JOINT FIDELITY AND PERCEPTUAL MODULATION VIA ADAPTIVE CORRECTIONS FOR IMPLICIT-MOTION NEURAL VIDEO COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Implicit-motion neural video codecs achieve remarkable throughput by modeling motion entirely within high-dimensional contextual features. However, this highly coupled temporal representation causes quantization errors to propagate implicitly, limiting both rate-distortion and perceptual performance at high bitrates. To bridge this gap without compromising decoding efficiency, we propose a temporally consistent and perceptual-aware encoder-side latent refinement framework. Our method operates exclusively during encoding, ensuring that the decoder side and bitstream format remain entirely unchanged. First, we introduce a joint optimization strategy across I-frame and P-frame chunks that balances fidelity and perceptual objectives. The optimization leverages a simulated gradient annealing process, augmented by a residual direction search, to ensure robust convergence within the discrete latent space. Second, we explicitly model the future-reference dependency inherent in implicit codecs through a lookahead mechanism. By combining the lookahead reconstruction of future frames with cycle-consistent bidirectional optical flow, we construct a temporally corrected Region-of-Interest (ROI). Guided by this ROI, the optimization explicitly directs bit allocation toward areas that serve as critical temporal references along reliable motion trajectories. Extensive experiments demonstrate the superiority of our approach. On HEVC Class C and D sequences, compared to the frozen DCVC-UF (HT-L) baseline, our method achieves average BD-Rate reductions of 13.27% (PSNR), 18.06% (LPIPS), and 28.94% (DISTS) over all frames, confirming that targeted test-time optimization effectively eliminates the high-rate bottleneck of implicit-motion architectures.

1 INTRODUCTION

Video compression is a fundamental technology for storing and transmitting the ever-growing volume of video data. Rate-distortion efficiency dictates the trade-off a codec makes between the allocated bit rate and the delivered reconstruction quality. Traditional hybrid codecs such as HEVC Sullivan et al. (2012) and VVC Bross et al. (2021) have steadily improved this efficiency through handcrafted transforms and prediction. End-to-end training allows these components to be optimized jointly for rate-distortion. Following DVC Lu et al. (2019), which first replaced every hybrid component with a neural network, the DCVC family Li et al. (2021; 2023; 2024) introduced feature-domain conditional coding. Recent advancements prioritize practical deployment: DCVC-RT Jia et al. (2025) achieves real-time coding, and DCVC-UF Li et al. (2026) further accelerates decoding by grouping eight P-frames into a single chunk.

The exceptional throughput of these recent codecs stems from deliberate architectural choices. DCVC-RT Jia et al. (2025) removes explicit motion estimation and compensation, which historically account for a substantial part of the end-to-end codec complexity. Instead, it heavily downsamples the latent representation and forces the decoder to infer motion solely from contextual features. DCVC-UF Li et al. (2026) then ties eight P-frames to a single latent, enabling highly parallelized decoding. While this implicit-motion design dramatically reduces operational complexity and decoding time, it discards explicit temporal alignment signals. Consequently, at high bitrates where

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.

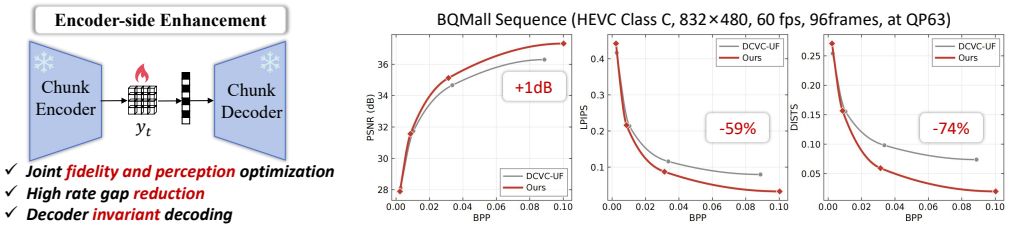


Figure 1: Encoder-side enhancement on BQMall. The latent is updated exclusively at the encoder; the decoder and its runtime remain completely unchanged.At the highest rate point, our method achieves a $+1.03$ dB PSNR gain while reducing LPIPS and DISTS by 59% and 74%, respectively.

quantization errors are otherwise small, this highly coupled temporal formulation causes errors to propagate implicitly, leading to a noticeable drop in rate-distortion performance.

We target the removal of this high-rate performance gap with an encoder-side latent refinement. By optimizing the bitstream exclusively at the encoder, the decoder parameters and decode time remain unchanged. This paradigm is suitable for offline applications where a video is encoded once and decoded many times, and where picture quality is paramount, such as video-on-demand and media archives. While prior test-time adaptation methods focus on optimizing fidelity and stop there Campos et al. (2019); Zhao et al. (2021); Lu et al. (2020); Bilican et al. (2025); Chen et al. (2025), we instead **jointly optimize fidelity and perceptual quality** within the same latent. This approach yields consistent improvements across all rates, with the most substantial gains at high bitrates. We construct a temporally corrected ROI using chunk-level bidirectional optical flow. A future region receives a higher penalty weight only when reliable motion corresponds to high-error locations in the current anchor frame. Bit allocation is directed toward these critical temporal references rather than uniformly across the future frame.

Concretely, the I-frame latent is refined first, and each subsequent P-frame chunk is optimized under the updated reference, with fidelity and perceptual losses jointly balanced. A simulated gradient annealing process, augmented by a residual direction search, evaluates a candidate set of step sizes and retains the configuration with the lowest loss after hard rounding. To construct the temporally corrected ROI, the next eight frames are simulated via a lookahead mechanism without bitstream commitment or decoder state updates. From this simulated reconstruction, the cycle-consistent part of the flow, the current-frame error warped onto the future, and the future-frame error are combined to form the ROI mask. On HEVC Class C and D sequences, our method achieves state-of-the-art test-time enhancement, reducing the average BD-Rate by 13.27% (PSNR), 18.06% (LPIPS), and 28.94% (DISTS) relative to the frozen DCVC-UF (HT-L) baseline.

Our main contributions are summarized as follows:

- We introduce a joint test-time optimization for I- and P-frame chunks to simultaneously improve fidelity and perceptual quality. It employs simulated gradient annealing with an adaptive step-size search to directly optimize the hard-rounded loss.
- We propose a temporally corrected ROI driven by a lookahead mechanism and chunk-level bidirectional optical flow, directing bit allocation toward reliable motion references.
- On HEVC Class C and D, our method achieves state-of-the-art encoder-side enhancement. By reducing BD-Rate by 13.27% (PSNR), 18.06% (LPIPS), and 28.94% (DISTS) against the DCVC-UF baseline, it effectively closes the high-rate gap of implicit-motion codecs.

2 Related Work

2.1 Neural Video Compression

Neural video compression has moved from explicit motion-compensated prediction toward feature-domain and implicit temporal modeling. DVC Lu et al. (2019) explicitly estimates and compresses optical flow, then performs motion compensation and residual coding. DCVC Li et al. (2021) shifts

prediction into a learned feature space with high-dimensional temporal contexts, and its successors enrich this with diverse contexts Li et al. (2023) and feature modulation Li et al. (2024). DCVC-RT Jia et al. (2025) replaces explicit motion modules with implicit temporal modeling, and DCVC-UF Li et al. (2026) groups multiple frames into one compact latent with cross-frame interactions, improving throughput and long-term temporal representation. Implicit neural representations have also been explored for practical video compression Gao et al. (2025).

2.2 Perceptual Optimization for Compression

Pixel-wise objectives such as MSE, PSNR, and MS-SSIM do not always match perceptual quality, and perceptual optimization has been explored to reduce the over-smoothing. PLVC Yang et al. (2022) uses a recurrent conditional GAN to improve spatial realism and temporal consistency, and CGVC-T Du et al. (2024) combines adversarial learning with contextual coding to improve LPIPS, DISTS, FID, and KID. PNVC-CR Liang et al. (2026) brings perceptual optimization into a modern conditional codec and substantially improves these metrics, though fidelity-oriented variants remain better in PSNR. Recent works inject diffusion priors to recover perceptually plausible details and temporal consistency at very low bitrates Ma & Chen (2026); Mao et al. (2026); Wang et al. (2026).

2.3 Latent Refinement and Content-Adaptive Optimization

A learned encoder performs amortized inference and may not yield the optimal latent for every sample. Latent refinement closes this gap by optimizing it at test time with the decoder fixed. Campos et al. Campos et al. (2019) optimize image latents under the R-D objective, and Zhao et al. Zhao et al. (2021) jointly optimize latents and side information in an encoder-side R-D framework. Test-time adaptation also refines latents under distribution shifts Chen et al. (2025). In video, Lu et al. Lu et al. (2020) adapt encoder parameters to each test video with the decoder unchanged, Tang et al. Tang et al. (2024) refine optical-flow features online, and Bilican et al. Bilican et al. (2025) address the train-test motion mismatch through content-adaptive inference. DCVC-DT Zhang et al. (2026) refines the encoded latent at inference time with encoder and decoder fixed, and adjusts frame-level R-D per reconstruction-quality fluctuations, improving adaptation and mitigating error propagation.

3 Proposed Method

3.1 Method Overview

To enhance the high rate coding of DCVC-UF, we keep the network parameters unchanged and modulate only the latent of the chunk. The modulated latent is quantized and entropy coded exactly as in the original codec, so the decoder and the bitstream syntax stay the same. Figure 2 shows where this modulation is placed in the codec and how the perceptual and fidelity loss and the flow aligned mask are formed.

Codec and task. For an 8-frame P chunk, DCVC-UF encodes one latent y and decodes $\hat{x}$ from $\hat{y}$ and the context feature. With a quantization step q_{enc} determined by the quantization parameter (QP) and entropy mean μ,

$$y \leftarrow y \cdot q_{\mathrm{enc}}, \qquad \hat{y} = \big(\mathrm{round}(y-\mu)+\mu\big) \cdot q_{\mathrm{dec}}, \tag{1}$$

and training minimizes $\mathcal{L}_{\mathrm{UF}} = \lambda\, \mathcal{D}_{\mathrm{UF}} + \mathcal{R}$, where $\mathcal{D}_{\mathrm{UF}}$ is the official UF 8-frame MSE with three frame weights 1.5, 0.16, and 0.4 (Appendix F). At test time the encoder emits y in one forward pass. We replace it by a modulated y^* that achieves a better trade-off among fidelity, perception, and rate at the current operating point, while $\hat{y}$ and the bitstream stay compatible with the frozen decoder.

Test-time objective. The current chunk loss refines the latent for the operating rate point,

$$\mathcal{L}_A = \lambda\, \mathcal{D}_{\mathrm{UF}} + \mathcal{R} + \lambda\, \alpha\big(\mathrm{LPIPS} + \mathrm{DISTS}\big), \quad \alpha = 0.0075. \tag{2}$$

The I-frame latent is refined first and each P chunk is optimized on the updated DPB. Hard rounding blocks the gradient, so the backward pass uses an annealed soft quantizer, and a line search along the latest increment rescales that step before the latent is updated. Section 3.2 specifies this optimizer.

We then raise the confidence of regions according to optical flow. The following chunk is previewed without writing bits. A frozen flow network aligns the current anchor with each future frame, and

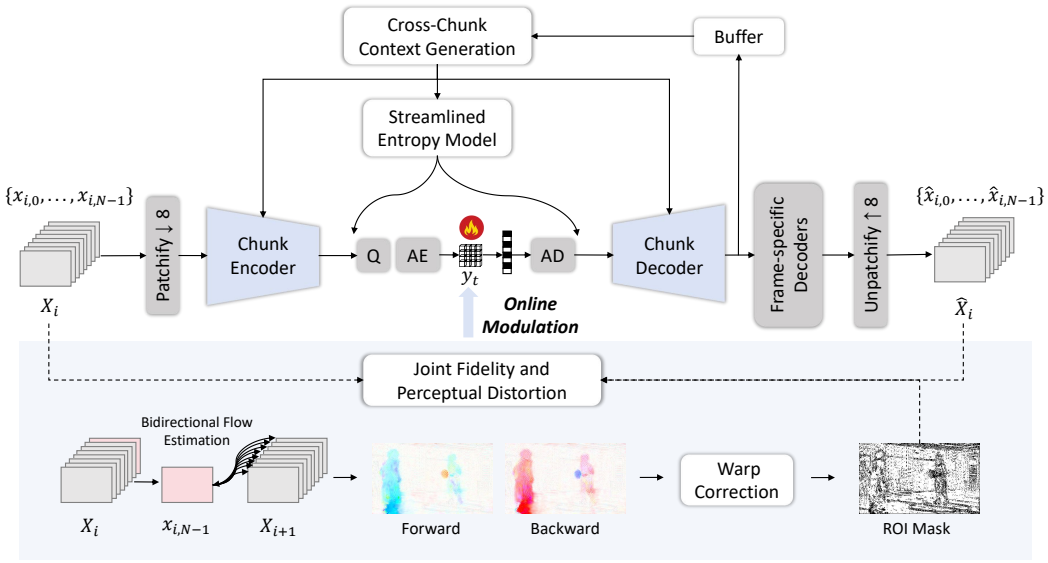


Figure 2: Method overview. The frozen DCVC-UF codec emits a chunk latent that we modulate at encode time. The current-chunk loss combines fidelity, rate, and perception. A preview of the next chunk, aligned by explicit flow, adds a motion-weighted future distortion. The gradient updates only the current latent.

cycle consistent trajectories mark where the anchor error and the future error coincide, raising the weight of the future distortion $\mathcal{D}_B$ only there. The preview loss is added as

$$\mathcal{L} = \mathcal{L}_A + \gamma\,\mathcal{L}_B, \tag{3}$$

The gradient is taken only with respect to the current latent. The spatial weights are computed once from the initial reconstruction and kept unchanged during the update. Section 3.3 specifies this ROI.

3.2 Online Joint Fidelity and Perception Optimization

This optimizer has three parts: joint modulation of the I frame and the P chunks, a joint fidelity and perception objective, and simulated annealing gradient descent with adaptive step sizes.

3.2.1 Quantization Aware Gradient with SGA and Tau Annealing

Hard quantization $\hat{y} = \mathrm{round}(y)$ has zero gradient. We use a Stochastic Gumbel Annealing (SGA) surrogate that places y between neighboring integers with a soft weight, so the gradient can pass through,

$$\tilde{y} = \mathrm{SGA}(y, \tau), \qquad \tau_t = \min\big(0.5,\ 0.5\,e^{-\kappa t}\big), \quad \kappa = 0.001, \tag{4}$$

where τ anneals from 0.5 toward 0. The backward pass uses $\tilde{y}$. The update and the step size search below use hard $\hat{y}$, so the training loss converges to the evaluation loss.

We extend the UF training loss with perceptual terms on BT.709 RGB,

$$\mathcal{L}_A(y) = \lambda\,\mathcal{D}_{\mathrm{UF}}(x,\hat{x}) + \mathcal{R}(\hat{y}) + \lambda\,\alpha\big(\mathrm{LPIPS}(x,\hat{x}) + \mathrm{DISTS}(x,\hat{x})\big), \quad \alpha = 0.0075, \tag{5}$$

where $\mathcal{D}_{\mathrm{UF}} = \sum_{t=0}^{7} c_t\,\mathrm{MSE}(x_t,\hat{x}_t)$ uses the codec frame weights $c = [1.5, 0.16, 0.4]$ over the chunk. LPIPS uses AlexNet and DISTS is computed on BT.709 converted RGB frames. The scalar α comes from a normalized MSE:LPIPS:DISTS ratio. The default $4{:}1{:}1$ gives $\alpha = 0.0075$. The formula is in Appendix E.

The I frame latent y_I is refined first, since it seeds the whole P frame reference chain and updates the DPB. Each P chunk is then optimized on this new reference.

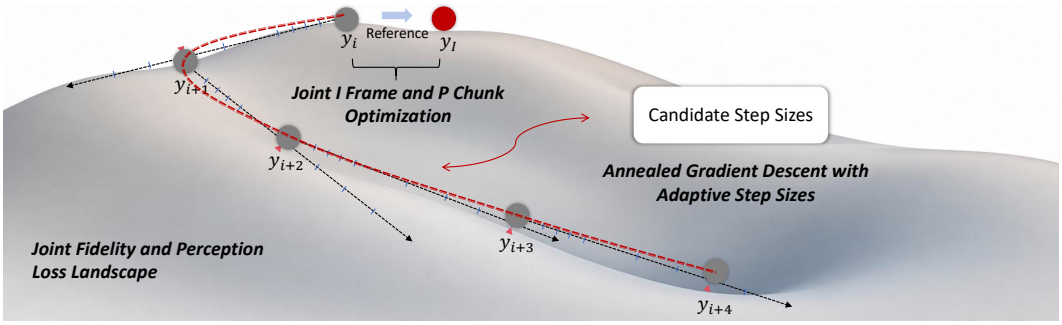


Figure 3: Simulated annealing gradient descent with adaptive step sizes. Black points are accepted latents. At the current anchor, hard round losses are evaluated along the latest Adam increment, and the minimizer (red) updates the latent. The anchor then moves there.

3.2.2 Simulated Annealing Gradient Descent with Adaptive Step Sizes

The loss after rounding is flat between integers, and successive Adam steps on y point in very different directions, so one fixed learning rate does not fit. After a warmup of $W = \min(8, N)$ Adam steps, each later step tries a set of step sizes along the latest increment,

$$y_\eta = y_{\text{anchor}} + \eta\,(y - y_{\text{anchor}}), \qquad \eta \in \mathcal{H}, \tag{6}$$

evaluates Eq. 5 with hard rounding and no backward pass, and keeps

$$\eta^\star = \arg\min_{\eta \in \mathcal{H}} \mathcal{L}_A(y_\eta), \qquad \mathcal{H} = \{0.15, 0.25, 0.35, 0.5, 1, 5, 8, 12, 16\}. \tag{7}$$

The anchor then moves to the updated y, so the next search only scales the latest Adam step. Adam moments are not reset. Figure 3 shows one step: the updated latent becomes the next anchor, and the next search follows only the new increment. This is how the adaptive step sizes are chosen at each step in the same optimizer.

3.3 Temporally Corrected ROI with Optical Flow

Let A denote the current 8-frame chunk with latent y_A, and B the subsequent chunk. Since decoding A produces the temporal reference features used to encode B, refining y_A affects the rate-distortion performance of both chunks. To account for this inter-chunk dependency, we introduce a lookahead mechanism to incorporate the coding cost of B into the optimization of y_A. We further use optical flow to align reconstruction errors in the last frame of A with those in B. A temporally corrected ROI then assigns greater weight to future regions where both errors are large and the motion correspondence is reliable. Figure 4 illustrates the construction.

For each candidate y_A, we decode A and use the resulting reference features to reconstruct B. Let $\mathcal{L}_A$ and $\mathcal{L}_B$ denote the joint objective in Eq. 5 evaluated on A and the preview of B, respectively. We optimize the lookahead objective

$$\mathcal{L}_{\text{LA}}(y_A) = \mathcal{L}_A(y_A) + \gamma\,\mathcal{L}_B(y_A), \qquad \gamma = 1. \tag{8}$$

During gradient updates, the preview uses hard rounding with a straight-through estimator. Gradients from $\mathcal{L}_B$ propagate through the reference features to y_A, which is the only optimization variable. The preview produces no bitstream, and the decoder state is restored afterward. We apply the lookahead when B is complete and no reference reset occurs between the two chunks; otherwise, we use $\mathcal{L}_A$ alone. As coding progresses, B becomes the current chunk and is refined using the resulting reference state.

Naive $\mathcal{L}_B$ assigns uniform spatial weights within each future frame. To prioritize regions with large reconstruction errors at corresponding locations in both chunks, we introduce a temporally corrected ROI based on reliable optical flow correspondences. We use the last frame of A as the temporal anchor because it is closest in time to B. Let $(x_A^{(7)}, \hat{x}_A^{(7)})$ and $\{(x_B^{(j)}, \hat{x}_B^{(j)})\}_{j=0}^{7}$ denote the

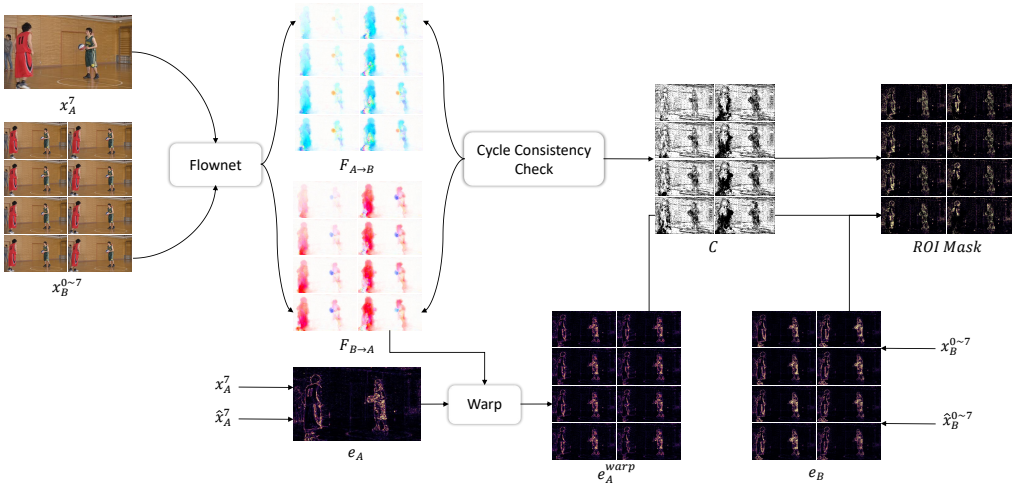


Figure 4: Temporally corrected ROI construction. Bidirectional optical flow establishes correspondences between the anchor frame $x_A^{(7)}$ and each frame of B. Backward warping aligns the anchor error e_A with the future error e_B. Their product, filtered by the cycle consistency mask C and normalized by the 95th percentile of positive scores in each frame, yields the ROI mask.

source and reconstructed frames of the anchor and the future chunk, respectively. Their error maps are computed by

$$e_A = \mathrm{dist}\left(x_A^{(7)}, \hat{x}_A^{(7)}\right), \qquad e_B^{(j)} = \mathrm{dist}\left(x_B^{(j)}, \hat{x}_B^{(j)}\right), \tag{9}$$

where dist evaluates the pixel-wise reconstruction error. These maps define ROI weights that remain fixed throughout refinement, whereas the lookahead loss uses a new preview for each candidate y_A.

To align the anchor error with each future frame, a frozen pretrained SPyNet Ranjan & Black (2017) estimates the forward field $F_{A\to B}^{(j)}$ and the backward field $F_{B\to A}^{(j)}$ from the source frames. We warp the anchor error into frame j of B by sampling e_A at $p + F_{B\to A}^{(j)}(p)$, using bilinear interpolation with zero padding:

$$e_A^{\mathrm{warp},(j)}(p) = \mathrm{warp}\left(e_A;\, F_{B\to A}^{(j)}\right)(p). \tag{10}$$

To reject unreliable correspondences caused by occlusion or flow errors, we check forward–backward consistency. The cycle residual is

$$r^{(j)} = F_{B\to A}^{(j)} + \mathrm{warp}\left(F_{A\to B}^{(j)};\, F_{B\to A}^{(j)}\right). \tag{11}$$

For pixels whose sampling coordinates $p + F_{B\to A}^{(j)}(p)$ and return coordinates $p + r^{(j)}(p)$ both lie within the image boundaries, we define the binary reliability mask as

$$C^{(j)} = \mathbf{1}\left[\|r^{(j)}\|^2 \le 0.01\left(\|F_{B\to A}^{(j)}\|^2 + \|\mathrm{warp}(F_{A\to B}^{(j)};\, F_{B\to A}^{(j)})\|^2\right) + 0.5\right]. \tag{12}$$

We set $C^{(j)} = 0$ elsewhere. All norms are evaluated per pixel. The ROI score emphasizes regions with large reconstruction errors in both the anchor and future frames along reliable motion correspondences:

$$S^{(j)} = C^{(j)} \cdot e_B^{(j)} \cdot e_A^{\mathrm{warp},(j)}. \tag{13}$$

To account for differences in score magnitude across future frames, we normalize each frame independently. Let $q^{(j)}$ denote the 95th percentile of the strictly positive values in $S^{(j)}$. The normalized mask is

$$M^{(j)} = \mathrm{clip}\left(S^{(j)}/q^{(j)},\, 0,\, 1\right). \tag{14}$$

Table 1: BD-Rate against DCVC-UF (HT-L). Negative values indicate bitrate savings.

Method	Class C			Class D		
	PSNR	LPIPS	DISTS	PSNR	LPIPS	DISTS
DCVC-UF	0	0	0	0	0	0
DCVC-RT	−3.95	+26.84	+40.02	+7.66	+30.86	+37.60
DCVC-DT*	−8.73	+2.09	+12.27	−7.44	+4.10	+7.27
Ours	**−14.35**	**−19.39**	**−26.53**	**−12.19**	**−16.74**	**−31.35**

Table 2: Ablation of the online optimizer.

Method	PSNR	LPIPS	DISTS
Adam (50)	−2.07	−5.42	−10.30
search (50)	−4.21	−6.75	−9.76
Adam (500)	−9.60	−12.06	−20.15
search (500)	−11.11	−16.68	−28.00
Adam (1000)	−13.18	−16.12	−29.10
search (1000)	**−13.88**	**−18.90**	**−35.85**

Table 3: Ablation of the perceptual loss ratio.

Loss ratio	PSNR	LPIPS	DISTS
2:1:1	−7.60	−10.19	−20.54
4:1:1	−8.43	−12.67	−14.98
6:1:1	−8.75	−10.25	−12.08
8:1:1	−8.46	−10.56	−10.85

Table 4: Ablation of the future distortion term.

Method	PSNR	LPIPS	DISTS
Optimizer only	−11.53	−15.98	−30.07
+ lookahead	−14.72	−20.39	−33.80
+ Correct ROI	**−15.00**	**−20.71**	**−34.12**

Table 5: Ablation of the candidate step sizes.

Step set	PSNR	LPIPS	DISTS
[0.5, 5]	−11.11	−16.68	−28.00
[0.15, 16]	−11.60	−18.00	−31.63

If $S^{(j)}$ contains no positive entries, we set $M^{(j)} = 0$. We then convert the mask into spatial weights with unit mean:

$$w^{(j)} = \frac{1 + \beta\, M^{(j)}}{\mathrm{mean}\big(1 + \beta\, M^{(j)}\big)}, \tag{15}$$

where the mean is computed over the valid image area; padded pixels retain weight one. The additive baseline gives every pixel a positive weight, while normalization keeps the mean spatial weight fixed as the ROI size changes. Setting $\beta = 0$ recovers uniform lookahead.

We multiply both the source and reconstructed future frames by $\sqrt{w^{(j)}}$ before evaluating the codec's fidelity distortion. The weighted squared errors are averaged spatially before the original nonlinear channel aggregation, and the codec's temporal frame weights are retained. The rate, LPIPS, and DISTS terms remain unchanged. We use the resulting $\mathcal{L}_B$ in Eq. 8 for both gradient updates and the candidate search in Sec. 3.2. This objective uses temporal error correspondences to guide refinement of y_A, accounting for both the reconstruction quality of A and its role as a reference for B.

4 Experiments

4.1 Experimental Setup

We evaluate on the first 96 frames of HEVC Class C (832×480) and Class D (416×240) sequences Boyce et al. (2018) at QPs $\{0, 21, 42, 63\}$. The pre-trained DCVC-UF (HT-L) Li et al. (2026) serves as both our base framework and evaluation anchor, with all network parameters frozen. By exclusively refining chunk latents at encode time, our method ensures bitstream compliance and zero decoder overhead. We compare against DCVC-RT Jia et al. (2025) and DCVC-DT*, a baseline transferring the DCVC-DT Zhang et al. (2026) refinement strategy to DCVC-UF. Quality metrics include YUV-PSNR (YUV 6:1:1), LPIPS (AlexNet), and DISTS. BD-Rate Bjøntegaard (2001) is computed against the DCVC-UF anchor, with all bitrates measured directly from the actual generated bitstreams. We conduct all test-time optimizations over 500 steps. I-frame latents are refined via AdaRound, whereas P-frame chunks employ Adam Kingma & Ba (2015) ($lr = 10^{-3}$) augmented by Stochastic Gumbel Annealing ($\kappa = 0.001$) and a 9-candidate residual direction search. The lookahead dependency and ROI scaling are fixed at $\gamma = 1$ and $\beta = 1$, respectively.

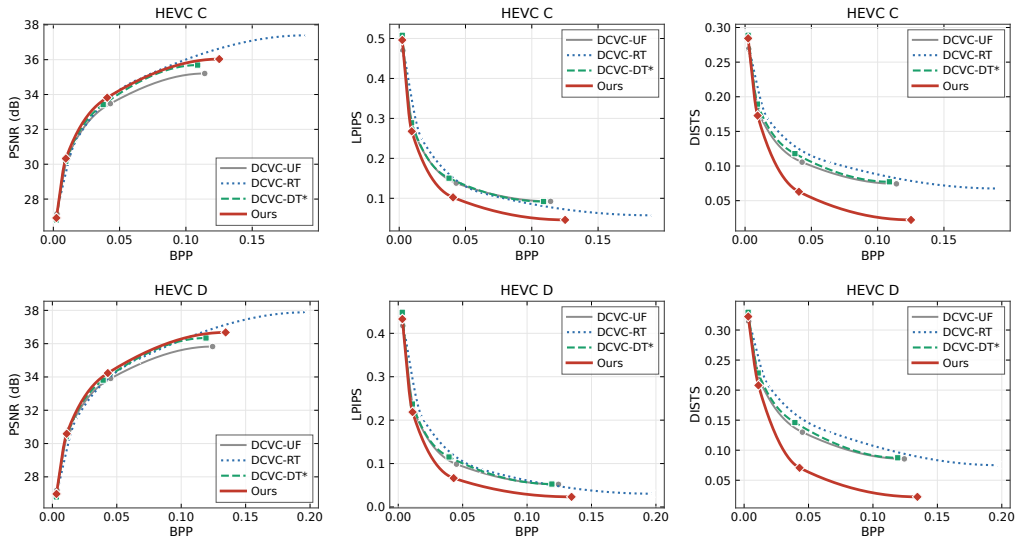


Figure 5: RD curves on HEVC Class C and Class D, DCVC-UF vs. Ours, 500 online steps. Each point is the mean over the four sequences.

4.2 Quantitative Evaluation

Table 1 reports the BD-Rate performance relative to the frozen DCVC-UF anchor. To isolate the source of our gains, we analyze DCVC-DT*, which represents a PSNR-only test-time optimization (omitting our perceptual loss, lookahead ROI, and residual search). While this setting improves fidelity (-8.73% PSNR on Class C and -7.44% on Class D), it notably degrades perceptual quality ($+12.27\%$ and $+7.27\%$ DISTS). This confirms that our perceptual improvements stem directly from our proposed joint formulation rather than the test-time optimizer alone. Conversely, our full method with 500 refinement steps yields substantial bitrate savings across all quality metrics.

Averaged across all sequences, our approach achieves BD-Rate reductions of 14.35% (PSNR), 19.39% (LPIPS), and 26.53% (DISTS) on Class C. Similarly, Class D exhibits savings of 12.19% (PSNR), 16.74% (LPIPS), and 31.35% (DISTS). Notably, DISTS improvements are more pronounced on Class D, whereas PSNR gains are larger on Class C. These positive trends hold consistently across all eight sequences. The most striking perceptual gain occurs on *BQSquare* (-36.14% DISTS BD-Rate), a texture-heavy sequence where the implicit-motion bottleneck of the baseline is typically most severe. Detailed rate-distortion (RD) curves are provided in Fig. 5.

4.3 Ablation Study

We conduct the ablation studies on the HEVC Class D dataset to validate the effectiveness of our joint optimization and lookahead ROI mechanisms.

Optimizer and Step Count. Table 2 compares Adam with our residual direction search at 50, 500, and 1000 steps. The search directly evaluates the hard-rounded objective that SGA only approximates, so it wins at every matched step count, by 7.9% DISTS at 500 steps. We adopt 500 steps: 1000 roughly doubles the cost for only 2.8% more PSNR.

Perceptual Weight Ratio. Table 3 sweeps the MSE:LPIPS:DISTS ratio on *BasketballPass*. 4:1:1 ($\alpha = 0.0075$) is the best trade-off: larger perceptual weights improve DISTS but erode PSNR, smaller ones do the opposite. The other rows use $\alpha \in \{0.015, 0.005, 0.00375\}$.

Future ROI Configuration. Table 4 ablates the future term on Class D. Uniform lookahead already recovers most of the gap by removing the short-sightedness of independent chunk optimization. Correct ROI (Eq. 13) adds consistent PSNR (-0.28%) and LPIPS (-0.32%) gains and a -0.31%

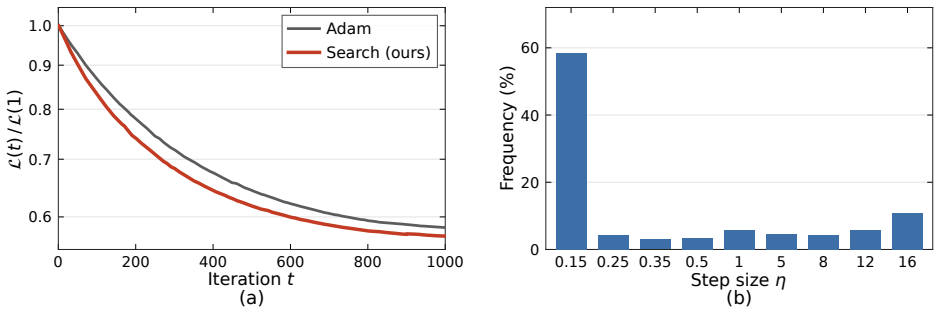


Figure 6: (a) Training loss on Class D. Residual direction search descends faster than Adam early. (b) Picked step size η on one chunk.

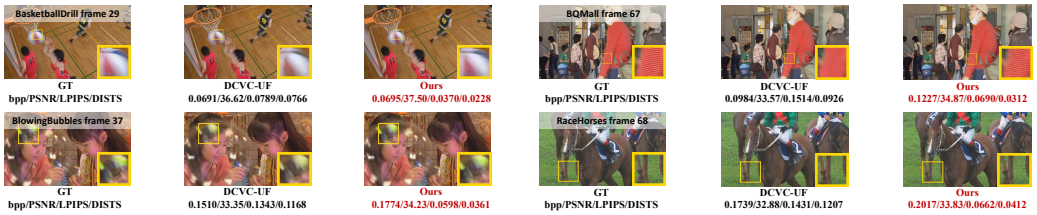


Figure 7: Subjective comparison at the highest rate (QP 63). Each row shows the source, DCVC-UF, and Ours for one P-frame. The yellow box marks the crop enlarged at the lower right.

DISTS mean, though *BQSquare* slightly regresses (Lookahead -43.95% vs. Correct -43.35%): on texture-heavy frames with weaker cycle consistency, a uniform future term can be safer for DISTS.

I-frame modulation and step size grid. I-only modulation improves all-frame PSNR BD-Rate (-12.8% vs. -6.6% for P-only) but degrades P-frame PSNR from -9.6% to -5.4%, so joint I/P modulation is required. Extending the candidate set from $[0.5, 5]$ to $[0.15, 16]$ at 500 steps improves DISTS by -3.6% (Table 5). The lower edge is hit 50% of the time at 0.5 on the short grid and 58% at 0.15 on the long grid, so a single learning rate multiplier fails in both shrink and enlarge regimes. Figure 6(a) shows that residual direction search descends faster than Adam on Class D and the two meet by 1000 steps. Figure 6(b) shows that the picked η often lands on 0.15 or 16.

4.4 Subjective Analysis

Figure 7 shows reconstructed P-frames at the highest rate point on four sequences from HEVC Class C and D. DCVC-UF over-smooths texture edges and thin structures, which is the visual form of the implicit-motion bottleneck. On *BQMall* the clothing stripes merge into a flat band. On *BasketballDrill* the ball seam is lost. On *RaceHorses* the grass and the leg edge lose definition. On *BlowingBubbles* the bubble highlights are dull. The refined latent restores these textures by putting extra rate on the temporally referenced locations.

5 Conclusion

Implicit motion codecs drop explicit motion, so only one chunk latent y is transmitted, and that y is not optimal at a given rate point. We refine the I-frame and P-chunk latents at encode time under a joint objective of fidelity, rate, and perception. After each Adam update, simulated annealing gradient descent with adaptive step sizes keeps the scale with the lowest hard rounded loss, and the next chunk is previewed so that a cycle consistent flow ROI can raise the weight of future errors along reliable motion. The decoder and the bitstream syntax stay unchanged. The same procedure can be applied to other implicit motion codecs that emit a chunk latent, without retraining the decoder.

REFERENCES

Ahmet Bilican, M Akın Yılmaz, and A Murat Tekalp. Content-adaptive inference for state-of-the-art learned video compression. *IEEE Open Journal of Signal Processing*, 6:498–506, 2025.

Gisle Bjøntegaard. Calculation of average psnr differences between RD-curves. In *ITU-T SG16 Q.6 Document VCEG-M33*, 2001.

Jill Boyce, Karsten Suehring, Xiang Li, and Vadim Seregin. JVET-J1010: JVET common test conditions and software reference configurations. JVET document JVET-J1010, 2018.

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.

Joaquim Campos, Simon Meierhans, Abdelaziz Djelouah, and Christopher Schroers. Content adaptive optimization for neural image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

Kecheng Chen, Pingping Zhang, Tiexin Qin, Shiqi Wang, Hong Yan, and Haoliang Li. Test-time adaptation for image compression with distribution regularization. In *International Conference on Learning Representations*, 2025.

Pengli Du, Ying Liu, and Nam Ling. CGVC-T: Contextual generative video compression with transformers. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 14(2): 209–223, 2024.

Ge Gao, Ho Man Kwan, Fan Zhang, and David Bull. PNVC: Towards practical INR-based video compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 3068–3076, 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.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

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, 2023.

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, 2024.

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.

Xiongzhuang Liang, Chuanbo Tang, Zhuoyuan Li, Li Li, and Dong Liu. Perceptual neural video compression with color separation and rank chain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5348–5358, 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.

Guo Lu, Chunlei Cai, Xiaoyun Zhang, Li Chen, Wanli Ouyang, Dong Xu, and Zhiyong Gao. Content adaptive and error propagation aware deep video compression. In *European conference on computer vision*, pp. 456–472. Springer, 2020.

Wenzhuo Ma and Zhenzhong Chen. DiffVC-RT: Towards practical real-time diffusion-based perceptual neural video compression. *arXiv preprint arXiv:2601.20564*, 2026.

Qi Mao, Hao Cheng, Tinghan Yang, Libiao Jin, and Siwei Ma. Generative neural video compression via video diffusion prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Anurag Ranjan and Michael J. Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4161–4169, 2017.

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.

Chuanbo Tang, Xihua Sheng, Zhuoyuan Li, Haotian Zhang, Li Li, and Dong Liu. Offline and online optical flow enhancement for deep video compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 5118–5126, 2024.

Lingdong Wang, Guan-Ming Su, Divya Kothandaraman, Tsung-Wei Huang, Mohammad Hajiesmaili, and Ramesh K Sitaraman. Low-bitrate video compression through semantic-conditioned diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4495–4505, 2026.

Ren Yang, Radu Timofte, and Luc Van Gool. Perceptual learned video compression with recurrent conditional gan. In *IJCAI*, pp. 1537–1544, 2022.

Tiange Zhang, Rongqun Lin, Xiandong Meng, Haofeng Wang, Xing Tian, Qi Zhang, and Siwei Ma. Neural video compression with domain transfer. In *2026 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1551–1555. IEEE, 2026.

Jing Zhao, Bin Li, Jiahao Li, Ruiqin Xiong, and Yan Lu. A universal encoder rate distortion optimization framework for learned compression. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1880–1884. IEEE, 2021.

A Appendix

A AI Use Statement

In compliance with ICLR guidelines, we disclose that generative AI tools were utilized exclusively to edit this manuscript to improve readability and refine the academic prose. We explicitly confirm that no AI tools were employed in any core research workflows, including formulating mathematical claims, implementing methods, designing experiments, or interpreting empirical results. All authors have meticulously reviewed the AI-assisted revisions to ensure absolute factual accuracy and strict preservation of our original scientific intent. Ultimately, we assume full responsibility for the final content of this work.

B Reproducibility Statement

All results in this paper are obtained with the official DCVC-UF (HT-L) checkpoint Li et al. (2026) frozen, and only the chunk latents are refined at encode time, so the decoder and the bitstream syntax are unchanged. The test-time optimizer, the lookahead ROI construction, and all hyperparameters ($\alpha = 0.0075$, $\gamma = 1$, $\beta = 1$, $\kappa = 0.001$, 500 online steps, the 9-candidate step set $\mathcal{H} = \{0.15, 0.25, 0.35, 0.5, 1, 5, 8, 12, 16\}$) are described in Sec. 3.2 and Sec. 3.3. Evaluation uses the first 96 frames of HEVC Class C and D at QPs $\{0, 21, 42, 63\}$, with bitrates measured from the actual generated bitstreams and BD-Rate computed against the DCVC-UF anchor. We will release the source code upon acceptance.

C Per-Sequence BD-Rate

Table 6 reports the per-sequence BD-Rate of DCVC-RT, DCVC-DT*, and our method against the DCVC-UF (HT-L) anchor on all eight HEVC Class C and Class D sequences. The class means reproduce the headline numbers in Table 1. Our method is the only one that simultaneously improves all three metrics on every sequence, and the perceptual gain is largest on texture-heavy sequences such as *BQSquare* (-36.14% DISTS).

Table 6: Per-sequence BD-Rate (%) against DCVC-UF (HT-L), all-frame (I+P). Negative values indicate bitrate savings.

	DCVC-RT			DCVC-DT*			Ours		
Sequence	PSNR	LPIPS	DISTS	PSNR	LPIPS	DISTS	PSNR	LPIPS	DISTS
BasketballDrill	+3.58	+51.90	+59.19	−8.46	+6.12	+14.44	**−10.78**	**−20.03**	**−30.01**
BQMall	−1.52	+35.19	+50.47	−9.86	+0.26	+12.00	**−16.45**	**−18.02**	**−24.46**
PartyScene	+19.97	+38.98	+45.46	−4.60	+5.85	+23.98	**−13.06**	**−15.95**	**−21.81**
RaceHorses	−37.85	−18.73	+4.96	−12.01	−3.85	−1.34	**−17.13**	**−23.55**	**−29.85**
Class C	−3.95	+26.84	+40.02	−8.73	+2.09	+12.27	**−14.35**	**−19.39**	**−26.53**
BasketballPass	−1.77	+38.28	+36.22	−9.74	+4.79	+9.01	**−13.06**	**−13.46**	**−27.30**
BlowingBubbles	+18.96	+33.70	+41.68	−6.61	+2.23	+8.18	**−10.91**	**−19.42**	**−31.43**
BQSquare	+37.71	+52.77	+61.28	−2.54	+11.63	+13.05	**−9.49**	**−13.34**	**−36.14**
RaceHorses	−24.28	−1.29	+11.21	−10.86	−2.26	−1.16	**−15.31**	**−20.73**	**−30.53**
Class D	+7.66	+30.86	+37.60	−7.44	+4.10	+7.27	**−12.19**	**−16.74**	**−31.35**

D Per-Sequence RD Curves

Figures 8 and 9 plot the full rate-distortion curves on every sequence, comparing our method against the frozen DCVC-UF anchor across PSNR, LPIPS, and DISTS. On every sequence and every metric, the red curve lies strictly on the favorable side of the grey anchor curve, and the gap widens at high rate, which is exactly the regime where the implicit-motion bottleneck of DCVC-UF is most severe.

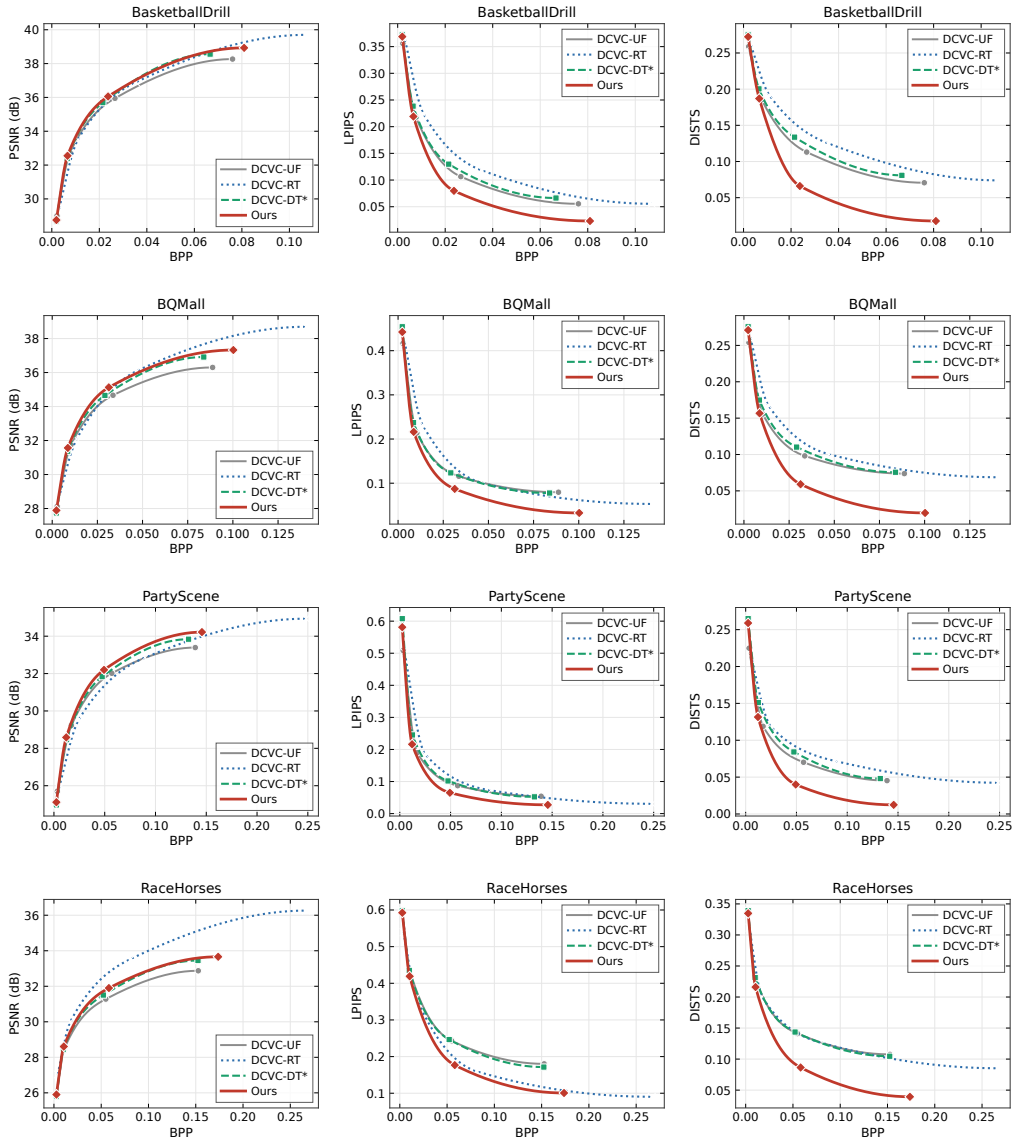


Figure 8: Per-sequence RD curves on HEVC Class C. Rows: BasketballDrill, BQMall, PartyScene, RaceHorses. Columns: PSNR, LPIPS, DISTS. Grey: DCVC-UF (HT-L); red: Ours.

E Perceptual Weight α

Let $r_{\text{mse}} : r_{\text{lpips}} : r_{\text{dists}}$ be the loss ratio. The MSE term against rate is left unchanged. Each perceptual term is then scaled by

$$\alpha_{\text{lpips}} = 0.03\,\frac{r_{\text{lpips}}}{r_{\text{mse}}}, \qquad \alpha_{\text{dists}} = 0.03\,\frac{r_{\text{dists}}}{r_{\text{mse}}}. \tag{16}$$

The factor 0.03 maps LPIPS and DISTS onto the scale of the UF 8-frame MSE. When $r_{\text{lpips}} = r_{\text{dists}}$, both reduce to the single α in Eq. 5. The default $4{:}1{:}1$ gives $\alpha = 0.0075$. Table 3 uses $2{:}1{:}1$, $4{:}1{:}1$, $6{:}1{:}1$, and $8{:}1{:}1$, which correspond to $\alpha \in \{0.015, 0.0075, 0.005, 0.00375\}$.

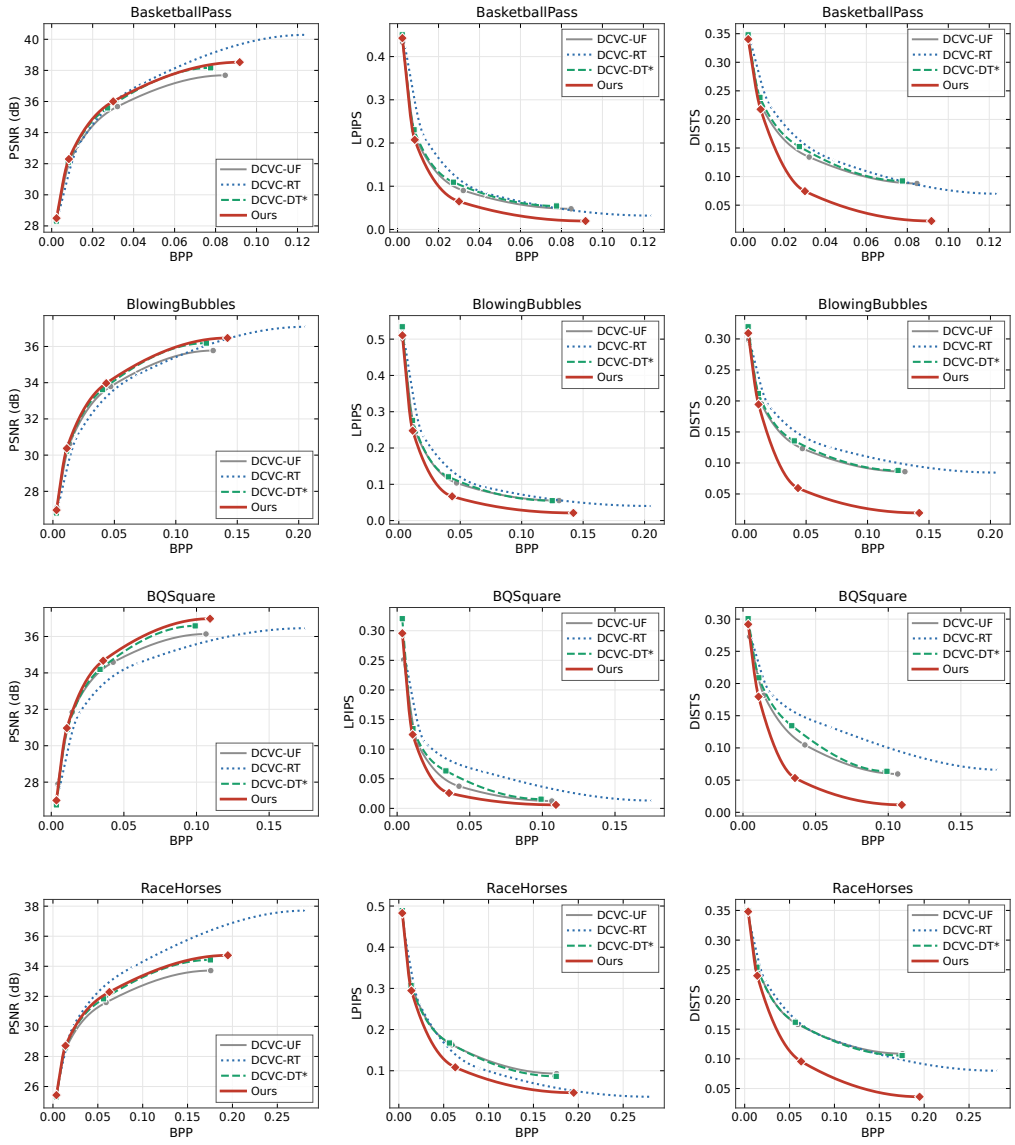


Figure 9: Per-sequence RD curves on HEVC Class D. Rows: BasketballPass, BlowingBubbles, BQSquare, RaceHorses. Columns: PSNR, LPIPS, DISTS. Grey: DCVC-UF (HT-L); red: Ours.

F UF Frame Weights

The official DCVC-UF (HT-L) training loss does not assign one weight per frame. It uses three coefficients $w = [1.5,\ 0.16,\ 0.4]$ and maps them onto the eight frames of a P chunk by

$$\mathcal{D}_{\mathrm{UF}} = 1.5\,\mathrm{MSE}(x_7, \hat{x}_7) + 0.4 \sum_{t\in\{1,3,5\}} \mathrm{MSE}(x_t, \hat{x}_t) + 0.16 \sum_{t\in\{0,2,4,6\}} \mathrm{MSE}(x_t, \hat{x}_t). \tag{17}$$

The last frame $t = 7$ receives 1.5 because it is the temporal reference of the next chunk. The odd frames receive 0.4 and the even frames receive 0.16. We keep this assignment unchanged in $\mathcal{L}_A$.