.

Michael S. Ryoo, A. J. Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. TokenLearner: Adaptive space-time tokenization for videos. In *Advances in Neural Information Processing Systems*, pages 12786–12797, 2021.

Dezhan Tu, Danylo Vashchilenko, Yuzhe Lu, and Panpan Xu. VL-Cache: Sparsity and modality-aware KV cache compression for vision-language model inference acceleration. In *International Conference on Learning Representations*, 2025.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and Hadi Pouransari. FastVLM: Efficient vision encoding for vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19769–19780, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Vasu_FastVLM_Efficient_Vision_Encoding_for_Vision_Language_Models_CVPR_2025_paper.html.

Hanshi Wang, Yuhao Xu, Zekun Xu, Jin Gao, Yufan Liu, Weiming Hu, Ke Wang, and Zhipeng Zhang. Each complexity deserves a pruning policy. In *Advances in Neural Information Processing Systems*, pages 130998–131017, 2025a.

Hesong Wang, Xin Jin, Lu Lu, Chenhaowen Li, Jian Chen, Qiang Liu, and Huan Wang. EarlyTom: Early token compression completes fast video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 40559–40568, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/papers/Wang_EarlyTom_Early_Token_Compression_Completes_Fast_Video_Understanding_CVPR_2026_paper.pdf.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, pages 95095–95169, 2024.

Zhenhailong Wang, Senthil Purushwalkam, Caiming Xiong, Silvio Savarese, Heng Ji, and Ran Xu. DyMU: Dynamic merging and virtual unmerging for efficient variable-length VLMs. In *Advances in Neural Information Processing Systems*, pages 81169–81192, 2025b.

Hao Wu, Yingqi Fan, Jinyang Dai, Junlong Tong, Yunpu Ma, and Xiaoyu Shen. HiDrop: Hierarchical vision token reduction in MLLMs via late injection, concave pyramid pruning, and early exit. In *International Conference on Learning Representations*, 2026.

xAI. Grok-1.5 Vision Preview. https://x.ai/news/grok-1.5v, 2024. Introduces the RealWorldQA benchmark.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14593–14603, 2025.

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

Linli Yao, Lei Li, Shuhuai Ren, Lean Wang, Yuanxin Liu, Xu Sun, and Lu Hou. DeCo: Decoupling token compression from semantic abstraction in multimodal large language models, 2024. URL https://arxiv.org/abs/2405.20985.

Hanxun Yu, Wentong Li, Xuan Qu, Song Wang, Junbo Chen, and Jianke Zhu. VisionTrim: Unified vision token compression for training-free MLLM acceleration. In *International Conference on Learning Representations*, 2026.

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

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-Mini: Efficient image and video large multimodal models with one vision token. In *International Conference on Learning Representations*, 2025.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H2O: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, pages 34661–34710, 2023.

A Reproducibility Details

For an input with t states entering an aggregation invocation, the anchor bank is resized to the length in Eq. 4; the same rule is used in training and inference. Assignment rows in Eq. 5 sum to one over dense tokens, and the compact coordinates must be mapped to the backbone's positional pathway.

For optional visual-KV pooling, let $\mathbf{H}^{(l)} \in \mathbb{R}^{M_e \times d_v}$ and $\overline{\mathbf{R}}^{(l)} \in \mathbb{R}^{M_e \times S}$ denote compact states and column-normalized token-to-anchor assignments at visual layer l. One attention head computes

$$\begin{aligned} \mathbf{Q}^{(l)} &= \mathbf{H}^{(l)}\mathbf{W}_Q^{(l)}, & \mathbf{K}^{(l)} &= \mathbf{H}^{(l)}\mathbf{W}_K^{(l)}, & \mathbf{V}^{(l)} &= \mathbf{H}^{(l)}\mathbf{W}_V^{(l)}, \\ \mathbf{K}_c^{(l)} &= \overline{\mathbf{R}}^{(l)\top}\mathbf{K}^{(l)}, & \mathbf{V}_c^{(l)} &= \overline{\mathbf{R}}^{(l)\top}\mathbf{V}^{(l)}, \\ \mathrm{Attn}_c^{(l)} &= \mathrm{softmax}\left(\frac{\mathbf{Q}^{(l)}\mathbf{K}_c^{(l)\top}}{\sqrt{d_h}}\right)\mathbf{V}_c^{(l)}. \end{aligned} \tag{9}$$

Here d_h is the head dimension. The assignments are normalized as

$$\mathbf{R}^{(l)} = \mathrm{softmax}_S\Big(\mathrm{norm}(\mathrm{LN}(\mathbf{H}^{(l)}))\,\mathrm{norm}(\mathrm{LN}(\mathbf{B}^{(l)}))^\top/\tau_R\Big), \qquad \overline{\mathbf{R}}_{is}^{(l)} = \frac{R_{is}^{(l)}}{\sum_j R_{js}^{(l)} + \epsilon}. \tag{10}$$

The anchor matrix is $\mathbf{B}^{(l)} \in \mathbb{R}^{S \times d_v}$. Softmax is applied over anchors, followed by token-axis normalization for stable pooling. Visual KV pooling is optional and disabled in the headline Qwen configuration; the compact-128 component reference separately tests its effect.

The general Qwen training pipeline fine-tunes a dense teacher, fine-tunes a compact student, then distills it from that teacher for each benchmark. The student SFT trains for one epoch with learning rate 5×10^{-6}, weight decay 0.01, per-device batch size one, gradient accumulation four, bfloat16, and seed 42. It uses cosine decay with 3% warmup, a 95/5 train/validation split, and checkpoint selection by minimum held-out loss. During student SFT the visual backbone stays frozen while the compression modules, last four language layers, language normalization, and multimodal MLP are updated. General distillation starts from the benchmark-specific teacher and student checkpoints and selects the student by validation loss. The Qwen compressor acts after vision layer 16 with a nominal local budget of 128 visual states and a zero minimum retention rate. Thus the dense prefix contributes to the full encoder timing, while the lower interface counts characterize the later compact path. The compact-128 component reference instead sets $C_e = 128$ and $\rho_{\min} = 0.1$ to evaluate the components at a fixed budget. DiffRate, DynamicViT, and HiDrop use checkpoints trained on pooled general benchmarks, unlike CIVIC's separate checkpoint per benchmark.

Evaluation uses fixed prompt templates, deterministic decoding, and device synchronization around measured stages. The resulting summaries record interface lengths, stage times, and output lengths used to derive standardized latency.

For Qwen3-VL, the interface adapter must keep the incoming `grid_thw`, compact grid and positional IDs, spatial-merger groups, visual placeholder counts, DeepStack feature sizes, and initialized decoder cache length mutually consistent. Correctness requires valid spatial and temporal ordering, agreement between projected tokens and language placeholders, and DeepStack features aligned to their target language layers. Weighted coordinates from Eq. 5 alone do not specify these native interfaces. These checks define the adapter contract for a subsequent backbone.

B Evaluation Measures and Coverage

Natural-response latency conflates system speed with generated answer length. Equation 8 computes time at a common dense reference length using each run's observed non-decoding cost and mean decode time per token. The reported 26.6% relative reduction is an aggregate of derived comparisons and need not equal one minus the ratio of the two cross-task median T_{std} entries in Table 5. Table 2 reports the corresponding task timings.

For Video-MME, the Dense and CIVIC scores in Table 2 cover the full evaluation. Their device times are arithmetic means of the three duration-group medians below rather than pooled medians across all videos. Figure 3 uses short-duration device time on its horizontal axis; its Dense and CIVIC vertical coordinates give the full-benchmark scores.

Table 3: Video-MME derived standardized latency by duration (s). The equal-duration mean gives the Video-MME device time in Table 2. Bold marks the lower latency.

Duration	Dense	CIVIC (Ours)
Short	2.292	**1.981**
Medium	2.330	**2.050**
Long	2.342	**2.148**
Equal-duration mean	2.321	**2.060**

The measurements separate interface lengths, device times, the encoder-work proxy, and standardized latency. Interface lengths are recorded tensor shapes; device times come from synchronized timers. The encoder-work proxy uses recorded shapes; standardized latency and relative changes combine recorded times and output lengths. Projector counts for DiffRate and DynamicViT use the Dense Qwen merger ratio, as noted in Table 5. The encoder-work proxy and derived latency describe different quantities.

Figure 1C summarizes task scores across available settings; Figure 3 presents scores and device latency for each setting. The mean in Figure 1C averages over available tasks for each method, whereas Table 2 gives task-specific Dense and CIVIC results. Cross-task device aggregates in Figure 1 and Table 5 include ODinW-13 and three separate video duration groups; the scored benchmark table instead consolidates Video-MME. Red rings denote partial OOM on the Jetson: high-token intermediate layers can exceed available memory.

C Additional Method Comparisons

Table 4: Available-run median device stage time (ms) on Jetson AGX Orin. Each entry is computed over its own finite-run population; stage medians are descriptive and are not additive. Bold identifies the lower measured time in the encoder, projector, and prefill stages; compression overhead is shown separately.

Stage	Dense	CIVIC (Ours)
Compression overhead	0.0	23.7
Visual encoder	573.7	**479.3**
Visual projector	5.5	**3.3**
Language-model prefill	1187.8	**810.1**

Table 5: Cross-task device summaries corresponding to Figure 1. Lengths and T_{std} are marginal medians over available records; projector counts for DiffRate and DynamicViT use the Dense merger ratio. R_F is the encoder-work proxy; R_T is the separately aggregated derived standardized-latency reduction and is not computed by dividing the two displayed T_{std} medians; the two summaries can even differ in the sign of their relative change. Coverage gives the finite-timing fraction. HiDrop's interface lengths and relative reduction are plotted in Figure 1; its absolute aggregate latency is left blank because its method-level and task-level summaries do not coincide. Bold and underline mark the best and second-best derived-latency entries among methods with complete numeric results; interface lengths are reported for structural comparison and are not quality rankings.

Method	n_e	n_p	Prefix	R_F (%)	T_{std} (s)	R_T (%)	Coverage
Dense	2912	728	816	0.0	2.330	0.0	70.6%
DyMU	2912	728	818	67.0	2.670	−14.6	72.2%
DiffRate	1600	400	490	60.2	<u>1.841</u>	<u>16.9</u>	100%
DynamicViT	1600	400	490	21.2	2.088	−3.4	100%
VisionTrim	2912	728	600	0.0	2.012	12.3	72.2%
ZOOPrune	2912	728	600	0.0	2.524	−8.9	72.2%
iLLaVA	2432	728	278	30.2	2.309	0.9	72.0%
AutoPrune	2464	616	110	99.8	34.360	−80.7	57.1%
HiDrop	1708	422	137	0.0	—	16.6	100%
CIVIC (Ours)	1392	433	470	54.8	**1.589**	**26.6**	100%

D Additional Component and Sensitivity Results

Figure 4 and Table 6 use the *same five-task set*: MATH-Vision, RealWorldQA, Video-MME-long, Video-MME-medium, and Video-MME-short. These duration-specific component scores are separate from the full-benchmark Video-MME score in Table 2. Every control is compared with the learned-aggregation compact-128 reference; quality is control minus reference, and percentage columns are the control's change relative to that reference. The headline and component experiments use different compression budgets, so component effects are interpreted relative to the compact-128 reference.

Table 6: Five-task medians matching Figure 4. Negative quality means the control scores lower, and negative derived latency means it is faster. Bold and underline indicate the best and second-best numerical quality/latency effects among controls; the other columns are diagnostic. Median-of-task percentages are not pooled sample-level effects.

Control	ΔQ	ΔT (%)	ΔN_{vis} (%)	Δ ms/tok (%)	Δ peak mem. (%)
Average pool	−0.060	<u>-2.96</u>	0.0	−2.85	0.00
Dense restore	<u>-0.020</u>	4.51	48.44	−4.16	−0.51
No compact KV	**0.000**	**-17.91**	0.0	−2.44	−0.17

Paired-bootstrap 95% latency intervals exclude zero across all five tasks: the dense-restoration intervals have endpoints from $+24.7$ to $+237.2$ ms, and the no-compact-KV intervals have endpoints from -377.3 to -225.6 ms. These controls share the compact-128 reference.

The budget comparison plotted in Figure 4C is reproduced numerically in Table 8; the five-task component contrasts are separately detailed in Table 7.

The aligned score preserves the observed RealWorldQA quality differences between token budgets relative to the 128-state anchor of 0.517. This visualization contrasts budget-dependent quality and latency without equating the plotted coordinate with a remeasured full-benchmark score.

With the 128-token visual-prefill length fixed, MATH-Vision peaks at $\rho_{\min} = 0.2$ while RealWorldQA varies less across retention floors. For $S \in \{128, 256, 512\}$ visual-KV anchors, $S = 128$ has the lowest measured latency on both tasks.

Table 7: Task-level controls corresponding to Table 6. Each task compares the same compact-128 reference. Quality differences are in native score units and derived latency changes are percentages. Bold and underline rank the best and second-best numerical effects within a task.

Task	Average pool		Dense restore		No compact KV	
	ΔQ	ΔT_{std} (%)	ΔQ	ΔT_{std} (%)	ΔQ	ΔT_{std} (%)
MATH-Vision	0.000	<u>-2.96</u>	<u>+0.020</u>	+16.28	**+0.160**	**-17.91**
RealWorldQA	<u>-0.020</u>	<u>-5.78</u>	-0.040	+11.85	**+0.080**	**-18.47**
Video-MME-L	-0.074	<u>-1.80</u>	**+0.037**	+4.51	<u>0.000</u>	**-17.58**
Video-MME-M	-0.080	<u>-0.92</u>	<u>-0.060</u>	+3.69	**0.000**	**-16.02**
Video-MME-S	<u>-0.060</u>	<u>-4.45</u>	**-0.020**	+1.89	**-0.020**	**-20.02**
Median	-0.060	<u>-2.96</u>	<u>-0.020</u>	+4.51	**0.000**	**-17.91**

Table 8: Plotted token-budget coordinates in Figure 4C. The quality coordinate is aligned at 128 states; latency is the measured RealWorldQA device time. Bold marks the best value and underline marks the second-best in each column.

C_e	Aligned score	T_{std} (s)
64	<u>0.497</u>	**1.366**
128	**0.517**	<u>1.451</u>
256	0.477	1.602
512	0.437	1.805