BEYOND RGB RECONSTRUCTION: PREDICTIVE STATE-TO-TOKEN ADAPTATION FOR VIDEO MLLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances in multimodal large language models (MLLMs) have improved video understanding, motivating more efficient visual interfaces. Conventional pipelines reconstruct compressed video into RGB frames and re-encode them, despite predictive decoders already producing temporally conditioned states. We introduce **PreSTA** (Predictive State-to-Token Adaptation), a receiver-side interface that maps these states into the intermediate visual space of a frozen MLLM. Under the same eight-group output budget as sparse RGB, PreSTA aggregates 32 P-frame states into compact visual groups, bypassing RGB reconstruction and early visual encoding for selected observations while preserving predictive state updates. A compact RGB teacher trained with privileged source-frame supervision guides the codec-domain adapter through feature- and semantic-level alignment; the codec, bitstream, and MLLM remain unchanged. Across four codec operating points on Video-MME and MVBench with Qwen3-VL-8B and Qwen3-VL-4B, PreSTA stays within 0.3 points of a stronger 32-frame RGB-Fusion baseline on both benchmarks and model scales, and improves Video-MME over the standard sparse Reconstructed RGB-8 pipeline by an average of 6.2 and 6.1 percentage points. Relative to RGB-Fusion, it achieves this with 49.2% fewer visual-front-end weights, 88.5% fewer MACs, and 93.4% lower measured latency from decoder outputs to visual block 2.

1 INTRODUCTION

Multimodal large language models (MLLMs) have rapidly advanced visual and video understanding through learned vision–language interfaces (Alayrac et al., 2022; Li et al., 2023b; Liu et al., 2023; Lin et al., 2024; Zhang et al., 2024). Recent systems such as Qwen3-VL (Bai et al., 2025) and InternVL3.5 (Wang et al., 2025) further improve general-purpose video understanding. However, video MLLMs remain expensive because temporally distributed observations must be converted into visual tokens under limited computation and token budgets.

Existing efficiency methods mainly reduce sampled frames or compress visual tokens, e.g., LongVU (Shen et al., 2024) and LLaVA-Mini (Zhang et al., 2025a), but still rely on decoded RGB frames as the visual input. For compressed video, this leaves the upstream reconstruction and vision-encoding pipeline intact. Prior work has begun to reconsider this interface: Bridge (Kao et al., 2025) and CoTAM (Liu et al., 2026) exploit learned image-codec representations, while compressed-domain video methods use I-frames, motion vectors, and residuals for efficient recognition or video MLLMs (Wu et al., 2018; Shou et al., 2019; Chen & Ho, 2022; Zhao et al., 2025; Sarkar et al., 2026). Yet these approaches either operate on independently coded images or explicit conventional-codec primitives, leaving the temporally conditioned internal states of learned predictive video decoders largely unexplored as an interface to MLLMs.

Meanwhile, we note that modern learned predictive video codecs increasingly maintain temporally conditioned representations directly in the feature domain. Representative DCVC models, including DCVC-DC (Li et al., 2023a), DCVC-FM (Li et al., 2024a), and DCVC-UF (Li et al., 2026), propagate or produce decoder-side features that integrate information from the transmitted latent and temporal context before pixel reconstruction. Yet these feature-domain states are typically discarded before downstream visual understanding. This raises a natural question: can they directly serve as the interface between a video codec and an MLLM?

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.

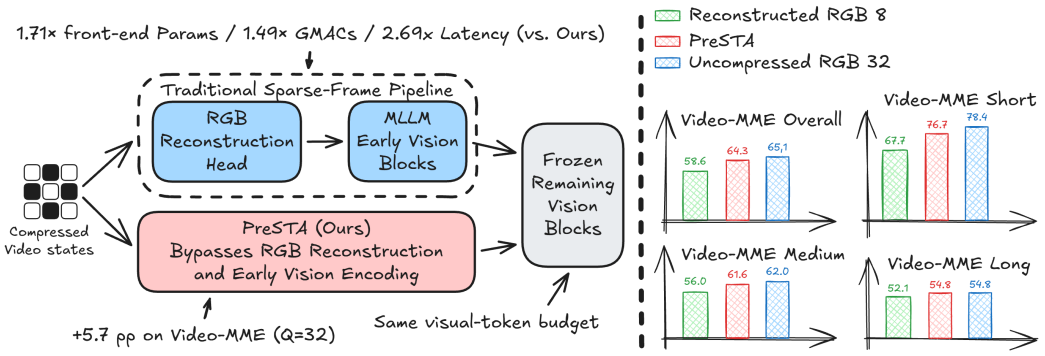

Figure 1: PreSTA and reconstructed RGB-8 use the same bitstream and eight-group visual output budget. Bars show Video-MME accuracy (%) at $q = 32$ with Qwen3-VL-8B; the cost ratios cover only the visual front end, excluding predictive decoding and the downstream MLLM.

To this end, we introduce **PreSTA** (Predictive State-to-Token Adaptation), a receiver-side interface that maps predictive decoder states to an intermediate visual space of a frozen MLLM. As illustrated on the left of Figure 1, the conventional receiver-side route reconstructs sampled observations into RGB frames and re-encodes them through the early vision tower, whereas PreSTA directly reuses predictive decoder states already produced during decoding and maps them to the same visual interface under an identical token budget, bypassing RGB reconstruction and early visual encoding. This substantially reduces the visual front-end parameter count and computation relative to the conventional sparse-frame RGB pipeline. As shown on the right of Figure 1, PreSTA consistently outperforms the reconstructed RGB-8 baseline across different video durations and, on long videos, even matches the uncompressed RGB-32 reference.

Our contributions are summarized as follows:

- We introduce PreSTA, a receiver-side codec–MLLM interface that directly reuses temporally conditioned feature-domain states from predictive neural video decoding, avoiding RGB reconstruction and early visual encoding without modifying the codec, transmitted bitstream, or pretrained MLLM.
- We develop a compact state-to-token adapter that aggregates dense predictive states under a fixed visual-token budget through temporal folding and cross-state fusion, together with a two-stage distillation scheme that bridges the codec and MLLM representation spaces using privileged RGB supervision.
- We evaluate PreSTA on Video-MME and MVBench with Qwen3-VL-8B and Qwen3-VL-4B across four codec operating points. PreSTA substantially outperforms the standard sparse reconstructed-RGB pipeline while closely matching a stronger 32-frame RGB-fusion baseline, with substantially lower visual front-end parameters, computation, and latency.

2 Related Work

Efficient video MLLMs. Prior work reduces video inference cost through frame selection and visual-token compression (Shen et al., 2024; Zhang et al., 2025a). LLaMA-VID compresses frame-level representations, STTM merges visual tokens across space and time, and FiLA-Video fuses information from multiple frames (Li et al., 2024c; Hyun et al., 2025; Guo et al., 2025). These methods operate on decoded frames or their visual features; PreSTA instead forms the MLLM interface from predictive decoder states before selected-frame RGB reconstruction.

Learned image compression for visual understanding. Perception directly from learned compressed representations predates MLLMs: Torfason et al. (2018) classify and segment compressed feature maps without reconstructing RGB. Bridge maps image-codec latents into MLLM visual fea-

tures with a lightweight transform-neck (Kao et al., 2025), whereas CoTAM adapts bit allocation and decoder-side reconstruction to preserve MLLM-relevant semantics (Liu et al., 2026). Recent work also benchmarks VLM robustness to compressed images and improves compressed-image understanding with lightweight adaptation (Zhang et al., 2025b). Unlike these image-focused approaches, PreSTA uses temporally conditioned video-decoder features whose interpretation depends on the propagated predictive state.

Understanding conventionally compressed video. Compressed-domain recognition exploits I-frames, motion vectors, and residuals to avoid dense pixel-domain processing (Wu et al., 2018; Shou et al., 2019; Chen & Ho, 2022), and recent codec-aware Video MLLMs build compact tokens from the same primitives (Zhao et al., 2025; Sarkar et al., 2026). Exploiting codec structure is therefore established; our difference is the representation consumed. PreSTA introduces no encoder for explicit motion or residual streams: it reads neural feature maps that a frozen predictive decoder already produces, and maps them into the intermediate visual space of a frozen MLLM.

Learned predictive video coding. Learned video codecs supply the temporally conditioned features our interface operates on. Starting from end-to-end rate–distortion training (Lu et al., 2019), this line replaced residual coding with conditional feature-domain contexts (Li et al., 2021; 2023a), added cross-rate modulation (Li et al., 2024a), and improved throughput via implicit temporal modeling and chunk-level decoder-feature propagation (Jia et al., 2025; Li et al., 2026); we use the low-delay DCVC-UF configuration with single-frame chunks. These works optimize coding efficiency and runtime. We leave the codec and the transmitted bitstream untouched and only adapt its receiver-side features, so the decoder's state updates remain on the inference path.

3 METHOD

3.1 RECEIVER-SIDE SETTING

Our goal is to reuse the temporally conditioned representations that are already produced during predictive video decoding, rather than returning to RGB before visual understanding. We therefore formulate the problem as a receiver-side interface: the codec, transmitted bitstream, and downstream MLLM remain unchanged, and only the mapping from decoder features to the MLLM visual space is learned (Figure 2).

Predictive decoder features. Let a video be denoted by $\{\mathbf{x}_t\}_{t=0}^{T-1}$, where $\mathbf{x}_0$ is the initial intra frame and the remaining frames are predictively coded. In this work, we focus on P-frames and use their predictive decoder states as the interface to the MLLM; the intra frame follows the original codec reconstruction path and is not included in the proposed adaptation. For a P-frame $\mathbf{x}_t$, the DCVC-UF encoder maps the current frame to a latent representation $\mathbf{y}_t$, which is quantized and transmitted in the bitstream:

$$\mathbf{y}_t = E_{\mathrm{P}}\big(\mathbf{x}_t, \mathbf{C}_t; q_t\big), \qquad \hat{\mathbf{y}}_t = Q(\mathbf{y}_t), \tag{1}$$

where E_{P} denotes the *P-frame Encoder*, $Q(\cdot)$ denotes quantization, and q_t is the operating-point-dependent quantization parameter. At the receiver, $\hat{\mathbf{y}}_t$ is recovered from the bitstream and decoded together with the propagated feature memory m_t. The receiver-side prediction chain is

$$\mathbf{C}_t = E_{\mathrm{ctx}}(m_t), \qquad \mathbf{F}_t = D\big(\hat{\mathbf{y}}_t, \mathbf{C}_t; q_t\big), \qquad m_{t+1} = A_M\big(m_t, \mathbf{F}_t\big), \tag{2}$$

where E_{ctx}, D, and A_M denote the context encoder, feature decoder, and state-update module, respectively, and $\mathbf{F}_t \in \mathbb{R}^{C\times H\times W}$.

In the conventional RGB route, the *RGB Reconstruction* head R further maps the decoder feature to the reconstructed frame,

$$\hat{\mathbf{x}}_t = R(\mathbf{F}_t). \tag{3}$$

As illustrated in Figure 2, PreSTA instead reads $\mathbf{F}_t$ directly before RGB reconstruction. Therefore, for the selected P-frame observations, $R(\cdot)$ and the early vision blocks are bypassed, while the predictive state update $A_M(\cdot)$ remains unchanged.

Unlike independently coded image latents, the conditional latent $\hat{\mathbf{y}}_t$ is meaningful only together with the propagated context $\mathbf{C}_t$. We therefore use the decoded feature $\mathbf{F}_t$, which already integrates

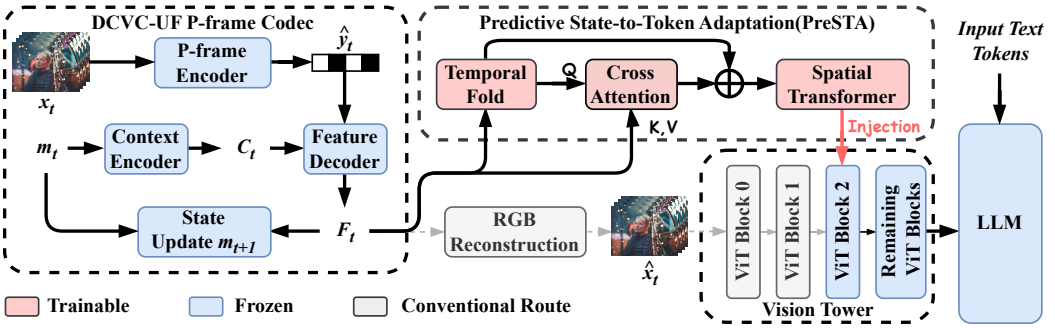

Figure 2: PreSTA architecture. The frozen DCVC-UF decoder produces predictive features $\mathbf{F}_t$ during sequential decoding. For each visual group, PreSTA aggregates an anchor P-frame feature and three preceding P-frame features, then injects the adapted tokens at the input to Qwen3-VL visual block 2. This bypasses RGB reconstruction and early visual encoding for the selected observations; block 2 onward and the LLM remain frozen.

both sources of information, rather than $\hat{\mathbf{y}}_t$ alone (Table 3). Moreover, predictive decoding remains sequential: the state m_t must be updated along the prediction chain before $\mathbf{F}_t$ becomes available. Hence, PreSTA reuses intermediate features produced by the required decoding process rather than eliminating predictive decoding itself. The intra frame and feature-memory resets follow the original codec path unchanged. The low-delay codec configuration is detailed in Appendix B.2.

3.2 PRESTA

PreSTA maps predictive decoder features into eight visual groups (Figure 2). We use the same eight P-frame anchor positions $\{a_g\}_{g=0}^{7}$ as Reconstructed RGB-8. Because reconstructing an anchor requires the predictive decoder to update its state up to a_g, preceding decoder features are also available. For each group g, we collect the anchor and three preceding P-frame features, denoted by $\{\mathbf{F}^s_{g,k}\}_{k=1}^{4}$, with $\mathbf{F}^s_{g,4} = \mathbf{F}_{a_g}$. This gives 32 feature slots organized into eight causal groups. The adapter parameters are shared across groups, and each group is processed independently. Missing preceding P-frame slots in short videos are handled as described in Appendix B.3.

Temporal fold. We first reduce the spatial resolution of each feature using 2×2 average pooling with stride 2:

$$\mathbf{P}_{g,k} = \mathrm{AvgPool}_{2\times 2}\left(\mathbf{F}^s_{g,k}\right), \qquad \mathbf{P}_{g,k} \in \mathbb{R}^{C\times \frac{H}{2}\times \frac{W}{2}}, \quad k = 1, \ldots, 4. \tag{4}$$

We use three preceding frames per anchor; Appendix Table 5 compares context lengths in an original-RGB-grid proxy. The four pooled features are then packed into a fixed 2×2 spatial layout according to their temporal order:

$$\mathbf{M}_g = \mathrm{Fold}_{2\times 2}\left(\mathbf{P}_{g,1}, \mathbf{P}_{g,2}, \mathbf{P}_{g,3}, \mathbf{P}_{g,4}\right) = \begin{bmatrix} \mathbf{P}_{g,1} & \mathbf{P}_{g,2} \\ \mathbf{P}_{g,3} & \mathbf{P}_{g,4} \end{bmatrix}, \qquad \mathbf{M}_g \in \mathbb{R}^{C\times H\times W}. \tag{5}$$

The concatenation is performed along the spatial dimensions rather than the channel dimension. The anchor occupies a fixed quadrant across all groups, providing a consistent temporal–spatial ordering. This allows four predictive observations to share the spatial footprint of one feature map without increasing the number of visual groups.

Cross attention. Temporal folding preserves four observations within one spatial footprint, but pooling discards fine-grained details. We therefore use the original unpooled features to refine $\mathbf{M}_g$. For each k, $\mathrm{Flat}(\mathbf{F}^s_{g,k})$ converts the $C \times H \times W$ feature map into HW spatial tokens of width C. Concatenating these four token sequences in temporal order gives

$$\mathbf{U}_g = \mathrm{Concat}_{k=1}^{4}\,\mathrm{Flat}(\mathbf{F}^s_{g,k}) \in \mathbb{R}^{4HW\times C}. \tag{6}$$

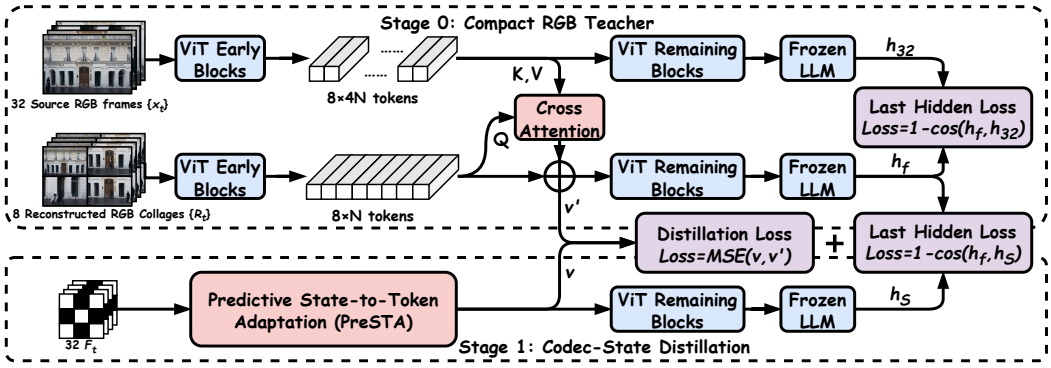

Figure 3: Two-stage training of PreSTA. Stage 0 trains a compact RGB teacher by fusing reconstructed-frame collages with individually encoded source-frame features, and aligns its MLLM hidden state with a 32-source-frame reference. Stage 1 freezes the teacher and trains PreSTA on 32 predictive decoder features using token- and hidden-state alignment, followed by feature-only refinement. Source frames provide privileged training supervision and are unavailable at inference; the codec and MLLM remain frozen throughout.

The concatenation is along the token dimension: $\mathbf{U}_g$ retains all spatial positions from all four observations and serves as the keys and values, while the folded feature $\mathrm{Flat}(\mathbf{M}_g) \in \mathbb{R}^{HW \times C}$ provides the queries. Let R_ϕ denote a multi-head cross-attention module with learnable parameters ϕ, using the folded representation as queries and $\mathbf{U}_g$ as both keys and values:

$$\widetilde{\mathbf{M}}_g = \mathbf{M}_g + \mathrm{Unflat}\left[R_\phi\left(\mathrm{Flat}(\mathbf{M}_g), \mathbf{U}_g, \mathbf{U}_g\right)\right]. \tag{7}$$

This coarse-to-fine refinement follows the principle of TokenPacker (Li et al., 2025), while both sides here originate from the same predictive states at different spatial resolutions. The output projection of R_ϕ is zero-initialized, so the module starts from the folded representation and learns only a residual correction.

Spatial neck and injection. Following Bridge (Kao et al., 2025), we use its lightweight transform-neck to map the refined codec feature $\widetilde{\mathbf{M}}_g$ into the intermediate visual space of Qwen3-VL:

$$\mathbf{Z}_g = \mathcal{N}_\phi(\widetilde{\mathbf{M}}_g) \in \mathbb{R}^{N \times d}, \tag{8}$$

where $\mathcal{N}_\phi$ denotes the spatial transform-neck. For 256×256 inputs, each group produces $N = 256$ tokens with $d = 1152$; N scales with spatial resolution. We inject $\mathbf{Z}_g$ at visual block 2, bypassing blocks 0–1 for the selected P-frame observations.

3.3 Two-Stage Training

We train PreSTA in two stages (Figure 3). Stage 0 constructs a compact RGB teacher; Stage 1 distills the teacher's representation into the codec-domain adapter. DCVC-UF and Qwen3-VL remain frozen throughout.

Stage 0: compact RGB teacher construction. Following the image-grid layout of IG-VLM (Kim et al., 2024), we arrange 32 reconstructed frames into eight 2×2 collages. To retain details that may be lost in a collage, we fuse each collage with features of its four individually encoded source frames. Let $E_{<2}$ be the frozen Qwen3-VL vision prefix before block 2. For group g, the collage feature $\mathbf{Q}_g^T = E_{<2}(\hat{\mathbf{c}}_g)$ serves as the query, and the source-frame features $\mathbf{U}_g^T = \mathrm{Concat}_{k=1}^4 E_{<2}(\mathbf{x}_{g,k})$ serve as keys and values. The teacher tokens are

$$\mathbf{Z}_g^T = \mathbf{Q}_g^T + R_\theta\left(\mathbf{Q}_g^T, \mathbf{U}_g^T, \mathbf{U}_g^T\right). \tag{9}$$

Let $\mathbf{h}_{32}(v, p)$ be the final hidden state of the frozen MLLM for 32 individually encoded source frames and prompt p. Let $\mathbf{h}_f(v, p)$ be the corresponding state for the eight teacher groups and the

same prompt. We train only R_θ with

$$\mathcal{L}^{(0)} = \mathbb{E}_{v,p}\left[1 - \cos(\mathbf{h}_f(v,p), \mathrm{sg}(\mathbf{h}_{32}(v,p)))\right]. \tag{10}$$

Here sg denotes stop-gradient. Source RGB frames provide privileged training information and are unavailable at inference. The source-frame branch is training-only, so the teacher's accuracy is not an attainable upper bound for the student.

Stage 1: codec-state distillation. We freeze the RGB teacher and train PreSTA on predictive features from the same 32 temporal positions. Its output $\mathbf{Z}_g$ and the teacher output $\mathbf{Z}_g^T$ share the visual interface at block 2. We align their tokens and the corresponding final hidden states:

$$\mathcal{L}_{\mathrm{MSE}} = \mathbb{E}_v\left[\frac{1}{GNd}\sum_{g=0}^{G-1}\left\|\mathbf{Z}_g - \mathrm{sg}(\mathbf{Z}_g^T)\right\|_F^2\right], \tag{11}$$

$$\mathcal{L}_{\mathrm{hid}}^{(1)} = \mathbb{E}_{v,p}[1 - \cos(\mathbf{h}_s(v,p), \mathrm{sg}(\mathbf{h}_f(v,p)))], \tag{12}$$

Here $G = 8$, and $\mathbf{h}_s(v,p)$ denotes the final hidden state at the answer position, obtained by passing PreSTA tokens and prompt p through the frozen vision suffix and LLM.

In the first phase, we jointly minimize $\mathcal{L}_{\mathrm{MSE}} + \lambda\mathcal{L}_{\mathrm{hid}}^{(1)}$ with $\lambda = 0.2$. This aligns both the tokens at the block 2 interface and the resulting MLLM hidden state with the RGB teacher. We then continue training with $\mathcal{L}_{\mathrm{MSE}}$ alone to refine the token-level alignment.

Only the codec cross-attention R_ϕ and spatial neck $\mathcal{N}_\phi$ are updated in either phase. Training schedules are given in Appendix B.5.

4 Experiments

4.1 Experimental Setup

We train on Kinetics-400 (Kay et al., 2017) and evaluate on Video-MME (Fu et al., 2025) and MVBench (Li et al., 2024b) using VLMEvalKit (Duan et al., 2024) with benchmark-specific evaluation protocols. We use Qwen3-VL-8B and Qwen3-VL-4B to assess performance across two model scales.

We use the low-delay variant of DCVC-UF Li et al. (2026) with intra-period $= -1$, corresponding to a single initial I-frame followed by predictive coding, and a feature-memory reset interval of 128 frames. Four codec quality settings, $q \in \{0, 11, 21, 32\}$, are evaluated, with a separate teacher and codec-to-token adapter trained for each operating point. All compressed-input methods share eight P-frame anchors and an eight-group visual-token budget. RGB-Grid, RGB-Fusion, and PreSTA also use the same 32 temporal positions; the uncompressed 16- and 32-frame references have larger visual-token budgets.

For rate–accuracy evaluation, bitrate is measured from the complete coded bitstream, including the initial I-frame and stream overhead, and normalized by the total number of source pixels. Additional training, preprocessing, token-accounting, and implementation details are provided in the supplementary material.

Baselines. We compare against several reconstructed-RGB and codec-domain baselines. *Reconstructed RGB-8* follows the standard sparse-frame pipeline with eight reconstructed frames. *Reconstructed RGB-Grid* reconstructs the same 32 temporal positions used by PreSTA and groups every four frames into one 2×2 collage, matching both temporal coverage and the eight-group visual-token budget. *Bridge-style* applies an image-style codec-to-MLLM adapter to temporally folded predictive decoder features, using Reconstructed RGB-Grid as the supervision teacher. *Reconstructed RGB-Fusion* serves as our primary RGB baseline. It follows the Stage 0 fusion design, using eight reconstructed RGB collages as queries and 32 reconstructed-frame features as keys and values. Unlike Stage 0, which uses source RGB as privileged training information, RGB-Fusion uses only receiver-available reconstructed RGB and the same training objective as PreSTA Stage 0. We also report uncompressed RGB references with 8, 16, and 32 individually encoded frames. At the same per-frame resolution, their pre-merger visual-token budgets scale as $1\times$, $2\times$, and $4\times$, respectively.

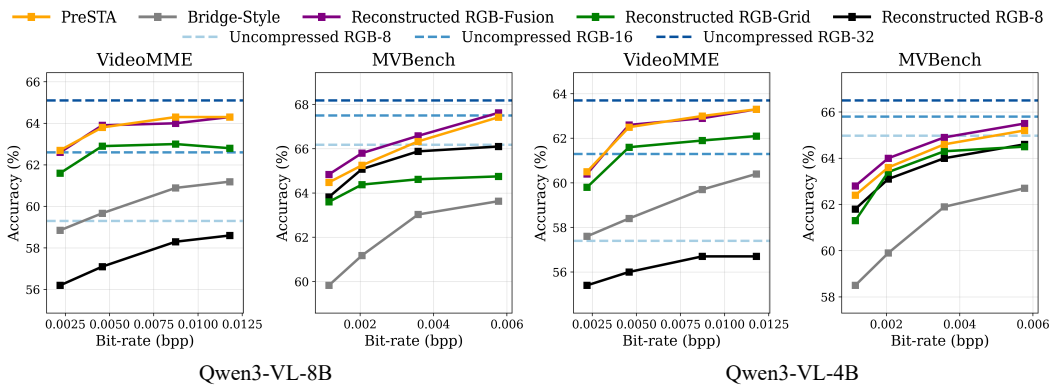

Figure 4: Rate–accuracy comparison on Video-MME and MVBench using Qwen3-VL-8B and Qwen3-VL-4B. Solid curves denote compressed-input methods, while dashed lines show uncompressed RGB references with 8, 16, and 32 individually encoded frames.

Table 1: Mean accuracy (%) over the four codec operating points on Qwen3-VL-8B and Qwen3-VL-4B, and visual front-end complexity. Front-end cost is measured at 720p up to the input of Qwen3-VL visual block 2; shared codec decoding and the downstream MLLM are excluded.

	Qwen3-VL-8B		Qwen3-VL-4B		Front-end Cost		
Method	V-MME	MVBench	V-MME	MVBench	Active Params (M)	GMACs	Time (ms)
Recon. RGB-8	57.6	65.2	56.2	63.4	35.6	1449.4	48.3
Recon. RGB-Grid	62.6	64.3	61.4	63.4	35.6	1705.4	80.5
Recon. RGB-Fusion	63.7	**66.2**	**62.3**	**64.3**	41.0	8448.8	273.9
PreSTA	**63.8**	65.9	**62.3**	64.0	**20.8**	**971.0**	**18.0**

4.2 Main Results

Figure 4 compares rate–accuracy performance across four codec operating points on Video-MME and MVBench with Qwen3-VL-8B and Qwen3-VL-4B. Averaged over these points, PreSTA improves Video-MME over Reconstructed RGB-8 by 6.2 and 6.1 percentage points, respectively. On MVBench, PreSTA is on par with Reconstructed RGB-8 (65.9 vs. 65.2 and 64.0 vs. 63.4) and within 0.3 points of RGB-Fusion; we attribute the smaller margin to MVBench's focus on short, perception-oriented clips, where PreSTA's extra temporal coverage is less decisive than on Video-MME's long-form temporal-reasoning items. It also outperforms Bridge-style across all operating points, benchmarks, and model scales. Relative to Reconstructed RGB-8, PreSTA cuts visual front-end weights, MACs, and latency by 41.6%, 33.0%, and 62.8% (Table 1).

PreSTA closely matches the Reconstructed RGB-Fusion baseline across both benchmarks and model scales, although RGB-Fusion reconstructs and individually encodes 32 RGB frames before feature fusion while PreSTA reuses predictive decoder features directly. Compared with RGB-Fusion, PreSTA reduces front-end parameters from 41.0M to 20.8M (49.2%), computation from 8448.8G to 971.0G MACs (88.5%), and measured front-end latency from 273.9 ms to 18.0 ms (93.4%). Because both routes consume the same coded bitstream and differ only in the receiver-side interface, this parity indicates that the temporally conditioned decoder state already carries the MLLM-relevant information for the selected observations. The adapter that achieves this is trained on Kinetics-400 only, and neither the codec, the bitstream, nor the MLLM is modified.

We count only the modules mapping decoder outputs to Qwen3-VL visual block 2: for RGB-Fusion, eight collage encodings plus 32 individual reconstructed-frame encodings and their fusion. The shared DCVC-UF predictive decoding and state update, and the downstream vision and language modules, are excluded for all methods. Latency is measured at 720p on an NVIDIA RTX 6000 Ada (five warm-up plus 20 timed runs, averaged), so the numbers quantify the visual front-end cost rather than end-to-end inference. A per-module breakdown is given in the appendix (Section B.6).



Figure 5: Qualitative VideoQA comparison: the fourth row is a shared visualization of PreSTA and reconstructed RGB-Fusion, where each fusion view combines information from four temporal positions; the RGB images are illustrative, as PreSTA operates on predictive decoder features.

4.3 Qualitative Example

Figure 5 presents an example that requires fine-grained temporal evidence and reading small on-screen scoreboards. While the uncompressed RGB-8 input predicts the correct answer (B), both Reconstructed RGB-8 and Reconstructed RGB-Grid predict C. Using predictive decoder states from the same 32 temporal positions, PreSTA preserves the relevant evidence and predicts B with only eight visual groups. Reconstructed RGB-Fusion also succeeds, but requires reconstructing and individually encoding all 32 frames before feature fusion. This example illustrates that PreSTA can recover temporally distributed visual evidence directly from predictive decoder states, without relying on the RGB-Fusion front end.

Table 2: Stage 0 RGB-teacher ablation with Qwen3-VL-8B at $q = 32$. The last two rows use the same RGB-Grid query and residual-fusion module, with reconstructed or source-frame features as keys and values. The source-frame branch is used only during training.

RGB representation	Video-MME Acc. (%)	MVBench Acc. (%)
Reconstructed RGB-8	58.6	66.1
Reconstructed collage Grid	62.8	64.8
Grid + Reconstructed-frame fusion	64.3	67.3
Grid + Source-frame fusion (Stage 0 teacher)	**64.8**	**67.6**

4.4 Ablations

All design choices below are ablated on Video-MME at q = 32 with Qwen3-VL-8B; the selected configuration is then applied unchanged to all other operating points, benchmarks, and model scales.

Stage 0 teacher construction. Table 2 examines the compact RGB teacher. Packing 32 reconstructed frames into eight collages is not a free win: it helps Video-MME but slightly hurts MVBench, because a collage is patchified under the same token budget as one frame, so each observation keeps roughly a quarter of the tokens and the frozen tower sees a montage rather than a natural image. Residual fusion over full-resolution frame features removes that loss and improves both benchmarks further. Replacing the reconstructed keys and values with privileged source-frame features adds only a small increment, so the teacher's advantage comes mainly from the fusion architecture rather than from information unavailable at the receiver; the RGB-Fusion baseline is therefore a fair reference.

Table 3: Stage 1 student and teacher design ablation on Video-MME with Qwen3-VL-8B at $q = 32$. "Input slots" counts codec-domain observations (eight anchors or four per group) and all teachers are training-only. Source RGB-8 encodes eight uncompressed source frames, RGB-Grid eight collages of 32 reconstructed frames, and the Stage 0 teacher additionally fuses source-frame features.

Student input / adapter	Input slots	Training teacher	Acc. (%)
$\hat{\mathbf{y}}_t$ (anchor only)	8	Source RGB-8	51.3
$\mathbf{F}_t$ (anchor only)	8	Source RGB-8	58.9
$\mathbf{F}_t$ (anchor only)	8	Reconstructed RGB-Grid	50.9
$\mathbf{F}_t$ (anchor only)	8	Stage 0 Teacher	51.3
$\mathbf{F}_t$ + temporal fold	32	Reconstructed RGB-Grid	61.2
$\mathbf{F}_t$ + temporal fold	32	Stage 0 Teacher	62.6
$\mathbf{F}_t$ + temporal fold + cross-attention (**PreSTA**)	32	Stage 0 Teacher	**64.3**

Stage 1 student and teacher design. Table 3 compares student inputs and training teachers under the same eight-group output interface. Temporal coverage dominates: adding three preceding states per anchor gives the largest gain. Decoding the predictive feature rather than the compressed latent also helps, indicating that the decoded state is the better alignment target. Teacher strength helps only when the student can express the teacher's layout: the higher-scoring RGB-Grid and Stage 0 teachers reduce accuracy for an anchor-only student, which cannot reproduce their four-observation 2×2 folding; the penalty disappears once folded, and residual cross-attention reaches the best accuracy of 64.3%.

Table 4: Ablation of the Stage 0 and Stage 1 loss on Video-MME with Qwen3-VL-8B at $q = 32$.

Stage 0	Stage 1 objective	Acc. (%)
✗	$\mathcal{L}_{\text{MSE}}$ only	60.6%
✗	$\mathcal{L}_{\text{MSE}} + \lambda\mathcal{L}_{\text{hid}}$	61.2%
✓	$\mathcal{L}_{\text{MSE}}$ only	63.6%
✓	$\mathcal{L}_{\text{MSE}} + \lambda\mathcal{L}_{\text{hid}}$ ($\lambda = 1$)	62.6%
✓	$\mathcal{L}_{\text{MSE}} + \lambda\mathcal{L}_{\text{hid}}$ ($\lambda = 0.1$)	63.8%
✓	$\mathcal{L}_{\text{MSE}} + \lambda\mathcal{L}_{\text{hid}}$ ($\lambda = 0.2$, **PreSTA**)	**64.3%**

Training objectives. Table 4 compares Stage 0 training and the Stage 1 hidden-state loss. With Stage 0 disabled, the adapter is unchanged and the fused Stage 0 teacher is replaced by the plain Reconstructed RGB-Grid teacher; the two differ in both their fusion module and their privileged source frames. Stage 0 is the dominant factor, accounting for most of the total improvement in the table; we attribute this to the fused teacher providing a better alignment target, not to the privileged source frames that also shaped it (Table 2). The hidden-state loss complements rather than substitutes for it: it yields a similar small gain with or without Stage 0, and helps most at a moderate weight, whereas letting it dominate ($\lambda = 1$) overrides the token-level alignment that defines the interface, since its signal is downstream and weaker. We therefore use $\lambda = 0.2$ during joint alignment and finish with feature-only refinement.

5 Conclusion

We presented PreSTA, a receiver-side interface that maps predictive video-decoder states into an intermediate visual space of a frozen MLLM without modifying the codec, the bitstream, or the model. By aggregating causal decoder features and learning from a compact RGB teacher, it avoids RGB reconstruction and early visual re-encoding for selected observations. Across Video-MME and MVBench, PreSTA matches a stronger 32-frame RGB-Fusion baseline to within 0.3 points while reducing front-end MACs by 88.5% and measured front-end latency by 93.4%. The interface still requires sequential predictive decoding, and its behavior under periodic intra refresh remains untested; the reported savings also cover the visual front end only, since end-to-end latency is dominated by predictive decoding and the downstream MLLM. Since PreSTA only requires a predictive decoder exposing pre-reconstruction states and a block-structured vision tower with an injectable intermediate layer—properties shared by DCVC-DC, DCVC-FM, and DCVC-UF—transferring the adapter across codecs, quality settings, and other MLLM backbones is a natural direction for future work.

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. *Advances in neural information processing systems*, 35:23716–23736, 2022.

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.

Jiawei Chen and Chiu Man Ho. Mm-vit: Multi-modal video transformer for compressed video action recognition. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 786–797. IEEE, 2022.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 11198–11201, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118. IEEE, 2025.

Yanan Guo, Wenhui Dong, Jun Song, Shiding Zhu, Xuan Zhang, Hanqing Yang, Yingbo Wang, Yang Du, Xianing Chen, and Bo Zheng. Fila-video: Spatio-temporal compression for fine-grained long video understanding. *arXiv preprint arXiv:2504.20384*, 2025.

Jeongseok Hyun, Sukjun Hwang, Su Ho Han, Taeoh Kim, Inwoong Lee, Dongyoon Wee, Joon-Young Lee, Seon Joo Kim, and Minho Shim. Multi-granular spatio-temporal token merging for training-free acceleration of video llms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 23990–24000. IEEE, 2025.

Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12543–12552. IEEE, 2025.

Chia-Hao Kao, Cheng Chien, Yu-Jen Tseng, Yi-Hsin Chen, Alessandro Gnutti, Shao-Yuan Lo, Wen-Hsiao Peng, and Riccardo Leonardi. Bridging compressed image latents and multimodal large language models. In *International Conference on Learning Representations*, volume 2025, pp. 7966–7986, 2025.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Wonkyun Kim, Changin Choi, Wonseok Lee, and Wonjong Rhee. An image grid can be worth a video: Zero-shot video question answering using a vlm. *IEEE Access*, 12:193057–193075, 2024.

Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *Advances in Neural Information Processing Systems*, 34:18114–18125, 2021.

Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22616–22626. IEEE, 2023a.

Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26099–26108. IEEE, 2024a.

Jiahao Li, Wenxuan Xie, Zhaoyang Jia, Bin Li, Zongyu Guo, Xiaoyi Zhang, and Yan Lu. Ultra-fast neural video compression. *arXiv preprint arXiv:2606.04410*, 2026.

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.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, et al. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22195–22206, 2024b.

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, 2025.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European conference on computer vision*, pp. 323–340. Springer, 2024c.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pp. 5971–5984, 2024.

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

Jinming Liu, Zhaoyang Jia, Jiahao Li, Bin Li, Xin Jin, Wenjun Zeng, and Yan Lu. When mllms meet compression distortion: A coding paradigm tailored to mllms. In *International Conference on Learning Representations*, volume 2026, pp. 7426–7449, 2026.

Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10998–11007. IEEE, 2019.

Sayan Deb Sarkar, Rémi Pautrat, Ondrej Miksik, Marc Pollefeys, Iro Armeni, Mahdi Rad, and Mihai Dusmanu. Cope-videolm: Leveraging codec primitives for efficient video language modeling. *arXiv preprint arXiv:2602.13191*, 2026.

Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, et al. Longvu: Spatiotemporal adaptive compression for long video-language understanding. *arXiv preprint arXiv:2410.17434*, 2024.

Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1268–1277. IEEE, 2019.

Robert Torfason, Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Towards image understanding from deep compression without decoding. *arXiv preprint arXiv:1803.06131*, 2018.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 6026–6035. IEEE, 2018.

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*, volume 2025, pp. 53285–53310, 2025a.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

Zifu Zhang, Tongda Xu, Siqi Li, Shengxi Li, Yue Zhang, Mai Xu, and Yan Wang. Benchmarking and enhancing vlm for compressed image understanding. *arXiv preprint arXiv:2512.20901*, 2025b.

Zijia Zhao, Yuqi Huo, Tongtian Yue, Longteng Guo, Haoyu Lu, Bingning Wang, Weipeng Chen, and Jing Liu. Efficient motion-aware video mllm. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24159–24168. IEEE, 2025.

A LLM USAGE

We used LLMs for manuscript language refinement, feedback on experimental comparisons and interpretation of results, and generation of auxiliary prompt–answer pairs from training metadata. Specifically, GPT-5.6 Sol was used to generate the auxiliary training prompt–answer pairs described in Appendix B.1. All AI-assisted material was reviewed by the authors, who verified the reported results and made all final methodological and scientific decisions.

B ADDITIONAL EXPERIMENTAL DETAILS

B.1 DATASETS AND EVALUATION PROTOCOLS

Training data. We use Kinetics-400 for training the Stage 0 teacher and the Stage 1 codec-to-token adapter. We use 234,619 videos for training and 19,761 videos for validation. No samples from Video-MME or MVBench are used to train the Stage 0 teacher or the Stage 1 adapter. Videos are decoded and processed using the same temporal sampling procedure described below.

Prompt and last-hidden construction. We train with generative question–answer prompts rather than multiple-choice options, since option-based prompts are too easy and provide a weak alignment signal. For each video we build two prompt sets and sample one uniformly at each training step. The first set contains 10 action-question templates derived from the original Kinetics-400 class label, answered with the free-form class name. The second set is produced by OpenAI's GPT-5.6 Sol, which generates 10 diverse prompt–answer pairs per video to avoid an action-only bias. For every sampled prompt, the last-hidden representation used in the alignment losses is the ℓ_2-normalized final hidden state of the frozen LLM at the answer position, computed identically for the teacher and the student.

Evaluation benchmarks. We evaluate on Video-MME and MVBench using VLMEvalKit with its benchmark-specific evaluation protocols. The numbers of evaluated questions are 2,700 for Video-MME and 4,000 for MVBench. Unless otherwise stated, we report accuracy as the evaluation metric.

Evaluation preprocessing. Evaluation videos are compressed at their original dataset resolution. We use visual inputs only, without subtitles or audio. For MVBench samples with annotated temporal boundaries, we restrict compression and evaluation to the corresponding annotated segment. Reconstructed RGB inputs subsequently undergo the official Qwen3-VL processor's resizing, as described in Appendix B.4. Within each benchmark, all compared methods use the same evaluation prompts and answer-scoring protocol provided by VLMEvalKit.

MLLM backbones. We evaluate the proposed interface with Qwen3-VL-8B and Qwen3-VL-4B. For both models, the vision backbone and language model remain frozen throughout training. The exact checkpoints are Qwen3-VL-8B-Instruct and Qwen3-VL-4B-Instruct. PreSTA injects its adapted visual tokens at visual block 2; all subsequent vision and language modules use the original pretrained weights without modification.

B.2 CODEC CONFIGURATION

We use the low-delay variant of DCVC-UF. The intra-period is set to -1, corresponding to one initial I-frame followed by predictive coding, and the feature-memory reset interval is 128 frames. With intra-period set to -1, only the initial frame is intra-coded. We select the same eight anchor positions from P-frame indices $1, \ldots, T-1$ for all methods. We evaluate four codec quality settings,

$$q \in \{0, 11, 21, 32\}.$$

Here, q denotes the codec operating-point parameter rather than the achieved bitrate. A separate Stage 0 teacher and Stage 1 adapter are trained for each operating point.

For rate–accuracy evaluation, the bitrate is measured directly from the complete coded bitstream. The reported bits-per-pixel (bpp) includes the initial I-frame and stream overhead and is computed

as

$$\mathrm{bpp} = \frac{\text{total number of coded bits}}{\text{number of frames} \times H \times W}.$$

Thus, the reported rate corresponds to the complete coded sequence rather than only the sampled observations used by the MLLM.

B.3 Temporal Sampling and Visual-Token Budget

All compared methods use the same eight uniformly sampled anchor positions. For the sparse reconstructed-RGB baseline, these eight positions are reconstructed and individually passed to the Qwen3-VL visual front-end.

For PreSTA, each anchor is associated with three preceding predictive P-frame states, yielding four temporally ordered states per group and 32 predictive states in total. The eight causal groups therefore preserve the same eight-group visual-token budget as the sparse RGB baseline. The initial I-frame is excluded from the proposed adapter input.

For 256×256 inputs, each of the eight groups produces 256 tokens at Qwen3-VL visual block 2, yielding 2,048 pre-merger tokens. In the native-720p complexity analysis, the token count scales with spatial resolution, while the number of temporal groups remains eight. These are intermediate vision tokens before spatial merging, not the final visual tokens received by the language model.

For short videos or groups without three valid preceding P-frames, we pad the missing preceding slots by replicating the current anchor frame's features.

B.4 RGB Preprocessing

RGB inputs are processed using the official Qwen3-VL processor with resizing. For collage-based inputs, four temporally ordered frames are arranged in a fixed 2×2 layout before being passed through the frozen vision prefix. The ordering is identical to that used by the temporal folding operation in PreSTA. The codec input resolution used for training is 256×256, while the complexity measurements are additionally reported at native 720p resolution.

B.5 Training Details

Stage 0. Stage 0 trains only the RGB residual fusion module R_θ. The DCVC-UF codec and the complete Qwen3-VL backbone remain frozen. The compact teacher is optimized using the hidden-state alignment loss

$$\mathcal{L}^{(0)} = 1 - \cos\left(\mathbf{h}_f, \mathrm{sg}(\mathbf{h}_{32})\right).$$

We train Stage 0 for 30,000 steps with the AdamW optimizer, a learning rate of 1×10^{-4}, a weight decay of 1×10^{-2}, and a batch size of 1.

Stage 1. After Stage 0, the RGB teacher is frozen. Stage 1 optimizes the codec-domain residual fusion module R_ϕ and the spatial neck $\mathcal{N}_\phi$. Training first uses

$$\mathcal{L}_{\mathrm{MSE}} + \lambda \mathcal{L}_{\mathrm{hid}},$$

where $\lambda = 0.2$, followed by feature-only refinement with $\mathcal{L}_{\mathrm{MSE}}$. The joint-alignment phase is trained for 10,000 steps and the feature-only refinement phase for 50,000 steps.

We use the AdamW optimizer with a learning rate of 2×10^{-4}, a weight decay of 1×10^{-4}, a batch size of 32, a cosine learning-rate schedule decaying to 1×10^{-5}, and 500 warm-up steps. We train on a single NVIDIA RTX 6000 Ada.

The output projection of each residual cross-attention module is zero-initialized. All codec states used for training are precomputed, and the Stage 0 teacher targets are cached during Stage 1.

Number of context frames (original RGB-grid). Table 5 studies the number K of context frames per causal group under the original RGB-grid setting, where source (uncompressed) frames are directly stitched into a spatial grid collage rather than passed through the codec adapter. Video-MME improves from 59.3% at $K = 0$ (anchor only) to 63.3% at $K = 3$, while $K = 5$ degrades

to 62.6%, indicating diminishing returns beyond three preceding frames. We therefore use $K = 3$, i.e., three preceding frames plus the anchor, consistent with the temporal folding in PreSTA.

Table 5: Context frames K per causal group (token budget fixed at eight groups). Video-MME, Original RGB-Grid.

K	Acc. (%)	Layout
0 (anchor only)	59.3%	1×1
1	62.1%	1×2
3 (**PreSTA** set)	**63.3%**	2×2
5	62.6%	2×3

Injection layer. Table 6 ablates where the adapted tokens enter the frozen vision tower. Injecting later bypasses more early vision blocks and therefore saves more parameters and computation (14.8M/478G at block 2 up to 75.8M/3105G at block 6), but the lightweight adapter cannot recover the deeper visual representations, so Video-MME drops sharply from 64.3% to 56.1% and 50.8%. We inject at block 2, which retains the best accuracy while already saving 14.8M parameters and 478G MACs over the reconstructed-RGB route.

Table 6: Injection layer in the frozen vision tower. Video-MME, $q = 32$.

Inject at	Acc. (%)	Net parameter reduction (M)	Net MAC reduction (G, 720p)
Block 2 (**PreSTA**)	**64.3%**	14.8	478
Block 4	56.1%	45.3	1791
Block 6	50.8%	75.8	3105

B.6 Front-End Complexity Breakdown

All four paths are profiled from available DCVC-UF decoder features to the input of Qwen visual block 2. The RGB routes include reconstruction of their sampled frames, while PreSTA uses predictive features directly. Shared predictive decoding and state updates, as well as block 2 and all subsequent modules, are excluded.

Params count unique executed weights, including frozen Qwen weights. In Reconstructed RGB-Fusion, the shared Qwen front end processes 40 images but its weights are counted once. The RGB routes also use the 2.654M-parameter Qwen positional table, which PreSTA does not invoke. MACs count convolution, linear, and attention query-key/value products at 256×256 and native 720p resolution. Pooling, interpolation, normalization, positional operations, and tensor rearrangement have zero MACs under this convention, although their latency is measured.

Per-module latencies marked "isolated" include individual launch and synchronization overhead and are not additive; the bold complete-path time is the measured latency of the full visual front end. Runtime is measured on one NVIDIA RTX 6000 Ada with 5 warm-up and 20 timed runs, and we report the mean. All totals are computed from unrounded values.

Table 7: PreSTA from 32 codec features $\mathbf{F}_t$ to eight visual groups entering Qwen visual block 2.

Module	Params (M)	GMAC_{256}	GMAC_{720}	ms_{720} (isolated)
Temporal fold / average pooling	0	0	0	0.865
Spatial stem	1.181	2.416	34.729	0.797
Spatial Transformer block	15.940	33.823	718.453	13.878
Output projection and token reorder	1.328	2.718	37.371	1.276
Residual token cross-attention	2.371	5.369	180.397	3.386
Complete path	**20.820**	**44.325**	**970.95**	**17.983**

Table 8: Reconstructed RGB-8: eight anchor positions are reconstructed and encoded by the Qwen visual front end. The learned Qwen positional table is counted once as active frozen parameters.

Module	Params (M)	GMAC_{256}	GMAC_{720}	ms_{720} (isolated)
DCVC-UF reconstruction head (8 frames)	0.745	6.068	85.333	8.648
Normalization and patchify	0	0	0	1.148
Qwen patch embed (8 images)	1.771	3.624	49.828	9.214
Qwen learned pos. table and add	2.654	0	0	0.257
Qwen visual block 0 (8 images)	15.240	32.388	657.111	15.046
Qwen visual block 1 (8 images)	15.240	32.388	657.111	15.064
Complete path	**35.649**	**74.469**	**1449.383**	**48.328**

Table 9: Reconstructed RGB-Grid: all 32 positions are reconstructed, then every four frames form one collage. Only the eight collages enter the Qwen visual front end.

Module	Params (M)	GMACs_{256}	GMACs_{720p}	ms_{720} (isolated)
DCVC-UF reconstruction head (32 frames)	0.745	24.272	341.331	41.411
2×2 collage construction	0	0	0	0.460
Normalization and patchify	0	0	0	1.148
Qwen patch embed (8 collages)	1.771	3.624	49.828	9.214
Qwen learned pos. table and add	2.654	0	0	0.257
Qwen visual block 0 (8 collages)	15.240	32.388	657.111	15.046
Qwen visual block 1 (8 collages)	15.240	32.388	657.111	15.064
Complete path	**35.649**	**92.673**	**1705.382**	**80.534**

Table 10: Reconstructed RGB-Fusion: eight collages form the queries and 32 individually reconstructed frames provide the keys and values for residual cross-attention fusion. The Qwen patch embedder and visual blocks are shared between the two branches; "shared" marks their second use, not extra weights.

Module	Params (M)	GMACs_{256}	GMACs_{720p}	ms_{720} (isolated)
DCVC-UF reconstruction head (32 frames)	0.745	24.272	341.331	41.411
2×2 collage construction	0	0	0	0.460
Normalization and patchify (8)	0	0	0	1.148
Qwen patch embed (8 collages)	1.771	3.624	49.828	9.214
Qwen learned pos. table and add (8)	2.654	0	0	0.257
Qwen visual block 0 (8 collages)	15.240	32.388	657.111	15.046
Qwen visual block 1 (8 collages)	15.240	32.388	657.111	15.064
Normalization and patchify (32)	0	0	0	4.490
Qwen patch embed (32 frames)	shared	14.496	199.313	39.892
Qwen learned pos. add (32)	shared	0	0	1.001
Qwen visual block 0 (32 frames)	shared	129.554	2628.444	64.225
Qwen visual block 1 (32 frames)	shared	129.554	2628.444	64.971
Residual cross-attention fusion	5.322	32.011	1287.232	20.684
Total unique / executed	**40.971**	**398.287**	**8448.816**	**273.871**