PECV-RT: REAL-TIME PERCEPTUAL ENHANCEMENT OF COMPRESSED VIDEO

Anonymous authors
Paper under double-blind review

ABSTRACT

Perceptual Enhancement of Compressed Video (PECV) methods have achieved substantial success. However, most methods suffer from high computational complexity, thereby limiting their deployment in real-time scenarios. To address this issue, we present PECV-RT, a lightweight framework that focuses on three bottlenecks in prior work: suboptimal computation allocation across resolutions, costly global spatial context modeling, and redundant computation across overlapping temporal windows. Resolution-aware computation allocation optimizes the distribution of computation by shifting intensive global context modeling to low resolutions, while making local processing lightweight at high resolutions. Linear spatial attention reduces computational complexity from quadratic to linear in the number of spatial tokens. Cross-frame computation sharing limits repeated feature extraction and reuses temporal-attention keys and values across output frames. Experiments show that PECV-RT achieves state-of-the-art LPIPS performance with negligible PSNR degradation. PECV-RT enhances 1080p compressed video at 102 FPS on a consumer-grade GPU, delivering 15.7× the throughput of the fastest evaluated baseline. To our knowledge, it is the first method to perform perceptual enhancement of 1080p compressed video at over 60 FPS on such hardware.

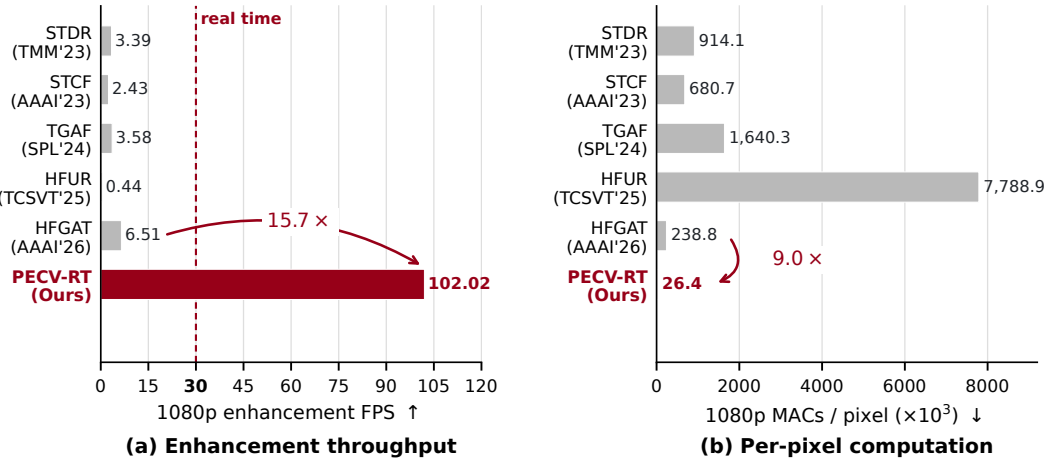


Figure 1: **Real-time compressed video enhancement at low computational cost.** (a) 1080p enhancement throughput on an RTX 4090D. (b) MACs per pixel at 1080p. Compared with HFGAT, PECV-RT achieves 15.7× the throughput with 9.0× fewer MACs per pixel.

1 INTRODUCTION

Lossy video compression reduces transmission and storage costs but typically degrades fine textures and introduces blocking and ringing artifacts (Deng et al., 2020). Decoder-side enhancement aims to mitigate these degradations, with perceptual enhancement focusing on visually meaningful detail rather than pixel fidelity alone (Blau & Michaeli, 2018; Peng et al., 2026). Existing enhance-

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.

ment methods have substantially improved visual quality by exploiting spatial and temporal information (Deng et al., 2020; Guan et al., 2021; Zhao et al., 2021; Luo et al., 2023; Zhang et al., 2023; Zhu et al., 2024; Dong et al., 2024; Zhang et al., 2025; Peng et al., 2026).

However, recent perceptual enhancement methods still face challenges in real-time playback, especially at high resolutions, where the increased pixel-processing workload makes it difficult to maintain the required frame rate. Although HFGAT (Peng et al., 2026) has made progress in reducing model size and accelerating inference, its throughput remains insufficient for real-time 1080p enhancement on a consumer-grade GPU. Our same-machine measurements at 1080p show that the five compared methods achieve only 0.44–6.51 FPS (Figure 1(a)). Even the fastest falls far below a 30-FPS real-time playback target. This substantial gap underscores the urgent need to accelerate perceptual enhancement while preserving its visual quality benefits.

With this motivation, PECV-RT is designed around three questions for reducing computational cost: *where to allocate computation*, *how to model spatial context efficiently*, and *how to share computation across frames*.

Resolution-aware computation allocation. Local detail reconstruction and context modeling have different resolution requirements. Reconstructing textures and edges benefits from higher-resolution features, while broader spatial and temporal context can be modeled at lower resolution. PECV-RT therefore performs global spatial and temporal context modeling in a low-resolution branch while retaining lightweight local processing at higher resolutions. This design reduces costly computation at high resolutions while preserving fine-scale features for reconstructing details.

Linear spatial attention. With context modeling assigned to a low-resolution branch, the next question is how to compute global spatial interactions efficiently. Standard global self-attention computes interactions between all pairs of tokens, resulting in quadratic complexity with respect to the spatial token count (Vaswani et al., 2017). Following SANA (Xie et al., 2025), PECV-RT uses linear attention to capture global spatial context without constructing a full pairwise attention matrix. Its computational complexity scales linearly with the number of spatial tokens.

Cross-frame computation sharing. While resolution-aware computation allocation and linear spatial attention reduce the cost of context modeling, cross-frame computation sharing reduces redundancy between temporal windows. Overlapping temporal windows share many input frames, but existing methods process each window independently, resulting in repeated computation over the shared frames. Inspired by chunk-based processing in DCVC-UF (Li et al., 2026), PECV-RT processes consecutive frames together, reducing repeated feature extraction and sharing temporal-attention keys and values across output frames.

Experiments on HEVC- and VVC-compressed video demonstrate that PECV-RT achieves state-of-the-art LPIPS performance with slight PSNR degradation. Our method also achieves substantially higher throughput and lower computational cost than the compared methods (Figure 1).

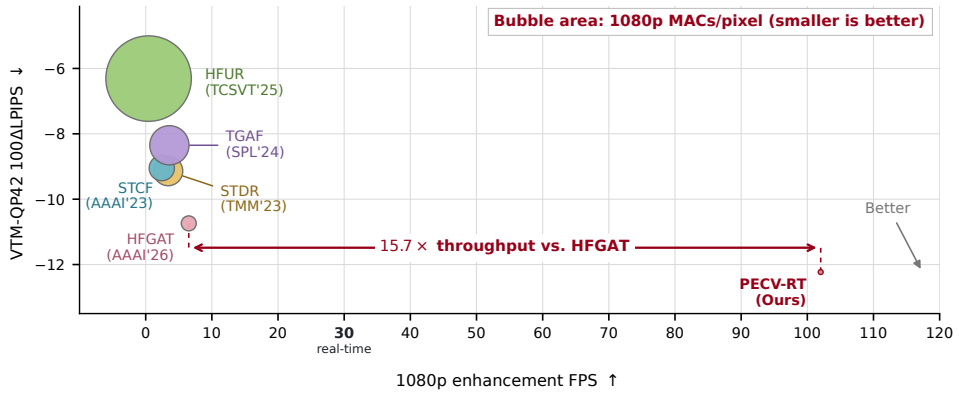


Figure 2: **Perceptual quality–efficiency trade-off on VTM-QP42.** 1080p enhancement FPS is measured on an RTX 4090D. Lower 100ΔLPIPS indicates a larger improvement over decoded video. Bubble area is proportional to MACs/pixel, and smaller is better.

2 METHOD

Figure 3 presents the overall architecture of PECV-RT. The network extracts multiscale features from decoded frames and jointly enhances a chunk of consecutive frames. Lightweight local processing preserves higher-resolution detail for reconstruction, while low-resolution context modeling captures global spatial dependencies and enables temporal interaction. Within each chunk, features are extracted once per target frame and used for joint context modeling across frames. The resulting context features are fused with each frame's local features and progressively upsampled to predict residual corrections, which are added to the corresponding decoded frames.

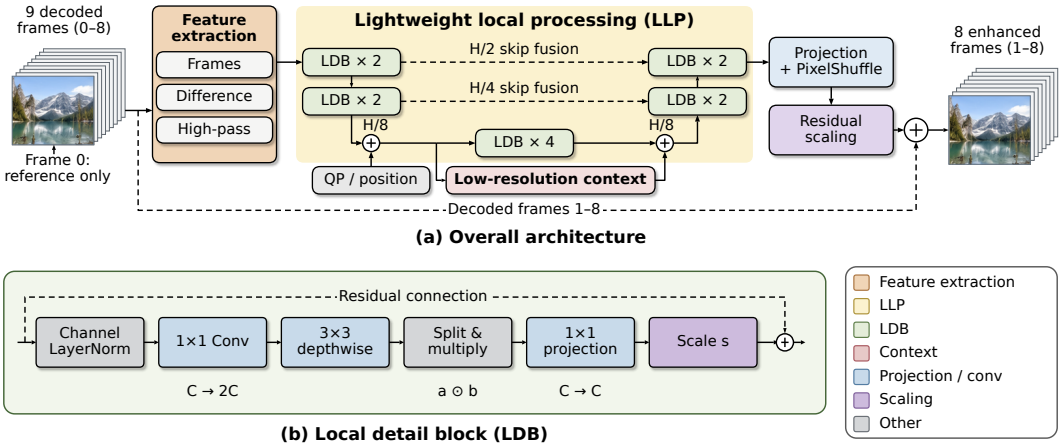


Figure 3: **Overview of PECV-RT.** (a) The overall enhancement network. Feature extraction follows Eq. 1. QP and output-position embeddings condition the $H/8$ features. The predicted RGB residual is scaled and added to the decoded frames to form the enhanced output. (b) The local detail block (LDB) preserves input dimensions. C denotes channels, a, b are equal channel-wise splits, $\odot$ denotes elementwise multiplication, and s a learned per-channel scale.

2.1 RESOLUTION-AWARE COMPUTATION ALLOCATION

Computation allocation philosophy. Our goal is to determine how far context resolution can be reduced without substantially compromising enhancement quality. Figure 4 compares different context resolutions with other modules unchanged. Moving context modeling from $H/4$ to $H/16$ reduces its share of total network MACs from 63.4% to 11.1%, and its share of runtime from 46.0% to 4.6%. Compared with $H/4$ and $H/8$, $H/16$ yields only a small degradation in LPIPS (at most 0.00145), while slightly improving PSNR (Table 1). This limited quality benefit motivates allocating context computation to $H/16$, while retaining higher-resolution local features for detail reconstruction rather than routing all features through the low-resolution bottleneck (Zamir et al., 2021; Wang et al., 2022). Appendix B provides a module-wise computation breakdown.

Feature extraction. Given 9 decoded RGB frames $\tilde{\boldsymbol{x}}_0, \ldots, \tilde{\boldsymbol{x}}_8$, PECV-RT enhances frames 1–8 using RGB, inter-frame difference, and high-frequency information. For each target frame, the initial feature is

$$\boldsymbol{f}_p = E_s(\tilde{\boldsymbol{x}}_p) + E_d(\tilde{\boldsymbol{x}}_p - \tilde{\boldsymbol{x}}_{p-1}) + E_h(\tilde{\boldsymbol{x}}_p - A(\tilde{\boldsymbol{x}}_p)), \quad p = 1, \ldots, 8, \tag{1}$$

where A is a 3×3 average filter and E_s, E_d, E_h are separate learned 2×2 stride-2 convolutional projections. The resulting feature map has spatial resolution $H/2 \times W/2$.

Lightweight local processing. Local detail blocks (Figure 3(b)) use pointwise channel mixing and gated depthwise convolution to process each frame independently (Chen et al., 2022). Strided convolutions reduce the resolution from $H/2$ to $H/8$, retaining $\boldsymbol{f}_p^{(2)}$ and $\boldsymbol{f}_p^{(4)}$ at $H/2$ and $H/4$ as skip features. Learned QP and output-position embeddings condition the $H/8$ features.

Low-resolution context modeling. The context branch (Figure 5(a)) downsamples to $H/16$ for spatial and temporal interaction, then upsamples to $H/8$ for addition to the local features. Section 2.2 details these operations.

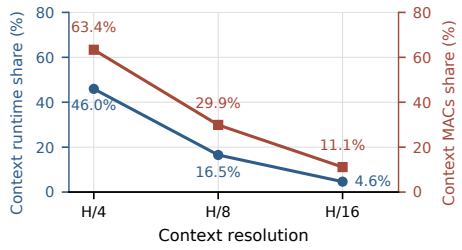


Figure 4: **Effect of context resolution on computational cost.** Context runtime (blue) and MACs (red) as shares of whole-network totals, at 1080p.

Table 1: **Context-resolution comparison on HM-QP42.** PSNR and LPIPS differences relative to $H/16$, with negative values indicating a decrease.

Resolution	ΔPSNR ↑ (dB)	ΔLPIPS ↓ ($\times 10^{-2}$)
$H/16$	0	0
$H/8$	−0.02202	−0.0733
$H/4$	−0.05261	−0.1449

2.2 Linear Spatial Attention

Reducing the context resolution lowers the spatial token count, but standard global attention still scales quadratically with this count. Our goal is to retain global spatial interaction while avoiding its quadratic computational cost. Following SANA (Xie et al., 2025), the low-resolution context branch uses ReLU-based linear attention at $H/16 \times W/16$ to aggregate global context. Mix-FFN (Xie et al., 2021) uses a 3×3 depthwise convolution to mix features from neighboring spatial positions.

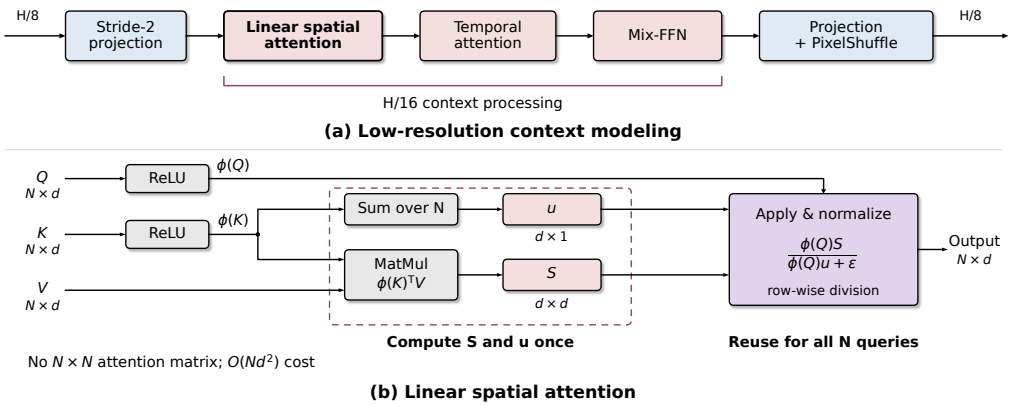


Figure 5: **Low-resolution context modeling and linear spatial attention.** (a) The context branch. (b) Attention within one head of one frame. The shared terms S and $\boldsymbol{u}$ in Eqs. 2–3 are reused by all spatial queries. Here, N is the spatial token count, $d = d_{\text{head}}$, and $\phi = \text{ReLU}$. Division is row-wise.

Compact spatial interaction. The low-resolution features are projected into queries, keys, and values. For each attention head in each frame, we compute the key–value aggregation S and key sum $\boldsymbol{u}$ once, then reuse them for all spatial queries. Let $N = (H/16)(W/16)$ denote the number of spatial positions in each frame. The shared terms and the normalized attention output are

$$S = \sum_{j=1}^{N} \phi(\boldsymbol{k}_j)^\top \boldsymbol{v}_j, \tag{2}$$

$$\boldsymbol{u} = \sum_{j=1}^{N} \phi(\boldsymbol{k}_j)^\top, \tag{3}$$

$$\text{LA}(Q, K, V)_i = \frac{\phi(\boldsymbol{q}_i) S}{\phi(\boldsymbol{q}_i)\boldsymbol{u} + \epsilon}, \tag{4}$$

where $Q, K, V \in \mathbb{R}^{N \times d_{\text{head}}}$ are the query, key, and value matrices, with $d_{\text{head}} = 32$. The row vectors $\boldsymbol{q}_i, \boldsymbol{k}_j, \boldsymbol{v}_j \in \mathbb{R}^{1 \times d_{\text{head}}}$ are the corresponding rows of Q, K, V, respectively. Here, $\phi = \text{ReLU}$ and $\epsilon > 0$ is a numerical-stability constant.

The shared terms S and $\boldsymbol{u}$ have dimensions $d_{\text{head}} \times d_{\text{head}}$ and d_{head} respectively, independent of spatial resolution. Reusing them across queries provides global spatial context without constructing an $N \times N$ attention matrix. Consequently, the attention computation requires $O(N d_{\text{head}}^2)$ operations rather than $O(N^2 d_{\text{head}})$, making its complexity linear in N at a fixed head width (Katharopoulos et al., 2020).

Temporal interaction and fusion. After spatial attention, bidirectional temporal attention exchanges information among the 8 target frames at each spatial location. This requires only 8×8 attention matrices, rather than attention over all space–time token pairs. A spatial–temporal Mix-FFN then refines these features through pointwise channel expansion, a 3×3 depthwise spatial convolution, and a residual depthwise temporal convolution with kernel size 3. The refined features are split into two equal channel groups, a and b, and combined through SiLU-based gating, $a \odot \text{SiLU}(b)$. A pointwise projection then restores the original channel dimension. To fuse context with local detail, the output is projected and upsampled to $H/8$ using pixel shuffle (Shi et al., 2016), then added to the output of the 4 local detail blocks.

2.3 Cross-frame computation sharing

In existing single-output methods, feature extraction and context processing are repeated across overlapping temporal windows (Figure 6(a)). We instead extract features once for each of the eight target frames and jointly process them within a chunk (Figure 6(b)). At each spatial location, temporal attention computes keys and values once and reuses them across the queries of all 8 output positions. For example, frame 4's key and value at position j are reused by the queries of all eight target frames (Figure 6(b)). Our model performs joint temporal processing, rather than merely batching independent single-frame networks.

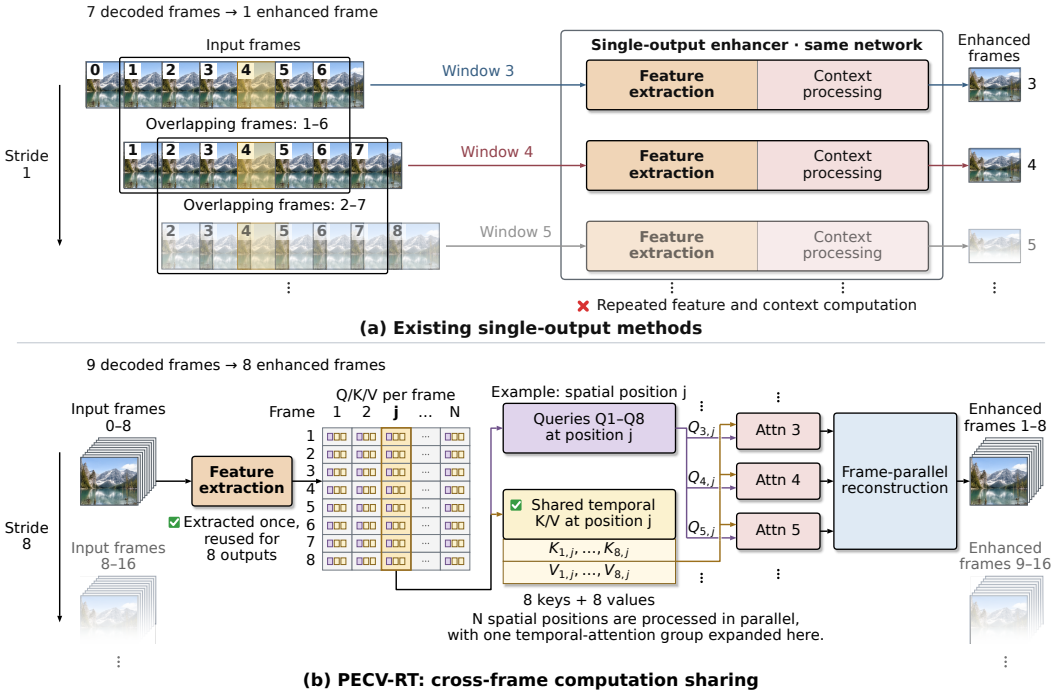


Figure 6: (a) Single-output methods repeatedly process overlapping windows. (b) Joint eight-frame enhancement. Matrix rows denote frames and columns denote spatial positions. Each frame provides N queries, keys, and values per attention head. At the highlighted position j, all eight queries share the same eight keys and values; $Q_{3,j}$–$Q_{5,j}$ are shown as examples. Faded stacks in (b) indicate the next chunk.

3 EXPERIMENTS

3.1 EXPERIMENTAL SETUP

Datasets and coding settings. The training and test sets follow HFGAT (Peng et al., 2026). Training videos are sourced from Xiph, VQEG, and JCT-VC (Xiph.Org, n.d.; VQEG, n.d.; Bossen, 2013), and evaluation uses 18 JCT-VC Class A–E sequences. We first extract 33-frame clips from the training videos, then randomly sample 9-frame windows from these clips, using the last 8 frames as enhancement targets. Videos are encoded with HM-16.5 and VTM-17.0 in low-delay P mode at QP37 and QP42, with IntraPeriod $= -1$ (Sullivan et al., 2012; Bross et al., 2021).

Training details. Training uses random 256×256 crops and a global batch size of 8 for 16k iterations. AdamW (Loshchilov & Hutter, 2019) uses $\beta_1 = 0.9$, $\beta_2 = 0.999$, and weight decay 10^{-4}. Initial learning rates are 2×10^{-4} for the context backbone, 7.5×10^{-5} for adaptation, conditioning, and reconstruction layers, and 2.5×10^{-5} for residual scaling, with cosine decay (Loshchilov & Hutter, 2017) to 20% of their initial values. Fine-tuning lasts 4k iterations, starting at 5% of the original learning rates. Each codec/QP setting is trained separately. The loss combines mean squared error (MSE) for pixel fidelity with learned perceptual image patch similarity (LPIPS) (Zhang et al., 2018) for deep-feature similarity (Johnson et al., 2016; Ledig et al., 2017). For enhanced outputs $\hat{\boldsymbol{x}}$ and ground-truth source frames $\boldsymbol{x}$, the objective is

$$\mathcal{L} = \mathrm{MSE}(\hat{\boldsymbol{x}}, \boldsymbol{x}) + \lambda\, \mathrm{LPIPS}(\hat{\boldsymbol{x}}, \boldsymbol{x}), \tag{5}$$

Here, LPIPS uses an AlexNet backbone, and $\lambda = 0.004$ controls the weight of the perceptual term.

Evaluation protocol. PSNR and LPIPS are evaluated in RGB over all P-frames and averaged across sequences. Tables report changes relative to codec reconstruction. Appendix A details evaluation lengths, padding, and boundary handling. Baseline quality results are taken from HFGAT's Tables 1–2 (Peng et al., 2026), whereas throughput is measured independently over repeated runs on the same RTX 4090D, with no competing workloads on the machine.

3.2 QUANTITATIVE RESULTS

Tables 2 and 3 report per-sequence results at QP42 and average results at QP37 and QP42 for VTM and HM, respectively.

Table 2: QP42 per-sequence and QP37 average enhancement over the VTM-17.0 reconstruction on 18 full-length JCT-VC sequences. PSNR denotes ΔPSNR (dB), and LPIPS denotes 100ΔLPIPS; higher PSNR and lower LPIPS are better. **Red bold**, blue underlined, and green values indicate the best, second-best, and third-best results per row and metric, respectively.

QP	Class	Sequence	STDR (TMM'23)		STCF (AAAI'23)		TGAF (SPL'24)		HFUR (TCSVT'25)		HFGAT (AAAI'26)		PECV-RT (Ours)	
			PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
42	A	Traffic	−0.38	−9.11	−0.34	−10.42	−0.20	−8.44	−0.28	−8.13	**+0.01**	**−12.05**	−0.07	−11.14
		PeopleOnStreet	+0.05	−6.22	+0.01	−6.66	+0.16	−6.99	+0.11	−3.18	**+0.33**	−7.54	−0.05	**−7.61**
	B	Kimono	−0.22	−10.60	−0.17	−9.42	−0.07	−8.96	−0.15	−8.92	**+0.08**	−13.44	−0.04	**−18.12**
		ParkScene	−0.28	−20.26	−0.14	**−20.47**	−0.10	−16.92	−0.14	−13.22	**+0.05**	−19.14	−0.08	−20.37
		Cactus	−0.40	−10.46	−0.24	−11.54	−0.19	−10.10	−0.32	−5.46	**+0.04**	**−12.49**	−0.08	−11.94
		BQTerrace	−0.35	−11.19	−0.36	−10.11	−0.17	−8.94	−0.23	−5.65	−0.09	−11.46	**−0.08**	**−17.25**
		BasketballDrive	−0.43	−10.02	−0.28	−9.60	−0.20	−10.66	−0.22	−5.23	**+0.02**	−13.78	−0.12	**−18.16**
	C	RaceHorses	−0.13	−8.44	−0.07	−8.61	−0.11	−8.58	−0.06	−5.52	**+0.02**	−10.19	−0.08	**−13.73**
		BQMall	−0.08	−8.63	−0.03	−8.48	+0.02	−7.56	+0.08	−8.02	**+0.19**	**−10.25**	−0.04	−9.60
		PartyScene	−0.28	−6.90	−0.32	−6.64	−0.24	−5.64	−0.13	−6.26	**+0.12**	**−8.99**	−0.03	−8.91
		BasketballDrill	−0.51	−8.18	−0.38	−6.71	−0.26	−7.88	−0.35	−2.55	**+0.01**	−10.26	−0.09	**−13.81**
	D	RaceHorses	+0.02	−9.63	+0.08	−7.67	+0.02	−9.46	+0.14	−7.82	**+0.19**	−10.69	−0.07	**−13.06**
		BQSquare	−0.26	−6.43	−0.22	−7.34	−0.17	−5.13	−0.06	−5.01	**+0.05**	−7.42	−0.04	**−7.72**
		BlowingBubbles	−0.16	−9.87	−0.19	−10.63	−0.11	−8.33	−0.02	−9.76	**+0.22**	**−12.56**	+0.01	−9.49
		BasketballPass	−0.02	−7.63	+0.09	−8.88	+0.05	−6.80	+0.14	−7.14	**+0.31**	**−9.80**	−0.07	−9.23
	E	FourPeople	−0.49	−4.87	−0.47	−3.77	−0.30	−4.63	−0.46	−3.57	**+0.08**	**−7.21**	−0.11	−6.53
		Johnny	−1.08	−8.41	−0.60	−8.26	−0.47	−8.07	−0.81	−4.14	**−0.06**	−7.79	−0.14	**−12.72**
		KristenAndSara	−0.77	−7.61	−0.50	−7.64	−0.44	−7.29	−0.77	−3.92	**+0.04**	−8.20	−0.10	**−10.82**
		Average	−0.32	−9.14	−0.23	−9.05	−0.15	−8.35	−0.20	−6.31	**+0.09**	−10.74	−0.07	**−12.23**
37		**Average**	−0.32	−8.24	−0.18	−8.31	−0.13	−7.31	−0.26	−5.82	**+0.07**	−8.88	−0.11	**−10.99**

On VTM, PECV-RT reduces average LPIPS by 0.1099 and 0.1223 at QP37 and QP42, and the corresponding HM reductions are 0.0725 and 0.0869. PECV-RT achieves state-of-the-art average LPIPS performance across all 4 evaluated settings. For HM, perceptual gains are accompanied by slight improvements in average PSNR, whereas VTM exhibits a small PSNR decrease. Overall, average PSNR changes range from -0.11 to $+0.04$ dB relative to codec reconstruction. Together with the efficiency results in the next section, these comparisons show that PECV-RT combines competitive enhancement quality with substantially lower computational cost and higher throughput.

Table 3: QP42 per-sequence and QP37 average enhancement over the HM-16.5 reconstruction. The metric definitions, protocol, and ranking notation follow Table 2.

QP	Class	Sequence	Dong et al. (TCSVT'24)		STCF (AAAI'23)		TGAF (SPL'24)		HFUR (TCSVT'25)		HFGAT (AAAI'26)		PECV-RT (Ours)	
			PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓	PSNR ↑	LPIPS ↓
42	A	Traffic	–	−5.50	−0.20	−8.09	−0.12	−5.47	−0.23	−4.47	**+0.18**	−8.56	+0.10	**−8.88**
		PeopleOnStreet	–	−1.40	+0.30	−2.12	+0.42	−3.07	+0.43	+1.54	**+0.60**	−5.20	+0.08	**−5.90**
	B	Kimono	–	−2.90	−0.28	−1.75	−0.19	−0.46	−0.13	−1.41	−0.03	−4.31	**+0.06**	**−10.19**
		ParkScene	–	−11.90	−0.13	**−14.68**	−0.12	−9.75	−0.12	−8.00	+0.04	−13.20	**+0.08**	−11.42
		Cactus	–	**−10.70**	−0.12	−9.14	−0.05	−7.57	−0.14	−3.96	**+0.21**	−10.17	+0.01	−7.92
		BQTerrace	–	−5.70	−0.02	−6.72	+0.08	−5.89	+0.04	−3.90	**+0.28**	−8.53	+0.03	**−11.45**
		BasketballDrive	–	−10.20	+0.02	−8.69	+0.04	−9.18	+0.14	−6.13	**+0.28**	−11.82	−0.03	**−12.95**
	C	RaceHorses	–	−7.20	+0.09	−6.36	+0.01	−4.98	+0.08	−3.01	**+0.20**	−5.95	−0.01	**−10.20**
		BQMall	–	−7.20	+0.16	−6.63	+0.17	−6.38	+0.20	−7.00	**+0.40**	**−8.09**	+0.10	−6.10
		PartyScene	–	−8.50	−0.06	−7.72	−0.07	−6.10	−0.03	−6.41	**+0.30**	**−8.59**	+0.07	−6.91
		BasketballDrill	–	−5.60	−0.01	−6.75	−0.02	−7.15	+0.02	−2.24	**+0.36**	−7.80	+0.04	**−10.00**
	D	RaceHorses	–	−5.40	+0.20	−6.59	+0.13	−6.54	+0.31	−4.56	**+0.33**	−6.53	+0.00	**−8.26**
		BQSquare	–	−4.40	+0.00	−5.46	+0.08	−4.83	+0.16	−4.33	**+0.25**	**−7.59**	−0.01	−4.60
		BlowingBubbles	–	−9.40	−0.03	−9.85	−0.02	−7.90	+0.06	−8.42	**+0.33**	**−10.73**	+0.08	−7.18
		BasketballPass	–	−6.00	+0.21	**−7.38**	+0.19	−4.83	+0.34	−5.74	**+0.44**	−6.93	+0.06	−6.64
	E	FourPeople	–	−3.80	−0.18	−3.35	−0.17	−3.38	−0.37	−2.37	**+0.33**	**−6.09**	+0.07	−5.66
		Johnny	–	−6.20	−0.35	−8.54	−0.27	−7.67	−0.79	−3.37	**+0.18**	−8.79	−0.01	**−13.43**
		KristenAndSara	–	−4.40	−0.11	−6.42	−0.19	−4.93	−0.63	−1.89	**+0.37**	−7.30	+0.07	**−8.76**
		Average	–	−6.50	−0.03	−7.01	+0.00	−5.89	−0.03	−4.20	**+0.28**	−8.12	+0.04	**−8.69**
37		**Average**	–	−6.40	+0.06	−5.75	−0.02	−5.28	+0.13	−3.87	**+0.24**	−7.24	+0.01	**−7.25**

3.3 Efficiency Comparison

Table 4 reports complexity, memory, and throughput. PECV-RT achieves 238.9 FPS at 720p and 102 FPS at 1080p, delivering 15.7× the throughput of HFGAT. At 1080p, PECV-RT requires 26,401 MACs per pixel, 9.0× fewer than HFGAT. PECV-RT also has the fewest parameters among the 6 implementations. Its 4,377 MiB peak allocation is much lower than HFGAT's 8,387 MiB, although STCF and TGAF have lower peak GPU memory usage.

Table 4: Complexity and throughput on one RTX 4090D. GFLOPs/frame, MACs/pixel, and peak VRAM (MiB) are measured at 1080p. The 4 rightmost columns report enhanced output FPS. Peak VRAM reports the peak GPU memory allocated by PyTorch. Measurement details are provided in Appendix B.

Method	Params ↓	GFLOPs/frame ↓	MACs/pixel ↓	Peak VRAM ↓	240p ↑	480p ↑	720p ↑	1080p ↑
STDR (TMM'23)	1.475M	3791.14	914,144	12,622	84.53	18.63	7.62	3.39
STCF (AAAI'23)	2.080M	2823.06	680,715	**2,726**	26.60	12.89	5.17	2.43
TGAF (SPL'24)	1.785M	6802.74	1,640,320	3,564	90.04	18.82	7.93	3.58
HFUR (TCSVT'25)	6.525M	32302.01	7,788,873	18,371	10.01	2.53	1.02	0.44
HFGAT (AAAI'26)	1.702M	990.30	238,787	8,387	56.00	40.65	15.88	6.51
PECV-RT (Ours)	**1.382M**	**109.49**	**26,401**	4,377	**3801.14**	**588.33**	**238.92**	**102.02**

Figure 7 provides an overview of enhancement quality and computational efficiency across methods. PECV-RT demonstrates balanced performance across the evaluated metrics, with advantages across multiple dimensions and best results on most metrics.

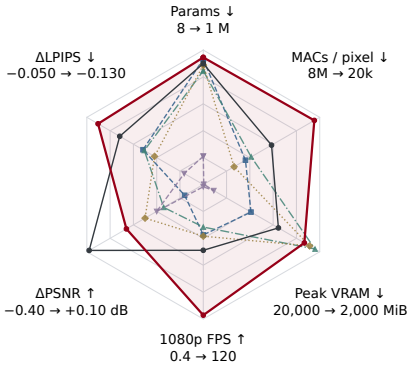


Figure 7: **Quality and efficiency across 6 metrics.** Quality: VTM-QP42; efficiency: 1080p. Farther outward is better. Axis labels give center-to-boundary ranges. MACs/pixel and FPS use logarithmic normalization, and the remaining metrics use linear normalization. Raw values appear in Tables 2 and 4.

3.4 Qualitative Results

Figure 8 compares locally evaluated outputs on Johnny under HM-QP42 and BasketballDrive under VTM-QP37. In Johnny, PECV-RT produces more continuous jacket pinstripes, whereas the baselines leave smooth regions or fragmented patterns. In BasketballDrive, it produces finer fabric texture with fewer artifacts, whereas HFGAT introduces dark, crease-like lines on the light-colored stripe that are absent from the GT. PECV-RT also achieves the lowest LPIPS among the compared methods in both examples, consistent with the visual comparisons.

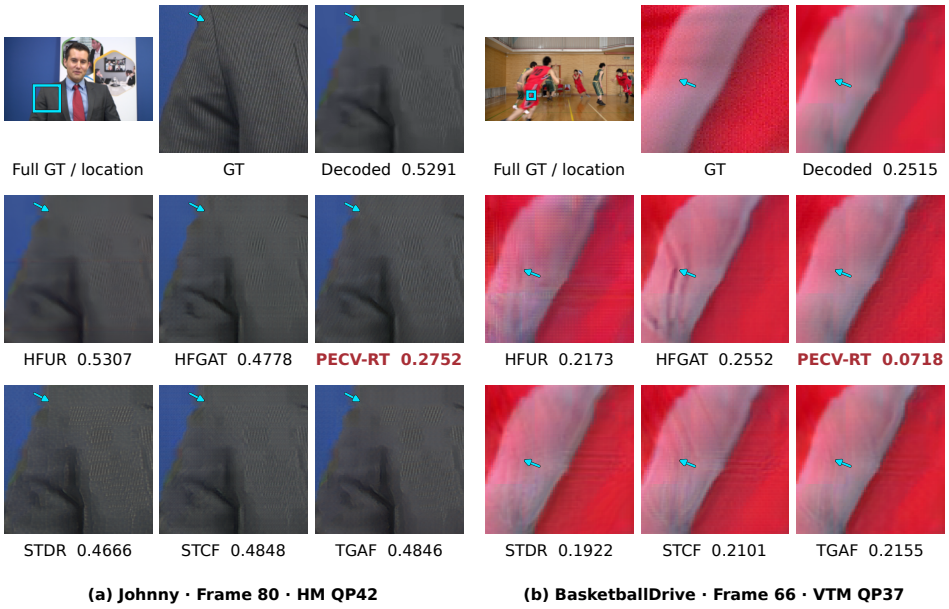


Figure 8: **Visual comparisons on decoded video.** Cyan boxes mark the displayed regions; matching arrows indicate corresponding details across methods. Values indicate LPIPS↓ against GT.

3.5 Ablation Study

Resolution-aware computation allocation. Moving context modeling from $H/16$ to $H/8$ incurs higher computational cost (33,468 vs. 26,401 MACs/pixel) and lower throughput (82.31 vs. 102.02 FPS), without improving LPIPS or PSNR. The compact context grid at $H/16$ already captures sufficient information for enhancement, leaving limited benefit from additional spatial tokens.

Linear spatial attention. Global softmax attention provides no improvement in average LPIPS or PSNR over linear attention, despite its higher computational cost (38,681 vs. 26,401 MACs/pixel) and slightly lower throughput (99.67 vs. 102.02 FPS). Within our architecture, linear attention works

more effectively with local feature processing than global softmax attention, achieving better enhancement quality at lower computational cost.

Table 5: VTM-QP42 ablations relative to the full model. FPS and MACs/pixel are measured at 1080p. $0^{\dagger}$ indicates that single-output inference produces the same enhanced frames as the full model, with only cross-frame computation sharing disabled.

Configuration	ΔPSNR (dB) ↑	100ΔLPIPS ↓	FPS ↑	MACs/pixel ↓	Params (M)
Full model	0	0	102.02	26,401	1.382
Context at $H/8$	-0.0165	$+0.0109$	82.31	33,468	1.271
Softmax attention	-0.0044	$+0.2788$	99.67	38,681	1.382
Single-output inference	$0^{\dagger}$	$0^{\dagger}$	25.07	105,275	1.382

Cross-frame computation sharing. Single-output inference produces the same enhanced frames as the full model, but generates only one frame per forward pass. Features and context are recomputed for each output, with no computation reused across output frames. This requires nearly $4\times$ as many MACs/pixel and reduces throughput by 75.4%, highlighting the computational benefit of cross-frame sharing. Appendix C presents additional component ablations.

4 Related Work

Improving enhancement quality. A central strategy for compressed video enhancement is to recover details by exploiting complementary information preserved in neighboring frames (Guan et al., 2021). MFQE (Yang et al., 2018) follows this strategy by using motion compensation to transfer information from neighboring high-quality frames to lower-quality targets. However, compression artifacts can make optical-flow estimation unreliable, motivating STDF (Deng et al., 2020) to jointly align and fuse neighboring information through deformable sampling. Beyond alignment, subsequent methods seek to exploit a broader range of supporting information. RFDA (Zhao et al., 2021) uses recursive fusion to incorporate longer-range temporal information, while STCF (Zhang et al., 2023) combines motion compensation with shifted-window attention to capture complementary context. While the above methods focus on exploiting neighboring frames, frequency-aware methods address compression-induced detail loss by modeling information in different frequency bands. HFUR (Zhang et al., 2025) uses DCT-domain priors and hierarchical refinement to reconstruct frequency details and exchange information across scales. HFGAT (Peng et al., 2026) incorporates frequency guidance into alignment and models both within-band and cross-band interactions, exploiting complementary frequency information while limiting interference between bands.

Improving computational efficiency. One route to efficient restoration is cheaper feature processing. NAFNet (Chen et al., 2022) uses multiplicative gating to simplify restoration blocks, while multi-frame deformable look-up tables (He et al., 2025) replace temporal extraction and multiscale fusion with table lookup. Recent efficient restoration methods explore adaptive convolutions and deployment-friendly architectures. ECMA (Zhou et al., 2024) uses cascaded multiscale adaptive convolutions, PlainUSR (Wang et al., 2024) employs structural reparameterization for faster inference, and ESC (Lee et al., 2025) replaces much of its self-attention with convolutional attention. Beyond operator complexity, learned video codecs address execution overhead and parallelism: DCVC-RT (Jia et al., 2025) uses compact representations to reduce operational overhead, while DCVC-UF (Li et al., 2026) jointly processes frame chunks for parallel reconstruction.

5 Conclusion

PECV-RT improves the trade-off between perceptual quality and computational complexity through resolution-aware computation allocation, linear spatial attention, and cross-frame computation sharing. Evaluations on HEVC- and VVC-compressed video yield the lowest average LPIPS among the compared methods while largely maintaining the PSNR of decoded video. PECV-RT enhances 1080p video at 102 FPS on an RTX 4090D, bringing these perceptual gains to real-time processing on consumer-grade hardware.

AI USE STATEMENT

Generative AI tools assisted language polishing and figure refinement. The authors reviewed and revised the AI-assisted material and take responsibility for the final manuscript, including its claims, citations, and reported results.

REPRODUCIBILITY STATEMENT

Section 2 describes the network architecture, and Section 3.1 specifies the datasets, coding configurations, and training settings. Appendix A details evaluation protocols. Appendix B provides architectural settings, timing procedures, and module-wise computation. Appendix C reports additional component ablations.

REFERENCES

Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/papers/Blau_The_Perception-Distortion_Tradeoff_CVPR_2018_paper.pdf.

Frank Bossen. Common test conditions and software reference configurations. Technical Report JCTVC-L1100, Joint Collaborative Team on Video Coding, Geneva, Switzerland, 2013. URL https://www.itu.int/wftp3/av-arch/jctvc-site/2013_01_L_Geneva/documents/JCTVC-L1100-v1.zip.

Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. URL https://ieeexplore.ieee.org/document/9503377/.

Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pp. 17–33, 2022. URL https://www.ecva.net/papers/eccv_2022/papers_ECCV/papers/136670017.pdf.

Jianing Deng, Li Wang, Shiliang Pu, and Cheng Zhuo. Spatio-temporal deformable convolution for compressed video quality enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 10696–10703, 2020. doi: 10.1609/aaai.v34i07.6697. URL https://ojs.aaai.org/index.php/AAAI/article/view/6697.

Cunhui Dong, Haichuan Ma, Zhuoyuan Li, Li Li, and Dong Liu. Temporal wavelet transform-based low-complexity perceptual quality enhancement of compressed video. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(5):4040–4053, 2024. doi: 10.1109/TCSVT.2023.3316834. URL https://ieeexplore.ieee.org/document/10254565/.

Zhenyu Guan, Qunliang Xing, Mai Xu, Ren Yang, Tie Liu, and Zulin Wang. MFQE 2.0: A new approach for multi-frame quality enhancement on compressed video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(3):949–963, 2021. doi: 10.1109/TPAMI.2019.2944806. URL https://ieeexplore.ieee.org/document/8855019/.

Gang He, Guancheng Quan, Chang Wu, Shihao Wang, Dajiang Zhou, and Yunsong Li. Multi-frame deformable look-up table for compressed video quality enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3392–3400, 2025. doi: 10.1609/aaai.v39i3.32351. URL https://ojs.aaai.org/index.php/AAAI/article/view/32351.

Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12543–12552, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Jia_Towards_Practical_Real-Time_Neural_Video_Compression_CVPR_2025_paper.html.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pp. 694–711, 2016. doi: 10.1007/978-3-319-46475-6_43. URL https://link.springer.com/chapter/10.1007/978-3-319-46475-6_43.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 5156–5165, 2020. URL https://proceedings.mlr.press/v119/katharopoulos20a.html.

Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4681–4690, 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Ledig_Photo-Realistic_Single_Image_CVPR_2017_paper.html.

Dongheon Lee, Seokju Yun, and Youngmin Ro. Emulating self-attention with convolution for efficient image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 24467–24477, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Lee_Emulating_Self-attention_with_Convolution_for_Efficient_Image_Super-Resolution_ICCV_2025_paper.html.

Jiahao Li, Wenxuan Xie, Zhaoyang Jia, Bin Li, Zongyu Guo, Xiaoyi Zhang, and Yan Lu. Ultra-fast neural video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 41311–41321, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Li_Ultra-Fast_Neural_Video_Compression_CVPR_2026_paper.html.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Dengyan Luo, Mao Ye, Shuai Li, Ce Zhu, and Xue Li. Spatio-temporal detail information retrieval for compressed video quality enhancement. *IEEE Transactions on Multimedia*, 25:6808–6820, 2023. doi: 10.1109/TMM.2022.3214775. URL https://ieeexplore.ieee.org/document/9919404/.

Liuhan Peng, Shuai Li, Yanbo Gao, Mao Ye, and Chong Lv. Hierarchical frequency-guided alignment transformer for compressed video quality enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 8349–8357, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/37784.

Wenzhe Shi, Jose Caballero, Ferenc Huszar, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/Shi_Real-Time_Single_Image_CVPR_2016_paper.html.

Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, 2012. URL https://ieeexplore.ieee.org/document/6316136/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

VQEG. Video datasets and organizations, n.d. URL https://vqeg.org/video-datasets-and-organizations/. Accessed September 21, 2026.

Yan Wang, Yusen Li, Gang Wang, and Xiaoguang Liu. PlainUSR: Chasing faster ConvNet for efficient super-resolution. In *Proceedings of the Asian Conference on Computer Vision*, pp. 4262–4279, 2024. URL https://openaccess.thecvf.com/content/ACCV2024/html/Wang_PlainUSR_Chasing_Faster_ConvNet_for_Efficient_Super-Resolution_ACCV_2024_paper.html.

Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general U-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17683–17693, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_Uformer_A_General_U-Shaped_Transformer_for_Image_Restoration_CVPR_2022_paper.html.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *Advances in Neural Information Processing Systems*, volume 34, pp. 12077–12090, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html.

Enze Xie, Junsong Chen, Junyu Chen, Han Cai, Haotian Tang, Yujun Lin, Zhekai Zhang, Muyang Li, Ligeng Zhu, Yao Lu, and Song Han. SANA: Efficient high-resolution text-to-image synthesis with linear diffusion transformers. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/34345e243156da67605d4b63d71c8d98-Abstract-Conference.html.

Xiph.Org. Derf's test media collection, n.d. URL https://media.xiph.org/video/derf/. Accessed September 21, 2026.

Ren Yang, Mai Xu, Zulin Wang, and Tianyi Li. Multi-frame quality enhancement for compressed video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6664–6673, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Yang_Multi-Frame_Quality_Enhancement_CVPR_2018_paper.html.

Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14821–14831, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Zamir_Multi-Stage_Progressive_Image_Restoration_CVPR_2021_paper.html.

Qianyu Zhang, Bolun Zheng, Xingying Chen, Quan Chen, Zunjie Zhu, Canjin Wang, Zongpeng Li, Xu Jia, and Chengang Yan. Hierarchical frequency-based upsampling and refining for HEVC compressed video enhancement. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(5):4423–4436, 2025. doi: 10.1109/TCSVT.2024.3517840. URL https://ieeexplore.ieee.org/document/10802929/.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Xinjian Zhang, Su Yang, Wuyang Luo, Longwen Gao, and Weishan Zhang. Video compression artifact reduction by fusing motion compensation and global context in a Swin-CNN based parallel architecture. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp.

3489–3497, 2023. doi: 10.1609/aaai.v37i3.25458. URL https://ojs.aaai.org/index.php/AAAI/article/view/25458.

Minyi Zhao, Yi Xu, and Shuigeng Zhou. Recursive fusion and deformable spatiotemporal attention for video compression artifact reduction. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 5646–5654, 2021. URL https://dl.acm.org/doi/10.1145/3474085.3475710.

Yichen Zhou, Pan Zhou, and Teck Khim Ng. Efficient cascaded multiscale adaptive network for image restoration. In *Computer Vision – ECCV 2024*, pp. 92–110, 2024. doi: 10.1007/978-3-031-72655-2_6. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/2840_ECCV_2024_paper.php.

Qiang Zhu, Yajun Qiu, Yu Liu, Shuyuan Zhu, and Bing Zeng. Compressed video quality enhancement with temporal group alignment and fusion. *IEEE Signal Processing Letters*, 31:1565–1569, 2024. URL https://ieeexplore.ieee.org/document/10555432/.

APPENDIX

A METRIC AND PROTOCOL DETAILS

Evaluation uses the complete 150–600-frame lengths of 18 JCT-VC Class A–E sequences. Excluding each initial I-frame leaves 7,962 P-frames. Incomplete final chunks repeat the last frame, but only valid outputs are scored. Spatial padding replicates boundary pixels to multiples of 16. Outputs are cropped back to their original extent before evaluation.

B IMPLEMENTATION AND MEASUREMENT DETAILS

Local processing uses channel widths 48/96/96 at $H/2$, $H/4$, and $H/8$, with 2, 2, and 4 local detail blocks, respectively. The context branch has one 192-channel SANA-derived block with 6 attention heads, temporal attention over 8 positions, and a feed-forward expansion ratio of 2.5. Its input projection downsamples by 2. Its output projection and pixel shuffle restore the $H/8$ grid. Reconstruction uses 2 local detail blocks at each skip-fusion scale.

GPU and host load are checked before and after measurement. After at least 1 second of warm-up, PECV-RT is timed using CUDA events in 3 runs of 50 measurements each. FPS uses the median of all 150 samples, counts 8 outputs per call, and is averaged arithmetically across QP37/42. HFGAT quality values come from the paper, whereas runtime uses the released model.

Single-output inference uses the same model and inputs but generates the eight enhanced frames in separate calls, without reusing intermediate features.

Table 6 compares context at $H/8$ and $H/16$ with the same local channel widths and one 192-channel context block. Moving context to $H/16$ reduces context-block computation from 19.540 to 4.885 GMACs/frame and total computation from 69.399 to 54.744 GMACs/frame, a reduction of 21.1%.

Table 6: Module-wise computation at 1080p (GMACs per enhanced frame). Spatial and temporal attention include their projections and attention matrix products. Conditioning includes QP embeddings and residual-gain prediction.

Module	Context at $H/16$	Context at $H/8$
Input feature extraction ($H/2$)	0.902	0.902
Local encoder ($H/2$)	8.122	8.122
Downsampling and local encoder ($H/4$)	10.077	10.077
Downsampling to $H/8$	1.203	1.203
Local refinement ($H/8$)	3.835	3.835
Context input projection	0.602	0.602
Context spatial attention	**1.307**	**5.227**
Context temporal attention	**1.228**	**4.913**
Context Mix-FFN	**2.350**	**9.400**
Context output projection	0.602	0.602
Reconstruction to $H/4$	11.280	11.280
Reconstruction to $H/2$	12.935	12.935
RGB output projection	0.301	0.301
Conditioning	0.000029	0.000029
Total GMACs/frame	**54.744**	**69.399**
MACs/pixel	**26,401**	**33,468**
Parameters (M)	**1.382**	**1.271**

C ADDITIONAL COMPONENT ABLATIONS

Table 7 reports additional VTM-QP42 ablations relative to the full model in Section 3.5.

No detail skips. The decoder's $H/2$ and $H/4$ skip inputs are replaced with zeros, leaving parameters and counted MACs unchanged. LPIPS increases by 9.391×10^{-3}, the largest deterioration among these variants, while PSNR decreases by 0.0046 dB. This supports the contribution of higher-resolution skip features to perceptual quality.

No temporal attention. Removing temporal attention decreases PSNR by 0.0069 dB and degrades LPIPS by 1.356×10^{-3}. Throughput changes from 102.02 to 102.58 FPS. Temporal attention thus provides an additional quality benefit at a small throughput cost.

No frame differences. Removing the adjacent-frame difference branch decreases PSNR by 0.0086 dB and degrades LPIPS by 2.543×10^{-3}. It saves 145 MACs/pixel and raises throughput from 102.02 to 104.51 FPS. These results support frame differences as a low-cost source of temporal information.

Frame-independent. Removing frame differences, temporal attention, and temporal convolution makes the eight frames independent during enhancement. This causes the largest PSNR drop (0.0230 dB) among these ablations and degrades LPIPS by 3.558×10^{-3}, while increasing throughput by only 6.0%. These results demonstrate the quality benefit of using cross-frame information.

Table 7: Additional VTM-QP42 component ablations. Quality differences are relative to the full model. Positive 100ΔLPIPS indicates worse perceptual quality. FPS and MACs/pixel use the same 1080p measurement protocol as Table 5.

Configuration	ΔPSNR (dB) ↑	100ΔLPIPS ↓	FPS ↑	MACs/pixel ↓	Params (M)
Full model	0	0	102.02	26,401	1.382
No detail skips	-0.0046	$+0.9391$	100.63	26,401	1.382
No temporal attention	-0.0069	$+0.1356$	102.58	25,808	1.233
No frame differences	-0.0086	$+0.2543$	104.51	26,256	1.381
Frame-independent	-0.0230	$+0.3558$	108.12	25,652	1.229