JRCC: JOINT RESIDUE AND CONDITIONAL CODING FOR OVERFITTED NEURAL VIDEO COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Residue coding has long been one of the most widely used paradigms for exploiting inter-frame correlation in both conventional and learned video coding. More recently, conditional coding has emerged as a more efficient paradigm for exploiting inter-frame correlation in offline-learned video coding. This raises a natural question: can conditional coding also be effectively used in overfitted learned video coding? The key difference between overfitted and offline-learned video coding is that, in the former, the neural network model parameters must be transmitted. Therefore, using a large model to exploit conditional dependencies between neighboring frames is less suitable for the overfitted setting. In this paper, we find that directly applying conditional coding degrades rate–distortion performance under this overfitted setting, while a moderate increase in conditioning capacity can partially alleviate the degradation, but further scaling alone does not consistently resolve it. We therefore propose a joint residue-conditional coding framework that retains explicit motion-compensated prediction while using motion-aligned temporal features to condition residue synthesis. Experiments on multiple benchmark datasets under random-access (RA) and low-delay (LD) configurations show that the proposed framework consistently improves Cool-Chic 4.1, achieving average BD-rate reductions of 9.7% and 27.8%, respectively.

1 INTRODUCTION

Neural video compression (NVC) exploits temporal redundancy between video frames to reduce coding rate. Early approaches such as DVC follow the residue-coding paradigm, where a motion-compensated prediction is constructed from decoded reference frames and the remaining residue is encoded separately (Lu et al., 2019). Subsequent methods improve temporal modeling through recurrent prediction and feature-domain representations (Lin et al., 2020; Yang et al., 2020; Hu et al., 2021). More recently, DCVC introduced conditional coding as an alternative inter-frame reconstruction paradigm, where the current frame is reconstructed from latent representations conditioned on temporal context, without an explicit prediction–residue decomposition (Li et al., 2021). Subsequent conditional-coding approaches further improve temporal context modeling and coding efficiency (Li et al., 2023; 2024). Despite their strong rate–distortion performance, these offline-trained learned codecs typically rely on relatively complex decoding networks.

To reduce the decoding complexity of neural codecs, recent studies have explored overfitting-based neural compression, where a content-specific neural representation is optimized for each image or video at encoding time. Unlike offline-trained codecs, the decoder network parameters must also be transmitted as part of the bitstream, making decoder size directly relevant to the coding rate. For example, NeRV-based methods represent multiple video frames using a content-specific neural network optimized for the target sequence (Chen et al., 2021; Li et al., 2022; Chen et al., 2023; Kwan et al., 2023). To achieve competitive compression performance with low decoding complexity, subsequent works have explored more compact content-adaptive codecs (Ladune et al., 2023; Kim et al., 2024). Cool-Chic Video follows this design and performs frame-wise optimization with a lightweight decoder (Leguay et al., 2024; Ladune et al., 2025). For inter frames, Cool-Chic Video adopts a residue-coding formulation for inter-frame reconstruction.

Given the strong performance of conditional coding in offline-trained neural video compression and its effective use of temporal context, a natural question is whether it can also replace residue coding

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.

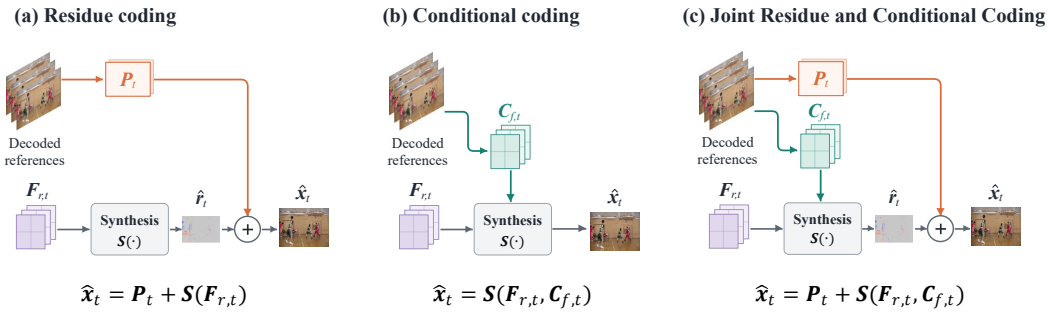


Figure 1: Comparison of residue coding, conditional coding, and the proposed JRCC framework.

in the overfitted setting. We find, however, that directly adopting conditional reconstruction degrades rate–distortion performance when the decoder is constrained to remain compact. A key challenge is that offline-trained codecs can employ high capacity conditional decoders without transmitting their model parameters, whereas overfitted codecs must transmit their frame-specific decoder parameters, making increased model capacity directly costly in bitrate.

Motivated by this limitation, we rethink how conditional coding should be formulated for compact overfitted video compression. To this end, we propose Joint Residue and Conditional Coding (JRCC), a new inter-frame reconstruction framework for compact overfitted video compression. As illustrated in Fig. 1, instead of directly reconstructing the target frame from conditional features, JRCC preserves the explicit prediction–residue decomposition and introduces motion-aligned temporal conditioning into residue synthesis. We evaluate JRCC under both random-access (RA) and low-delay (LD) configurations across multiple benchmark datasets, demonstrating consistent rate–distortion improvements over Cool-Chic 4.1.

Our main contributions are summarized as follows:

- We introduce JRCC, a new inter-frame reconstruction framework that reformulates conditional coding for compact overfitted neural video compression by preserving the explicit prediction–residue structure and conditioning residue synthesis on motion-aligned temporal features.
- We design a lightweight temporal feature-conditioning path that reuses decoded motion information for reference-feature alignment and refinement.
- Extensive experiments under both RA and LD configurations across multiple benchmark datasets demonstrate consistent rate–distortion improvements over Cool-Chic 4.1, reducing average BD-rate by 9.7% under RA and 27.8% under LD. Controlled ablation studies further validate the effectiveness of the proposed reconstruction design.

2 Related Work

2.1 Neural Video Compression

Neural video compression leverages learned transforms, motion estimation, and entropy modeling to exploit both spatial and temporal redundancies. In low-delay coding, each inter frame is coded using previously reconstructed reference frames, while inter-frame reconstruction is commonly formulated either through explicit prediction–residue coding or through learned conditional coding. DVC retains the conventional prediction–residue coding structure while jointly learning motion estimation, motion compression, motion compensation, and residue coding (Lu et al., 2019). Subsequent approaches further improve temporal modeling in low-delay settings through multiple reference frames, recurrent refinement, and scale-space flow (Lin et al., 2020; Yang et al., 2020; Agustsson et al., 2020). DCVC introduces conditional coding as an alternative inter-frame reconstruction paradigm, replacing the explicit prediction–residue decomposition with reconstruction conditioned on temporal context (Li et al., 2021). Subsequent low-delay methods further strengthen this paradigm through feature modulation with periodic refresh in DCVC-FM (Li et al., 2024),

reference-aware context modulation in DCMVC (Tang et al., 2025), and non-local correlation modeling across multiple frames in ECVC (Jiang et al., 2025). DCVC-RT further emphasizes practical low-delay deployment by reducing computational and operational overhead through implicit temporal modeling and a single low-resolution latent representation (Jia et al., 2025). Brand et al. (2024) investigate the information bottleneck affecting prediction signals in conditional coding.

Random-access coding adopts a distinct temporal structure based on hierarchical prediction and bidirectional references. DCVC-B extends contextual compression to B-frames by incorporating bidirectional motion and temporal contexts (Sheng et al., 2025). More recently, BRHVC improves the coordination of the two reference frames through bidirectional motion convergence and contextual fusion (Liu et al., 2025). These approaches primarily enhance context generation within shared, offline-trained codecs.

2.2 Overfitted Neural Video Coding

Overfitted codecs optimize content-specific representations at encoding time and transmit network parameters, latents, or both. NeRV represents a video with a neural network mapping frame indices directly to video frames (Chen et al., 2021). E-NeRV improves parameter efficiency by disentangling spatial and temporal contexts (Li et al., 2022), while HNeRV introduces content-adaptive embeddings and a redesigned decoder architecture to improve representation capacity (Chen et al., 2023). FFNeRV further incorporates optical-flow-guided frame aggregation to exploit temporal redundancy between neighboring frames (Lee et al., 2023). HiNeRV improves neural video representation through hierarchical positional encoding and a deeper lightweight architecture (Kwan et al., 2023). NVRC focuses on compressing neural representations through joint rate–distortion optimization of representation, quantization, and entropy-model parameters (Kwan et al., 2024). These approaches establish content-specific representation and model compression as important components of video coding.

Cool-Chic provides a compact overfitted design that jointly optimizes hierarchical latent grids, synthesis, and autoregressive entropy networks for each image (Ladune et al., 2023). C3 develops this approach for image and video compression through optimization and model improvements (Kim et al., 2024); related work also improves architectures and reduces encoding or decoding cost (Leguay et al., 2023; Blard et al., 2024). Cool-Chic Video extends this design to frame-wise motion-compensated coding (Leguay et al., 2024). Subsequent work further improves motion-guided optimization, sequence-level rate allocation, sub-pixel motion compensation, and temporal prediction (Leguay et al., 2025; Ladune et al., 2025). CNVC further develops motion representation, temporal context modeling, and parameter reuse (Li et al., 2026). However, conditional coding has been less explored in overfitted video compression, where decoder parameters contribute directly to the coding rate.

3 Method

JRCC is a joint residue and conditional coding framework for overfitted neural video coding, where the decoder is optimized frame by frame and its parameters are transmitted in the bitstream. JRCC retains the explicit prediction–residue reconstruction structure and introduces motion-aligned reference features as conditioning information for residue synthesis. In this way, temporal feature conditioning is introduced without replacing the prediction–residue reconstruction structure. We first present the overall framework and then detail the motion-aligned temporal feature conditioning and feature-conditioned residue synthesis.

3.1 Overall Framework

Figure 2(a) illustrates the overall decoding framework of JRCC. The motion and residue bitstreams are first processed by their respective decoders, as detailed in Fig. 2(b). In each decoder, latent grids are entropy-decoded from the corresponding bitstream and subsequently upsampled. The motion decoder output is further processed by the motion synthesis module to obtain $F_{m,t}$. From $F_{m,t}$, the local optical-flow fields v_i and, for B frames, the blending map β_t are obtained. Together with the transmitted global translations g_i, these motion quantities are used to construct the motion-

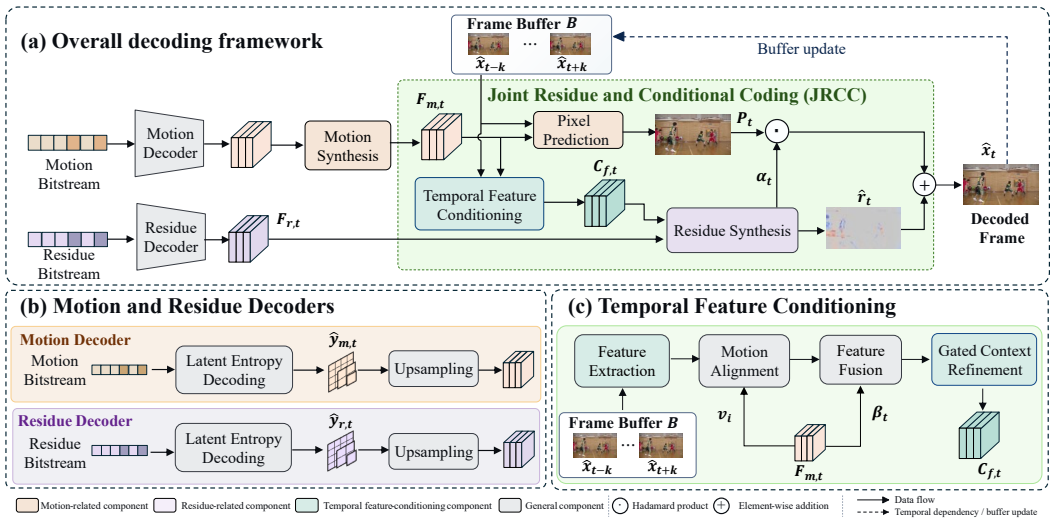


Figure 2: Overview of the proposed JRCC decoding framework. (a) Overall inter-frame decoding pipeline. (b) Motion and residue decoders, where latent grids $\widehat{y}_{m,t}$ and $\widehat{y}_{r,t}$ are entropy-decoded from their respective bitstreams and upsampled. (c) Temporal feature-conditioning path with feature extraction, motion alignment, feature fusion, and gated context refinement.

compensated prediction P_t. The same motion information is reused for temporal feature alignment and fusion.

In parallel, the residue decoder produces $F_{r,t}$, which is subsequently fed to the residue synthesis network together with $C_{f,t}$. The synthesis network generates the residue $\widehat{r}_t$ and a spatial prediction weighting map α_t, which are combined with P_t to reconstruct the current frame. The reconstructed frame is then stored in the frame buffer for subsequent inter-frame decoding. Intra frames follow the baseline intra-coding procedure.

For comparison, the reconstruction designs of residue coding, conditional coding, and JRCC are written as:

$$\begin{aligned} \text{Residue coding:} \quad & (\widehat{r}_t, \alpha_t) = S^{\mathrm{RC}}_{\widehat{\theta}_t}(F_{r,t}), && \widehat{x}_t = \widehat{r}_t + \alpha_t \odot P_t, \\ \text{Conditional coding:} \quad & \widehat{x}_t = S^{\mathrm{CC}}_{\widehat{\theta}_t}(F_{r,t}, C_{f,t}), && \\ \text{JRCC:} \quad & (\widehat{r}_t, \alpha_t) = S^{\mathrm{JRCC}}_{\widehat{\theta}_t}(F_{r,t}, C_{f,t}), && \widehat{x}_t = \widehat{r}_t + \alpha_t \odot P_t. \end{aligned} \tag{1}$$

Here, $S^{(\cdot)}_{\widehat{\theta}_t}$ denotes the frame-specific synthesis network parameterized by the quantized parameters $\widehat{\theta}_t$. Residue coding synthesizes the residue $\widehat{r}_t$ and prediction weighting map α_t from $F_{r,t}$ alone. Conditional coding instead uses $F_{r,t}$ and $C_{f,t}$ to directly synthesize the reconstructed frame $\widehat{x}_t$. JRCC preserves the prediction–residue reconstruction structure while conditioning the synthesis of $\widehat{r}_t$ and α_t on $C_{f,t}$.

3.2 Motion-Aligned Temporal Feature Conditioning

As illustrated in Fig. 2(c), the temporal feature-conditioning path consists of feature extraction, motion alignment, feature fusion, and gated context refinement. Let $\mathcal{I}_t$ denote the index set of decoded reference frames $\widehat{x}_i$. For each reference i, g_i denotes the transmitted global translation, while v_i denotes the decoded local optical-flow field toward the current frame. The prediction introduced above is given by

$$P_t = \mathcal{B}_p\Big(\{\mathcal{W}_\ell(\mathcal{W}_g(\widehat{x}_i, g_i), v_i)\}_{i\in\mathcal{I}_t}, \beta_t\Big), \tag{2}$$

where $\mathcal{W}_g$ and $\mathcal{W}_\ell$ denote global and local warping operators, respectively. For a P frame, $\mathcal{B}_p$ returns the single aligned reference. For a B frame, it blends the two aligned references using the

decoded blending map β_t. The temporal feature-conditioning path reuses the same decoded motion information and blending map for feature alignment and fusion.

Since the learnable modules in the temporal feature-conditioning path are frame-specific, their quantized parameters are also transmitted in the bitstream. We therefore adopt a lightweight design to control both model rate and decoding complexity while incorporating reference features into residue synthesis. Specifically, the same lightweight feature extractor E is applied independently to all decoded reference frames:

$$H_i = E(\widehat{x}_i), \qquad i \in \mathcal{I}_t. \tag{3}$$

The feature extractor operates on the decoded reference frames $\widehat{x}_i$, and the same frame-specific parameters are shared across all references of the current coded frame. E preserves the spatial resolution and is implemented as a single residual depthwise-convolution block with pointwise projections.

The feature-alignment stage reuses the motion information already decoded for prediction. For each reference feature H_i, the decoded global translation g_i is applied first, followed by the decoded local optical flow v_i:

$$\bar{H}_i = \mathcal{W}_g(H_i, g_i), \qquad \widetilde{H}_i = \mathcal{W}_\ell(\bar{H}_i, v_i). \tag{4}$$

Therefore, no additional motion field needs to be estimated for temporal feature conditioning.

Feature extraction is performed before motion alignment and reference fusion. The extracted reference features are first aligned to the current-frame coordinates and then fused across references. For B frames, our default design reuses the decoded blending map β_t to fuse the two aligned reference features, consistent with the prediction:

$$C_t = \begin{cases} \widetilde{H}_1, & \text{P frame}, \\ \beta_t \odot \widetilde{H}_1 + (1-\beta_t) \odot \widetilde{H}_2, & \text{B frame}. \end{cases} \tag{5}$$

The resulting feature C_t is then processed by the Gated Context Refinement module $\mathcal{R}$, which preserves the channel dimensionality and applies a spatially gated update:

$$\begin{aligned} U_t &= \mathrm{DWConv}_{3\times3}(C_t), \\ \Delta C_t &= \mathrm{GELU}(\mathrm{Conv}_{1\times1}(U_t)), \qquad A_t = \mathrm{sigmoid}(\mathrm{Conv}_{1\times1}(U_t)), \\ C_{f,t} &= \mathcal{R}(C_t) = C_t + A_t \odot \Delta C_t. \end{aligned} \tag{6}$$

Here, ΔC_t represents the learned refinement of the fused temporal feature, while A_t controls the spatial strength of this refinement. The refined feature $C_{f,t}$ is then used as the temporal condition for residue synthesis.

3.3 Feature-Conditioned Residue Synthesis

As shown in Fig. 2(a), the decoded residue feature $F_{r,t}$ and the temporal feature condition $C_{f,t}$ are jointly fed into the residue synthesis network. Before synthesis, both $F_{r,t}$ and $C_{f,t}$ are represented at the full frame resolution and concatenated along the channel dimension to form the fused feature $\widetilde{F}_{r,t}$:

$$\widetilde{F}_{r,t} = \mathrm{Concat}(F_{r,t}, C_{f,t}), \qquad (\widehat{r}_t, \alpha_t) = S^{\mathrm{JRCC}}_{\widehat{\theta}_t}(\widetilde{F}_{r,t}). \tag{7}$$

Here, $F_{r,t}$ is the decoded residue feature for the current frame, while $C_{f,t}$ provides motion-aligned temporal information extracted from the decoded reference frames. Their concatenation allows the synthesis network to jointly exploit the decoded residue feature and temporal reference information when generating $\widehat{r}_t$ and α_t.

The current frame is then reconstructed as in Eq. 1 by combining the synthesized residue with the spatially weighted prediction. In JRCC, temporal conditioning is therefore incorporated into the synthesis of $\widehat{r}_t$ and α_t, rather than being used for direct full-frame reconstruction.

3.4 Rate–Distortion Optimization

Following the overfitted codec formulation (Leguay et al., 2024; Ladune et al., 2025), each inter frame is first optimized in the continuous domain using

$$\mathcal{L}_t = D(x_t, \widehat{x}_t) + \lambda \left(R_{\mathrm{motion},t} + R_{\mathrm{res},t}\right), \tag{8}$$

where D denotes the reconstruction distortion and λ controls the rate–distortion trade-off. $R_{\mathrm{motion},t}$ and $R_{\mathrm{res},t}$ denote the estimated rates of the motion and residue latent grids $\widehat{y}_{m,t}$ and $\widehat{y}_{r,t}$, respectively. The same rate–distortion objective and optimization protocol are used for residue coding, conditional coding, and JRCC. During this stage, the latent grids and frame-specific network parameters are jointly optimized.

Following Cool-Chic 4.1, the same network-parameter quantization and coding procedure is applied to all frame-specific networks. After continuous optimization, the network parameters are quantized and transmitted together with the latent grids, and their coded sizes are included in the final bitrate. The motion bitstream contains the entropy-coded motion latent grids and the quantized parameters of the motion-related networks, while the residue bitstream contains the entropy-coded residue latent grids and the quantized parameters of the residue-related networks. For JRCC, the residue bitstream additionally includes the frame-specific parameters of the learnable modules in the temporal feature-conditioning path.

4 Experiments

4.1 Experimental Setup

Data and coding configurations. We evaluate JRCC on the HEVC common test sequences (Bossen, 2013) and the UVG dataset (Mercat et al., 2020), using HEVC Classes A, B, and E at 2560×1600, 1920×1080, and 1280×720, respectively, and UVG at 1920×1080. All sequences are evaluated in 8-bit YUV 4:2:0 format, using the first 33 frames of each sequence, following Cool-Chic Video (Leguay et al., 2024). We consider both random-access (RA) and low-delay (LD) configurations: RA uses a hierarchical B-frame structure with an intra period of 32, where frames 0 and 32 are intra coded and frames 1–31 are B frames. LD uses an intra period of -1, with only the first frame intra coded and the remaining 32 frames coded as P frames.

Comparison codecs. We compare JRCC with conventional, learned, and overfitted video codecs. The conventional codecs include x264, x265, HM-16.20, and VTM-24.0. For learned video coding, we compare against DCVC-B under the RA configuration and DCVC-FM under the LD configuration. We further include Cool-Chic 4.1 (Ladune et al., 2025) as the overfitted neural video coding baseline. DCVC-B and DCVC-FM are evaluated using the official implementations and released pretrained checkpoints under RA and LD, respectively; detailed settings are provided in Appendix F. For x264, x265, and HM-16.20, we follow the conventional codec settings used in Cool-Chic Video (Leguay et al., 2024). For VTM-24.0, we use the official Random Access and Low-Delay P configurations for RA and LD, respectively.

Model and optimization configurations. We follow the model and optimization settings of Cool-Chic Video (Leguay et al., 2024) and Cool-Chic 4.1 (Ladune et al., 2025). We retain its frame-wise decoder configurations and assignment schedule. Intra frames use the Cool-Chic 4.1 intra-coding procedure, and JRCC is applied only to inter frames. We use Adam (Kingma & Ba, 2015) with cosine learning-rate annealing following Cool-Chic Video (Leguay et al., 2024). We evaluate four rate–distortion operating points with $\lambda \in \{0.002,\ 0.0007,\ 0.0002,\ 0.00006\}$.

Evaluation metrics. We evaluate compression performance using bits per pixel (bpp), YUV 4:2:0 PSNR, and BD-rate. BD-rate is computed following the Bjøntegaard protocol (Bjontegaard, 2001), with PSNR serving as the reconstruction quality metric. Since both the input and reconstructed videos are in YUV 4:2:0 format, the distortion is computed across the Y, U, and V components as

$$\mathrm{MSE}_{\mathrm{YUV}} = \frac{4\,\mathrm{MSE}_Y + \mathrm{MSE}_U + \mathrm{MSE}_V}{6}. \tag{9}$$

For the anchor-relative codec comparisons below, the method named in each row is used as the BD-rate anchor, and JRCC is compared against it. Thus, negative values indicate that JRCC requires less rate than the corresponding anchor at matched quality. Framework ablations use the residue-coding baseline (Cool-Chic 4.1) as the BD-rate anchor.

Table 1: Anchor-relative total-rate BD-rate (%) of JRCC. Each entry is reported as RA/LD. Avg. is the mean across the four datasets. A dash indicates that the anchor was not evaluated in that configuration. Negative values indicate lower rate for JRCC.

Anchor	HEVC Class A	HEVC Class B	HEVC Class E	UVG	Avg.
x264 (medium)	-36.8/-31.9	-41.6/-32.1	-11.8/+0.8	-33.0/-23.6	-30.8/-21.7
x265 (medium)	-18.8/-18.2	-17.2/-12.6	+16.1/+32.2	-4.8/+2.8	-6.2/+1.0
HM (16.20)	-16.6/-13.7	-6.4/+6.2	+14.6/+71.4	+4.9/+23.7	-0.9/+21.9
VTM-24.0	+41.4/+26.2	+61.1/+56.0	+86.0/+159.3	+62.5/+77.2	+62.7/+79.7
DCVC-B	-0.9/–	+17.9/–	+77.6/–	+16.7/–	+27.8/–
DCVC-FM	–/+84.4	–/+111.9	–/+288.6	–/+150.2	–/+158.8
Cool-Chic 4.1	-19.5/-29.3	-10.2/-35.7	-1.5/-19.0	-7.5/-27.2	**-9.7/-27.8**

4.2 Main Results

Random access. Table 1 reports anchor-relative BD-rates for RA and LD, with each entry shown as RA/LD. Under RA, JRCC achieves an average BD-rate reduction of 9.7% over Cool-Chic 4.1 across the four datasets. Compared with conventional codecs, JRCC achieves average savings of 30.8%, 6.2%, and 0.9% relative to x264-medium, x265-medium, and HM-16.20, respectively. VTM-24.0 achieves better rate–distortion performance, while DCVC-B also outperforms JRCC on average under RA. Figure 4 shows the HEVC Class B RD curves, where JRCC reduces total-rate BD-rate by 10.2% over Cool-Chic 4.1 and outperforms x264-medium, x265-medium, and HM-16.20.

To clarify how the coding gain is distributed across rate components, Fig. 3 compares Cool-Chic 4.1 and JRCC at four operating points on HEVC Class B under RA. Across all operating points, JRCC consistently reduces the residue rate while improving reconstruction quality. Specifically, the residue-rate reductions are 8.77%, 6.25%, 6.09%, and 3.62%, accompanied by YUV 4:2:0 PSNR gains of +0.28, +0.31, +0.26, and +0.20 dB, respectively. The residue-rate saving becomes more pronounced as the operating rate increases, while the model rate varies relatively little across the operating points. Compared with the residue-coding baseline, the proposed joint framework makes more effective use of temporal reference information during residue synthesis, reducing the required residue rate while improving reconstruction quality. Overall, despite the additional model rate, JRCC achieves better rate–distortion performance.

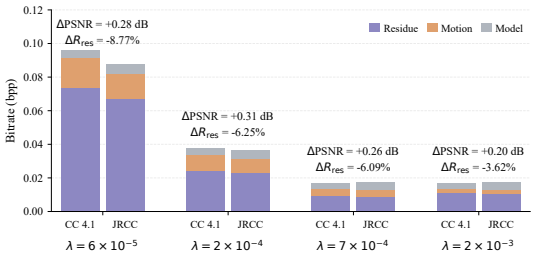


Figure 3: B-frame rate composition on HEVC Class B under RA. CC 4.1 denotes Cool-Chic 4.1. Each pair shows CC 4.1 (left) and JRCC (right), with annotations reporting the PSNR gain and residue-rate change of JRCC relative to CC 4.1. Residue and Motion denote coded latent-grid rates, while Model denotes transmitted model-parameter rate.

Low delay. We next evaluate JRCC under the LD configuration. As shown by the LD entries in Table 1, JRCC consistently improves over Cool-Chic 4.1 across all four datasets, achieving an average total-rate BD-rate reduction of 27.8%. Compared with conventional codecs, JRCC achieves an average saving of 21.7% over x264-medium and performs comparably to x265-medium. HM-16.20 and VTM-24.0 achieve better average rate–distortion performance, while DCVC-FM also remains stronger under LD. The right panel of Fig. 4 shows representative rate–distortion curves on HEVC Class B. On this dataset, JRCC reduces the total-rate BD-rate by 35.7% relative to Cool-Chic 4.1 and outperforms x264-medium and x265-medium by 32.1% and 12.6%, respectively.

4.3 Ablation Studies

Reconstruction framework. To isolate the effect of the reconstruction framework, all variants use the same architectures and configurations for corresponding components where applicable. Residue

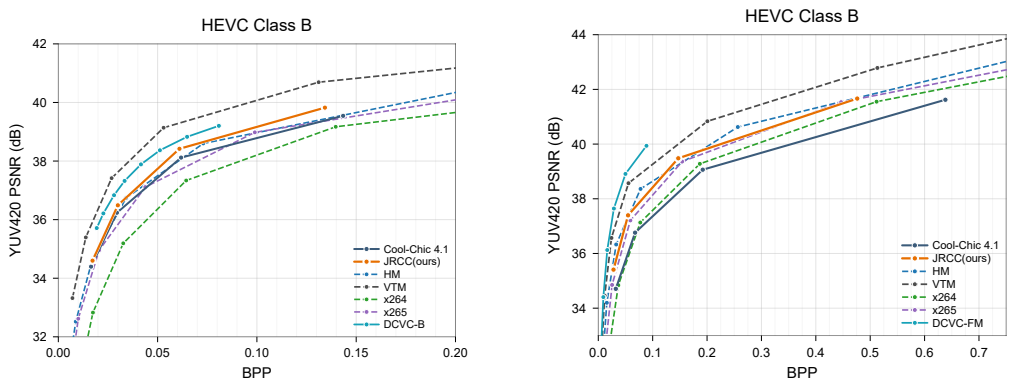


Figure 4: Rate-distortion performance on HEVC Class B under random-access (RA, left) and low-delay (LD, right) configurations.

Table 2: Reconstruction framework ablation on HEVC Class B under RA. BD-rate (%) is relative to residue coding; Avg. uses the five-sequence averaged RD curve. BDrive and BQTerr. denote BasketballDrive and BQTerrace.

Framework	BDrive	BQTerr.	Cactus	Kimono1	ParkScene	Avg. (%)
Residue coding	0.00	0.00	0.00	0.00	0.00	0.00
Conditional coding	+30.75	-4.50	+25.73	+12.44	+7.44	+14.58
JRCC	**−10.65**	**−20.25**	**−7.10**	**−6.58**	**−7.97**	**−10.20**

coding synthesizes $\widehat{r}_t$ and α_t from $F_{r,t}$ alone and reconstructs the current frame through the explicit prediction–residue structure. Conditional coding instead uses $F_{r,t}$ and $C_{f,t}$ to directly reconstruct the current frame, without an explicit prediction term in the final reconstruction. JRCC retains the explicit prediction–residue reconstruction while using $C_{f,t}$ to condition the synthesis of $\widehat{r}_t$ and α_t. As shown in Table 2, conditional coding increases the average BD-rate by 14.58% relative to residue coding, whereas JRCC reduces it by 10.20% and achieves improvements on all five sequences. These results show that directly replacing the residue coding reconstruction with conditional reconstruction is less effective in the compact overfitted setting. In contrast, the proposed JRCC framework achieves consistently better rate–distortion performance than both residue coding and conditional coding.

Conditional capacity. We examine how increasing the capacity of conditional coding affects coding efficiency in the overfitted setting, where frame-specific model parameters must be transmitted. Variants S, M, and L progressively increase the channel widths of the temporal feature-conditioning path; S corresponds to the conditional coding configuration used in Table 2. Table 3 reports the average number of frame-specific model parameters over the 31 B frames. All rate metrics are computed over the complete 33-frame sequences. Model bpp denotes the rate of transmitted model parameters. Latent-only BD-rate excludes model bits from both the tested variant and the residue-coding anchor, whereas total BD-rate includes them.

Table 3: Conditional coding with different model capacities on HEVC Class B under random access.

Variant	Params (K)	Model bpp	Latent-only	Total
S	1.21	0.00485	+12.24	+14.58
M	1.67	0.00650	+2.79	+9.99
L	2.90	0.01042	−1.05	+16.26

As the capacity increases from S to M and L, the latent-only BD-rate progressively improves from $+12.24\%$ to $+2.79\%$ and -1.05%, indicating that limited reconstruction capacity partly constrains the coding efficiency of conditional coding. However, the model bpp simultaneously increases from 0.00485 to 0.00650 and 0.01042. Consequently, the total BD-rate decreases from $+14.58\%$ for S to $+9.99\%$ for M, but increases to $+16.26\%$ for L. These results show that increasing model capacity improves the latent coding efficiency of conditional coding, but the additional cost of transmitting

Table 4: FE/CR ablation on HEVC Class B under RA. BD-rate (%) is relative to Cool-Chic 4.1; ✓/— denotes inclusion/removal. Sequence abbreviations follow Table 2.

Variant	FE	CR	BDrive	BQTerr.	Cactus	Kimono1	ParkScene	Avg.
Cool-Chic 4.1	—	—	0.00	0.00	0.00	0.00	0.00	0.00
JRCC w/o CR	✓	—	-6.95	-19.39	-5.17	-2.82	-8.55	-8.30
JRCC w/o FE	—	✓	-6.65	-17.11	-7.12	-4.99	-6.24	-8.01
JRCC	✓	✓	$\mathbf{-10.65}$	$\mathbf{-20.25}$	$\mathbf{-7.10}$	$\mathbf{-6.58}$	$\mathbf{-7.97}$	$\mathbf{-10.20}$

Table 5: Decoder architectures and decoding complexity of Cool-Chic 4.1 and JRCC. MACs/pixel is averaged over all 33 coded frames.

Method	Entropy	Upsample	Synthesis	FE	CR	MACs/pixel
Cool-Chic 4.1	$16 \to 16 \to 16 \to 2$ $8 \to 8 \to 8 \to 2$	$1 \to 1,\ 8\times8$ $1 \to 1,\ 8\times8$	$7 \to 16 \to X \to X \to X$ $5 \to 16 \to 5 \to 5$	– –	– –	831.3
JRCC (Ours)	$16 \to 16 \to 16 \to 2$ $8 \to 8 \to 8 \to 2$	$1 \to 1,\ 8\times8$ $1 \to 1,\ 8\times8$	$(7+C) \to 16 \to X \to X \to X$ $5 \to 16 \to 5 \to 5$	$3 \to C$ –	$C \to C$ –	1231.1

Notes. $I \to O$ denotes input/output channels; X/C denote synthesis output/temporal feature widths. Top/bottom rows represent residue/motion branches. FE and CR list only input/output widths.

the larger frame-specific model offsets this benefit in the overfitted setting. Together with Table 2, these results show that simply scaling conditional coding is insufficient in the compact overfitted setting, whereas the proposed JRCC framework achieves better rate–distortion performance than both residue coding and conditional coding.

Temporal-conditioning components. Here and below, FE and CR denote the Feature Extractor and Gated Context Refinement modules, respectively. In w/o FE, decoded reference frames are directly aligned and fused before the Gated Context Refinement module; in w/o CR, the fused feature C_t is directly used as $C_{f,t}$. As shown in Table 4, removing either component degrades performance relative to JRCC, although both variants still outperform Cool-Chic 4.1. With both components enabled, JRCC achieves the best average performance, reducing BD-rate by 10.20%. These results show that FE and CR provide complementary benefits for temporal feature conditioning.

4.4 Complexity Analysis

We finally analyze the decoding complexity of JRCC. Table 5 compares Cool-Chic 4.1 and JRCC using their worst-case B-frame configurations under RA. JRCC keeps the motion branch unchanged and introduces temporal feature conditioning into residue synthesis. Averaged over all 33 coded frames using their actual frame-wise configurations and warping operations, the complexity increases from 831.3 to 1231.1 MACs/pixel, mainly due to the Feature Extractor, the Gated Context Refinement module, and the increased input dimensionality of residue synthesis. Despite the additional 399.8 MACs/pixel, JRCC maintains a compact decoder while improving rate–distortion performance over Cool-Chic 4.1.

5 Conclusion

This paper studies inter-frame reconstruction for compact overfitted neural video coding, where frame-specific decoder parameters are transmitted. We propose JRCC, a new inter-frame reconstruction framework that jointly formulates the residue-coding and conditional-coding paradigms by preserving explicit prediction–residue reconstruction while conditioning residue synthesis on motion-aligned temporal information. Across multiple benchmarks, JRCC reduces average BD-rate over Cool-Chic 4.1 by 9.7% under random access and 27.8% under low delay. Ablations show that conditional coding is less suitable for the compact overfitted setting, where its coding efficiency is constrained by limited model capacity and the cost of transmitting additional model parameters. In contrast, the proposed joint residue and conditional coding framework achieves better rate–distortion performance than both residue coding and conditional coding.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and manuscript polishing, literature retrieval and discovery, and minor code editing. All research ideas, methodological design, experimental setup, results, and conclusions were developed and verified by the authors. All AI-assisted content was reviewed by the authors, who take full responsibility for the final manuscript.

REFERENCES

Eirikur Agustsson, David Minnen, Nick Johnston, Johannes Balle, Sung Jin Hwang, and George Toderici. Scale-space flow for end-to-end optimized video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *ITU SG16 Doc. VCEG-M33*, 2001.

Théophile Blard, Théo Ladune, Pierrick Philippe, Gordon Clare, Xiaoran Jiang, and Olivier Déforges. Overfitted image coding at reduced complexity. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pp. 927–931. IEEE, 2024.

Frank Bossen. Common test conditions and software reference configurations. Technical Report JCTVC-L1100, Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, 2013.

Fabian Brand, Jürgen Seiler, and André Kaup. Conditional residual coding: A remedy for bottleneck problems in conditional inter frame coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7):6445–6459, 2024. doi: 10.1109/TCSVT.2024.3359948.

Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 21557–21568. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/b44182379bf9fae976e6ae5996e13cd8-Paper.pdf.

Hao Chen, Matthew Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. Hnerv: A hybrid neural representation for videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10270–10279, June 2023.

Zhihao Hu, Guo Lu, and Dong Xu. Fvc: A new framework towards deep video compression in feature space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1502–1511, June 2021.

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, 2025. doi: 10.1109/CVPR52734.2025.01170.

Wei Jiang, Junru Li, Kai Zhang, and Li Zhang. Ecvc: Exploiting non-local correlations in multiple frames for contextual video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7331–7341, June 2025.

Hyunjik Kim, Matthias Bauer, Lucas Theis, Jonathan Richard Schwarz, and Emilien Dupont. C3: High-performance and low-complexity neural compression from a single image or video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9347–9358, June 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations*, 2015. URL https://mlanthology.org/iclr/2015/kingma2015iclr-adam/.

Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. Hinerv: Video compression with hierarchical encoding-based neural representation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 72692–72704. Curran Associates, Inc., 2023. doi: 10.52202/075280-3177. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/e5dc475c370ff42f2f96dddf8191a40c-Paper-Conference.pdf.

Ho Man Kwan, Ge Gao, Fan Zhang, Andrew Gower, and David Bull. Nvrc: Neural video representation compression. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 132440–132462. Curran Associates, Inc., 2024. doi: 10.52202/079017-4210. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/eed57814c16645298db3164829e2e45c-Paper-Conference.pdf.

Théo Ladune, Pierrick Philippe, Félix Henry, Gordon Clare, and Thomas Leguay. Cool-chic: Coordinate-based low complexity hierarchical image codec. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13515–13522, October 2023.

Théo Ladune, Thomas Leguay, Pierrick Philippe, Gordon Clare, and Félix Henry. Efficient subpixel motion compensation in learned video codecs. In *2025 Picture Coding Symposium (PCS)*, pp. 1–5. IEEE, 2025.

Joo Chan Lee, Daniel Rho, Jong Hwan Ko, and Eunbyung Park. Ffnerv: Flow-guided frame-wise neural representations for videos. In *Proceedings of the ACM International Conference on Multimedia*, pp. 7859–7870, 2023.

Thomas Leguay, Théo Ladune, Pierrick Philippe, Gordon Clare, Félix Henry, and Olivier Déforges. Low-complexity overfitted neural image codec. In *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6, 2023. doi: 10.1109/MMSP59012.2023.10337636.

Thomas Leguay, Théo Ladune, Pierrick Philippe, and Olivier Déforges. Cool-chic video: Learned video coding with 800 parameters. In *2024 Data Compression Conference (DCC)*, pp. 23–32, 2024. doi: 10.1109/DCC58796.2024.00010.

Thomas Leguay, Théo Ladune, Pierrick Philippe, and Olivier Déforges. Improved encoding for overfitted video codecs. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2025. doi: 10.1109/ISCAS56072.2025.11043596.

Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 18114–18125. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/96b250a90d3cf0868c83f8c965142d2a-Paper.pdf.

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

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

Yue Li, Chaoyi Lin, Junru Li, Kai Zhang, and Li Zhang. Cnvc: A compact neural video codec with instance-level adaptation. *IEEE Transactions on Circuits and Systems for Video Technology*, 36 (4):5525–5537, 2026. doi: 10.1109/TCSVT.2025.3640077.

Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*, pp. 267–284, Cham, 2022. Springer Nature Switzerland. ISBN 978-3-031-19833-5.

Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Yuxi Liu, Dengchao Jin, Shuai Huo, Jiawen Gu, Chao Zhou, Huihui Bai, Ming Lu, and Zhan Ma. Neural b-frame video compression with bi-directional reference harmonization. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 154369–154394. Curran Associates, Inc., 2025. doi: 10.52202/085713-5163. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/e3137c6e046fdf9f4af76fe08425f5cf-Paper-Conference.pdf.

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, 2019. doi: 10.1109/CVPR.2019.01126.

Alexandre Mercat, Marko Viitanen, and Jarno Vanne. Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *Proceedings of the 11th ACM multimedia systems conference*, pp. 297–302, 2020.

Xihua Sheng, Li Li, Dong Liu, and Shiqi Wang. Bi-directional deep contextual video compression. *IEEE Transactions on Multimedia*, 27:5632–5646, 2025. doi: 10.1109/TMM.2025.3543061.

Chuanbo Tang, Zhuoyuan Li, Yifan Bian, Li Li, and Dong Liu. Neural video compression with context modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12553–12563, June 2025.

Ren Yang, Fabian Mentzer, Luc Van Gool, and Radu Timofte. Learning for video compression with hierarchical quality and recurrent enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

A Additional Rate–Distortion Curves

Figures 5 and 6 provide additional rate–distortion curves under random access (RA) and low delay (LD), respectively. The top row shows dataset-level averaged RD curves for HEVC Classes A, B, E, and UVG, while the bottom row shows representative sequence-level results. Each panel reports YUV 4:2:0 PSNR as a function of BPP for Cool-Chic 4.1, JRCC, and the conventional or learned anchors. The RA results use the hierarchical B-frame structure, whereas the LD results use the low-delay prediction order.

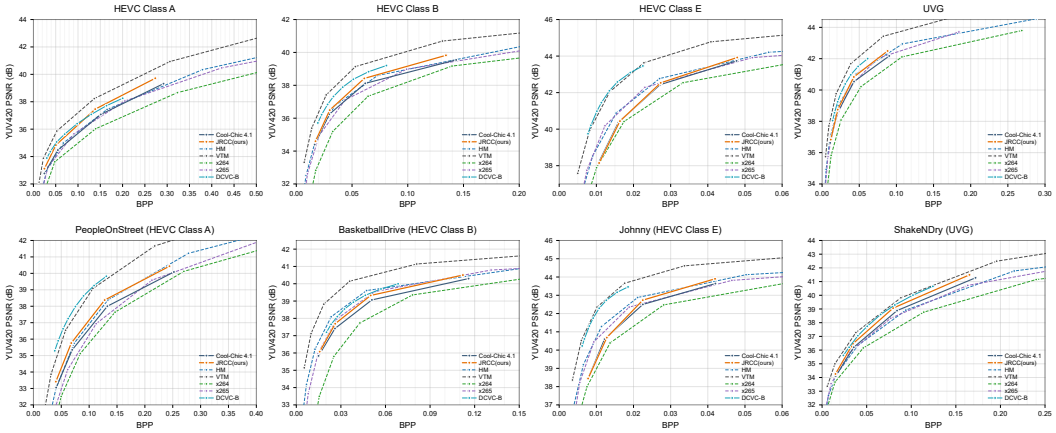


Figure 5: Additional rate–distortion curves under random access (RA). The top row shows dataset-level averaged curves for HEVC Classes A, B, E, and UVG, while the bottom row shows representative sequence-level results. The curves compare Cool-Chic 4.1, JRCC, and the conventional or learned anchors.

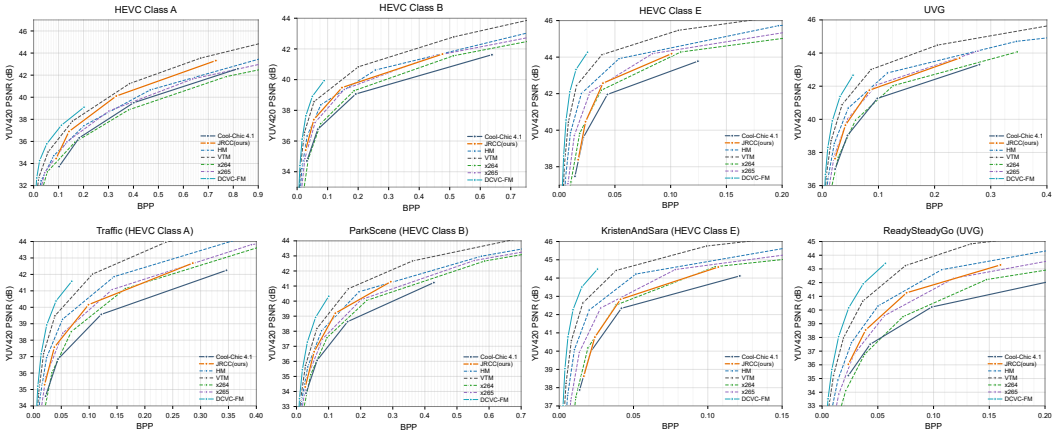


Figure 6: Additional rate–distortion curves under low delay (LD). The top row shows dataset-level averaged curves for HEVC Classes A, B, E, and UVG, while the bottom row shows representative sequence-level results. The curves compare Cool-Chic 4.1, JRCC, and the conventional or learned anchors.

In the top-row dataset-level curves, JRCC generally achieves lower BPP than Cool-Chic 4.1 at comparable PSNR under both RA and LD. The representative sequence-level curves in the bottom row further demonstrate that the rate–distortion improvements extend across diverse video content, although the magnitude of the gain varies across sequences and operating points. The consistent trend under both RA and LD indicates that the proposed framework is effective across different prediction structures. The BD-rate values reported in the main text are computed from the dataset-level averaged RD curves.

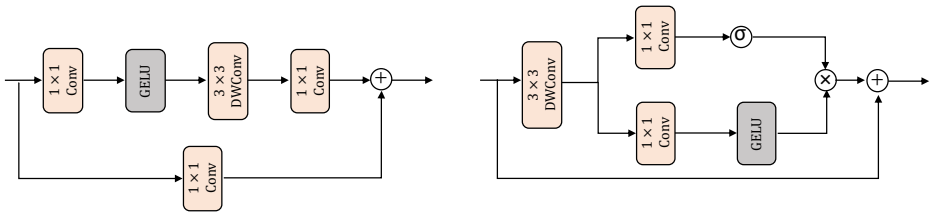
B Detailed Module Architectures

The detailed architectures of the two learned modules in the temporal conditioning path are shown in Fig. 7. Both modules operate on reference information already available to the decoder and form the learnable components of the temporal conditioning path.

B.1 Temporal Feature-Conditioning Path

Following the internal representation of Cool-Chic 4.1, decoded reference frames are represented as full-resolution three-channel YUV tensors before feature extraction. For each decoded reference frame, the Feature Extractor maps this three-channel representation into the temporal feature space with a 1×1 projection and GELU activation. A replicate-padded 3×3 depthwise convolution captures local structure, followed by a 1×1 output projection. When the input and output widths differ, a 1×1 skip adaptor matches the dimensions of the residual shortcut. The same extractor is applied independently to each reference before motion alignment, preserving a multi-channel representation for temporal conditioning.

The Gated Context Refinement module $R(\cdot)$ uses a depthwise convolution and two pointwise branches to predict a sigmoid gate and a GELU feature update. Their element-wise product is added to the input to form the temporal condition for residue synthesis.



(a) Feature Extractor (b) Gated Context Refinement

Figure 7: Detailed architectures of the temporal conditioning modules. (a) Feature Extractor. Each decoded reference is processed independently by a pointwise projection, GELU activation, depthwise convolution, and pointwise output projection, together with a 1×1 skip projection. (b) Gated Context Refinement. A depthwise convolution is followed by two pointwise branches that predict a gate and a feature update, which are multiplicatively combined and added to the input feature.

C Additional Ablation Studies

C.1 Temporal Feature Width

We study the effect of the temporal feature width C in the temporal feature-conditioning path. We evaluate $C \in \{6, 8, 12\}$ while keeping all other settings fixed. At each λ, bpp and YUV 4:2:0 PSNR are averaged over the five HEVC Class B sequences before computing the dataset-level BD-rate from the four averaged RD points.

Table 6: Effect of temporal feature width C on five HEVC Class B sequences under random access. Sequence columns report per-sequence BD-rate (%) relative to the residue-coding baseline; the Avg. column is computed from the four dataset-average RD points. Sequence labels follow the abbreviations in Table 2.

C	BDrive	BQTerr.	Cactus	Kimono1	ParkScene	Avg. (%)
6	-7.24	-18.81	-6.55	-5.86	-7.65	-8.97
8 **(JRCC)**	**-10.65**	**-20.25**	**-7.10**	**-6.58**	**-7.97**	**-10.20**
12	-8.66	-20.24	-6.73	+0.88	-7.65	-8.19

The $C = 8$ configuration achieves the best average BD-rate. The $C = 6$ and $C = 12$ configurations trail it by 1.23 and 2.01 percentage points, respectively.

C.2 Temporal Feature Fusion

For B frames, the default JRCC implementation fuses the two aligned reference features with the decoded blending map β_t. We compare this choice with direct feature concatenation while keeping the feature extractor, motion alignment, Gated Context Refinement module, and synthesis interface fixed. Both variants use residue coding as the anchor.

Table 7: Temporal feature-fusion ablation on HEVC Class B under random access. Entries are BD-rate (%) relative to the residue-coding anchor; negative values indicate bitrate savings. The two rows differ only in the fusion operation before context refinement.

Fusion strategy	BDrive	BQTerr.	Cactus	Kimono1	ParkScene	Avg. (%)
Feature concatenation	-6.63	**-21.67**	-6.46	-4.92	**-9.43**	-9.46
β-weighted fusion (JRCC)	**-10.65**	-20.25	**-7.10**	**-6.58**	-7.97	**-10.20**

The β-weighted fusion achieves a better dataset-average BD-rate and outperforms direct concatenation on BasketballDrive, Cactus, and Kimono1. Direct concatenation performs better on BQTerrace and ParkScene, but its dataset-level BD-rate reduction is 0.74 percentage points smaller than that of the default β-weighted fusion.

D Rate and Complexity Analysis

D.1 Module-wise Decoding Complexity

Table 8 compares module-wise decoding costs for the base Cool-Chic 4.1 and JRCC decoders. The reported values are averaged over all 33 coded frames using their actual frame-wise configurations and warping operations. Context feature warping denotes motion-aligned warping in the temporal-conditioning branch, whereas prediction warping denotes conventional motion compensation. Final output resizing denotes the nearest-neighbor resizing of the residue and motion synthesis outputs.

Table 8: Module-wise decoding complexity of the base Cool-Chic 4.1 decoder and the JRCC decoder. Values are averaged over all 33 coded frames under RA (31 B frames and two intra frames) using their actual frame-wise configurations. "–" denotes a module not present in Cool-Chic 4.1. Totals are computed from unrounded values.

(a) JRCC

Module	Avg. MAC/pixel
Residue ARM	199.105
Residue upsampling	64.761
Residue synthesis	422.424
Feature Extractor	191.758
Gated Context Refinement	72.242
Context feature warping	75.636
Motion ARM	6.421
Motion upsampling	2.927
Motion synthesis	18.059
Prediction warping	169.091
Final output resizing	8.636
Total average	**1231.059**

(b) Cool-Chic 4.1

Module	Avg. MAC/pixel
Residue ARM	199.105
Residue upsampling	64.761
Residue synthesis	362.303
Feature Extractor	–
Gated Context Refinement	–
Context feature warping	–
Motion ARM	6.421
Motion upsampling	2.927
Motion synthesis	18.059
Prediction warping	169.091
Final output resizing	8.636
Total average	**831.302**

E QUALITATIVE RESULTS

E.1 QUALITATIVE RECONSTRUCTION COMPARISONS

Figures 8 and 9 show representative qualitative reconstruction comparisons under RA and LD. The red boxes highlight high-contrast regions for enlarged visual comparison.

Under RA, JRCC exhibits fewer visible block-like artifacts and more spatially coherent reconstructions than Cool-Chic 4.1 at the shown operating points. The improvements are particularly visible around object boundaries and in relatively smooth regions, while JRCC also achieves higher PSNR at slightly lower bitrate. In the LD example, JRCC reaches 35.14 dB at 0.048 bpp, compared with 33.39 dB at 0.099 bpp for Cool-Chic 4.1. The enlarged region shows fewer block-like distortions and more continuous structures around the railing and object boundaries. These qualitative observations are consistent with the larger quantitative gains under LD and the improvements observed under RA.



Figure 8: Qualitative reconstruction comparisons under random access on Jockey and PeopleOnStreet. Each panel shows the original frame, the Cool-Chic 4.1 reconstruction, and the JRCC reconstruction; the red boxes mark the enlarged high-contrast regions.



Figure 9: Qualitative reconstruction comparison under low delay on BQTerrace. The original frame, Cool-Chic 4.1 reconstruction, and JRCC reconstruction are shown from left to right; the red box marks the enlarged lamp and railing region.

F Implementation Details

F.1 Conventional Codec Configuration

The experiments use the first 33 frames of every sequence in YUV 4:2:0 format. HEVC Classes A, B, and E use resolutions of 2560×1600, 1920×1080, and 1280×720, respectively, while the UVG sequences use 1920×1080. For RA, frames 0 and 32 are intra coded and frames 1–31 follow the hierarchical B-frame structure. For LD, only frame 0 is intra coded and the remaining 32 frames are coded as P frames.

For x264 and x265, we use the medium preset and follow the codec settings used in Cool-Chic Video (Leguay et al., 2024). For HM-16.20, we use the official random-access and low-delay P configuration files, `encoder_randomaccess_main.cfg` and `encoder_lowdelay_P_main.cfg`, respectively. The RA configuration uses $\mathrm{IntraPeriod} = 32$, while the LD configuration uses $\mathrm{IntraPeriod} = -1$.

For VTM-24.0, we use the official random-access and low-delay P configuration files, `encoder_randomaccess_vtm.cfg` and `encoder_lowdelay_P_vtm.cfg`, respectively. The RA configuration uses $\mathrm{GOPSize} = 32$ and $\mathrm{IntraPeriod} = 32$, while the LD configuration uses $\mathrm{IntraPeriod} = -1$.

F.2 Learned Codec Baselines

For DCVC-B and DCVC-FM, we use the official implementations and released pretrained checkpoints. DCVC-B is evaluated under RA and DCVC-FM under LD using the first 33 frames in 8-bit YUV 4:2:0 at the native sequence resolutions. Pretrained model parameters are excluded from the reported bitrate; bpp and YUV 4:2:0 PSNR are computed from the coded sequences, and BD-rate is computed using the same Bjøntegaard procedure as for our method.