A Cross-Framework Study of Temporal Information Buffering Strategies for Learned Video Compression

Kuan-Wei Ho* Yi-Hsin Chen* Martin Benjak† Jörn Ostermann† Wen-Hsiao Peng*
*National Yang Ming Chiao Tung University, Taiwan †Leibniz Universität Hannover, Germany

***Abstract*—Recent advances in learned video codecs have demonstrated remarkable compression efficiency. Two fundamental design aspects are critical: the choice of inter-frame coding framework and the temporal information propagation strategy. Inter-frame coding frameworks include residual coding, conditional coding, conditional residual coding, and masked conditional residual coding, each with distinct mechanisms for utilizing temporal predictions. Temporal propagation methods can be categorized as explicit, implicit, or hybrid buffering, differing in how past decoded information is stored and used. However, a comprehensive study covering all possible combinations is still lacking. This work systematically evaluates the impact of explicit, implicit, and hybrid buffering on coding performance across four inter-frame coding frameworks under a unified experimental setup, providing a thorough understanding of their effectiveness.**

***Index Terms*—Learned video compression, hybrid temporal information buffering, residual coding, conditional coding, conditional residual coding, masked conditional residual coding.**

I. Introduction

Learned video codecs [1], [2] achieve promising coding performance by leveraging neural networks to model temporal dependencies across frames and perform compression using neural inter-frame codecs. As illustrated in Fig. 1, two fundamental aspects are among the most critical in their development: the design of the inter-frame coding framework (highlighted in blue) and the mechanism of temporal information propagation (highlighted in red).

The design of the inter-frame coding framework focuses on how the temporal prediction x_c can be effectively utilized to assist the coding of the current frame x_t. Recent inter-frame coding frameworks can be categorized into four types: residual coding, conditional coding, conditional residual coding, and masked conditional residual coding. Residual coding [3], [4] encodes the residue $x_t - x_c$, while conditional coding [1], [2] enables nonlinear use of x_c by directly encoding x_t, with x_c serving as a condition signal for both the inter-frame encoder and decoder. While conditional coding generally outperforms residual coding, it can encounter an information bottleneck issue [5], where information carried by x_c is potentially lost during neural network processing of x_c. To alleviate this issue, conditional residual coding [6] combines residual coding and conditional coding by encoding the residue $x_t - x_c$ with a conditional inter-frame codec. Under the assumption that $H(x_t - x_c) \leq H(x_t)$, conditional residual coding achieves better coding performance. However, this assumption may not hold in local regions with dis-occlusion or unreliable motion estimates. To mitigate the sub-optimal performance caused by such violations, masked conditional residual coding [7] employs a soft mask to blend conditional coding and conditional residual coding at the pixel level.

This work is supported by the National Science and Technology Council (NSTC), Taiwan, under Grants NSTC 113-2634-F-A49-007- and NSTC 114-2221-E-A49-035-MY3. We thank the National Center for High-performance Computing (NCHC) for providing computational and storage resources.

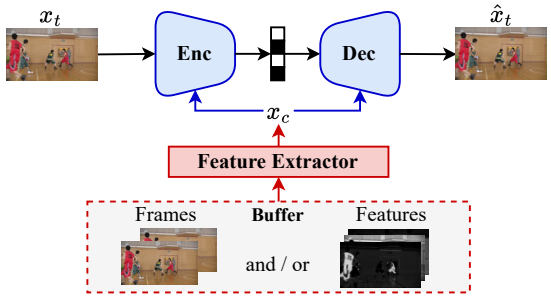

Fig. 1. Two fundamental aspects in the design of learned video codecs: the inter-frame coding framework (highlighted in blue) and the mechanism of temporal information propagation (highlighted in red).

TABLE I
Evaluated combinations of coding frameworks and buffering strategies in prior works with cross-variant discussion. This work evaluates all combinations.

Buffering Strategies	RC	CC	CRC	MCR
Explicit	[6], [7]	[6]–[9]	[6]–[9]	[7], [8], [10]
Implicit	-	-	[9]	[10]
Hybrid	-	[9]	[9]	[10]

Temporal propagation in learned video codecs focuses on effectively utilizing information produced during previous decoding process to formulate the temporal prediction x_c. Based on the nature of the buffered signals, recent works can be categorized into three buffering strategies: explicit, implicit, and hybrid. Explicit buffering [3]–[8], [11], [12] stores the previously decoded frame $\hat{x}_{t-1}$ as a reference, relying on its strong correlation with the current coding frame x_t. However, the decoded frames must simultaneously approximate the input accurately and summarize useful information from the past. Implicit buffering [1], [2] instead stores a large number of high-dimensional intermediate features during the decoding process, avoiding this dual constraint but often resulting in non-compact representations. Some models [1], [2] store more than 48 high-resolution feature maps, leading to high memory bandwidth demands during decoding. Hybrid buffering [9], [10] combines explicit and implicit mechanisms by storing $\hat{x}_{t-1}$ as the primary reference along with a few compact complementary features, reducing overall buffer size requirements.

As shown in Fig. 1, the inter-frame coding framework and

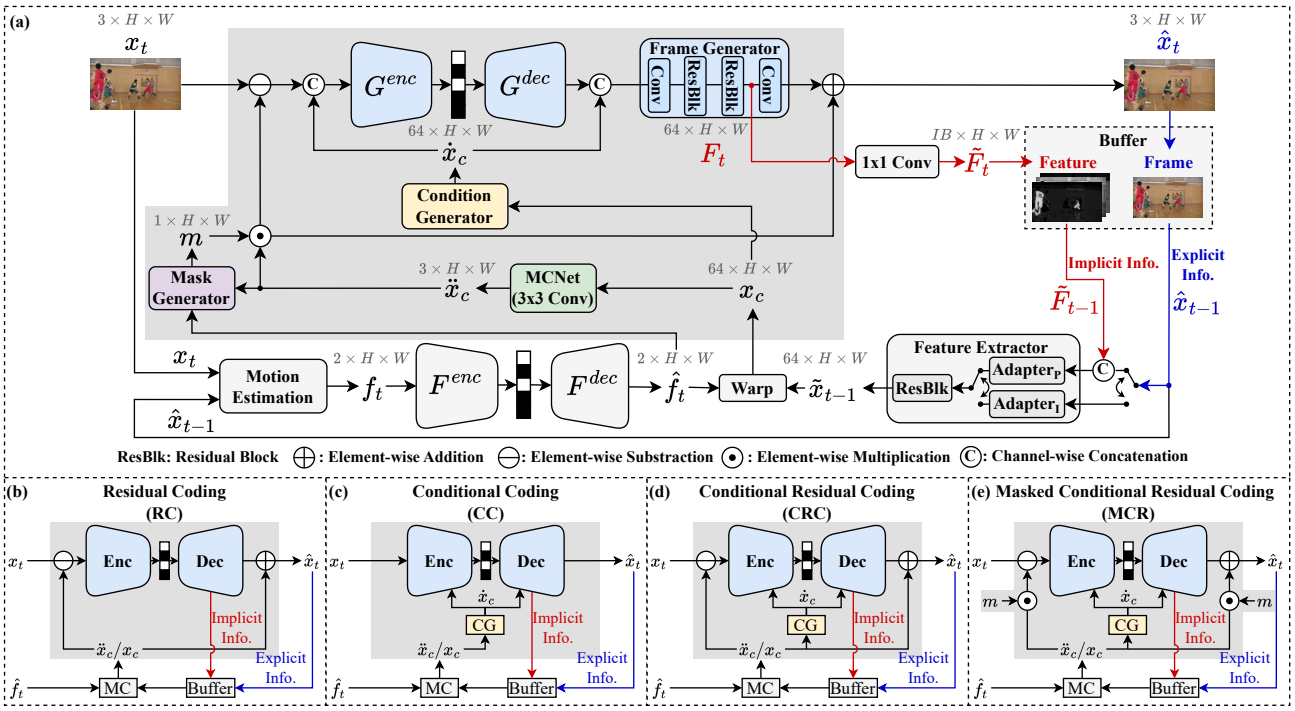

Fig. 2. System Overview. (a): the architecture of the masked conditional residual coding framework with hybrid buffering used in our study. (b)–(e): the architectural variations across different coding frameworks.

temporal propagation mechanism are two orthogonal design choices and can be combined flexibly. Table I shows that while some studies [6]–[10] explore partial combinations of different coding frameworks and temporal propagation strategies, none provides a comprehensive analysis of all four inter-frame coding frameworks across the three temporal buffering methods. In this work, we systematically investigate how explicit, implicit, and hybrid temporal buffering strategies impact coding performance across various inter-frame coding frameworks using a unified backbone to ensure a fair comparison.

II. Temporal Buffering Strategies across Coding Frameworks

Fig. 2(a) illustrates the architecture of the masked conditional residual coding (MCR) framework with hybrid buffering used in our study, which is slightly adapted from [8], [9], [11]. It consists of (1) a motion estimation network to estimate the motion between the reference frame $\hat{x}_{t-1}$ and the current coding frame x_t; (2) a motion codec $\{F^{enc}, F^{dec}\}$ to encode the estimated flow map f_t; (3) a feature extractor, comprising a switchable adapter (either a 3×3 convolution for Adapter$_I$ or a 1×1 convolution for Adapter$_P$) followed by a residual block, to extract features from explicit and implicit references; (4) an inter-frame coding module, highlighted with a gray background (including MCNet, Mask Generator, Condition Generator, an inter-frame codec $\{G^{enc}, G^{dec}\}$, and Frame Generator) to encode the coding frame x_t using the warped feature-domain temporal prediction x_c as reference; and (5) a 1×1 convolution to adapt the channel size of the intermediate features during decoding, F_t, for implicit buffering. In our study, we adjust the channel size IB of this convolutional layer to examine the impact of buffer size on coding performance.

To ensure a fair comparison among different buffering strategies and inter-frame coding framework variants, we minimally modify this framework according to each variant.

A. Inter-frame Coding Frameworks

In this work, we study four inter-frame coding frameworks: residual coding (RC), conditional coding (CC), conditional residual coding (CRC), and masked conditional residual coding (MCR). According to each framework, the inter-frame coding module with the gray background in Fig. 2(a) is replaced as illustrated in Fig. 2(b) to Fig. 2(e).

Residual coding (RC): As depicted in Fig. 2(b), RC encodes the residue $x_t - \ddot{x}_c$ between the coding frame x_t and its pixel-domain temporal prediction $\ddot{x}_c$. Since it does not require a conditioning signal for inter-frame coding nor a mask, Condition Generator and Mask Generator in Fig. 2(a) are removed.

Conditional coding (CC): As depicted in Fig. 2(c), CC encodes the coding frame x_t with $\dot{x}_c$ serving as the conditioning signal. Since it does not require a pixel-domain temporal prediction $\ddot{x}_c$ nor a mask, MCNet and Mask Generator in Fig. 2(a) are removed.

Conditional residual coding (CRC): As depicted in Fig. 2(d), CRC encodes the residue $x_t - \dot{x}_c$ with $\dot{x}_c$ serving as the conditioning signal. Since it does not require a mask, Mask Generator in Fig. 2(a) is removed.

Masked conditional residual coding (MCR): As depicted in Fig. 2(a) and (e), MCR encodes the residue $x_t - m \odot \ddot{x}_c$, where $\dot{x}_c$ serves as the conditioning signal and m is a pixel-wise soft mask ranging from 0 to 1. The mask m is generated using the decoded flow map $\hat{f}_{t-1}$ and the pixel-domain temporal predictor $\ddot{x}_c$, following [7], [8], [10].

TABLE II
BD-RATE (%) COMPARISON WITH HM [13] IN TERMS OF PSNR-RGB, USING RESIDUAL CODING WITH EXPLICIT BUFFERING AS THE ANCHOR.

Variants (Buffer size)	UVG	HEVC-B	HEVC-C	HEVC-D	HEVC-E	HEVC-RGB	MCL-JCV	Average
RC, Explicit (3+0)	0	0	0	0	0	0	0	0
RC, Implicit (0+67)	5.6	15.3	20.2	29.6	55.2	18.1	11.8	22.3
RC, Implicit (0+6)	12.3	15.2	33.8	35.4	92.4	22.2	27.0	34.0
RC, Hybrid (3+64)	-5.7	-0.3	3.8	1.2	11.3	-0.1	0.6	1.5
RC, Hybrid (3+3)	-4.0	-1.4	2.2	1.3	10.9	2.7	1.6	1.9
CC, Explicit (3+0)	-11.3	-9.3	-12.0	-18.2	-8.2	-7.9	-11.1	-11.1
CC, Implicit (0+67)	-22.4	-19.7	-21.5	-27.5	-25.2	-20.2	-20.9	-22.5
CC, Implicit (0+6)	-20.2	-17.6	-21.5	-25.4	-23.3	-16.9	-18.0	-20.4
CC, Hybrid (3+64)	-24.4	-20.7	-23.0	-28.4	-26.5	-20.7	-21.7	-23.6
CC, Hybrid (3+3)	-21.7	-18.6	-22.0	-27.3	-23.8	-18.8	-19.0	-21.6
CRC, Explicit (3+0)	-12.6	-11.6	-14.7	-19.2	-11.3	-10.8	-14.4	-13.5
CRC, Implicit (0+67)	-28.5	-22.8	-24.6	-30.0	-28.6	-23.8	-25.2	-26.2
CRC, Implicit (0+6)	-23.3	-18.4	-22.8	-26.9	-23.5	-18.2	-21.6	-22.1
CRC, Hybrid (3+64)	-31.0	-25.1	-26.7	-31.8	-32.2	-25.7	-27.5	-28.6
CRC, Hybrid (3+3)	-29.4	-23.3	-24.5	-28.1	-30.5	-23.5	-26.8	-26.6
MCR, Explicit (3+0)	-21.9	-16.3	-18.5	-23.2	-21.0	-16.9	-20.2	-19.7
MCR, Implicit (0+67)	-28.1	-23.3	-26.2	-30.8	-29.9	-24.5	-25.4	-26.9
MCR, Implicit (0+6)	-24.5	-20.3	-24.0	-29.0	-24.1	-20.7	-21.2	-23.4
MCR, Hybrid (3+64)	-31.3	-26.7	-28.4	-33.8	-33.4	-28.4	-28.9	-30.1
MCR, Hybrid (3+3)	-29.2	-24.5	-26.0	-30.8	-32.1	-26.1	-26.4	-27.9
HM 16.25 [13]	-32.5	-25.8	-39.4	-31.0	-45.1	-22.2	-31.9	-32.6

B. Temporal Buffering Strategies

In this work, we study three temporal buffering strategies: explicit, implicit, and hybrid buffering. The signal stored for temporal propagation in each strategy is described below:

- **Explicit buffering:** the decoded frame $\hat{x}_t$ as an explicit reference.
- **Implicit buffering:** the processed latent features $\tilde{F}_t$ as an implicit reference, where $\tilde{F}_t$ is obtained by applying a 1×1 convolution to the intermediate feature F_t from the decoder before reconstruction, following [9].
- **Hybrid buffering:** both $\hat{x}_t$ as an explicit reference and $\tilde{F}_t$ as an implicit reference.

To analyze the impact of buffer size, we consider two settings for the implicit and hybrid variants: a large buffer with 67 channels and a small buffer with 6 channels. For explicit buffering, the buffer stores only the decoded frame, which has a fixed size of 3 channels.

III. EXPERIMENTS

A. Experimental Setup

Training Details: We train our models on the Vimeo-90k dataset [14], using randomly cropped 256×256 patches. All model variants adopt the same training procedure as described in [9]. Following [8], [9], we train four models with $\lambda \in \{256, 512, 1024, 2048\}$ to target different rate-distortion trade-offs, where the hyperparameter λ balances rate and distortion in minimizing the rate-distortion loss $\lambda D + R$. Distortion is measured as mean squared error in the RGB domain.

Evaluation Methodologies: We evaluate our models on standard datasets: UVG [15], HEVC Class B–E [16], HEVC-RGB [17], and MCL-JCV [18]. Following [1], YUV420 sequences are converted to RGB444 via BT.709 [19], except HEVC-RGB. Each sequence is encoded for the first 96 frames with an intra-period of 32. Distortion and rate are measured using PSNR-RGB and bits-per-pixel (bpp), respectively. BD-rate is computed via piecewise cubic interpolation [20], where positive and negative values indicate bit rate increase and reduction, respectively. Dataset-level BD-rate is obtained by averaging per-sequence BD-rates [13], [21], [22].

B. Rate-Distortion Performance

Table II reports BD-rate results for four inter-frame coding frameworks (RC, CC, CRC, MCR) under three different buffering strategies (Explicit, Implicit, Hybrid). For implicit and hybrid buffering, we also evaluate variants with different buffer sizes. HM 16.25 [13], configured with the same low-delay-B setting as [10], is also reported as the baseline traditional codec. The following observations are made.

(1) *Effectiveness of implicit feature propagation except in residual coding*: For all frameworks except residual coding (RC), propagating intermediate features $\tilde{F}_t$ during the decoding process either solely (implicit) or together with the previously decoded frame (hybrid) outperforms using only the decoded frame (explicit). This indicates that feature propagation provides greater flexibility for temporal information propagation. RC is an exception because it encodes the residue $x_t - x_c$, so its intermediate features contain only residual information rather than rich frame content. As a result, these features provide limited contextual information for temporal prediction. This is supported by the observation that hybrid buffering, which combines explicit information, performs significantly better than implicit alone in RC. However, hybrid still performs worse than explicit in most RC cases, likely because the residual features $\tilde{F}_t$ in RC can be less effective than explicit information, $\hat{x}_{t-1}$. When the two temporal information sources are not fused well, the inclusion of $\tilde{F}_t$ may hinder performance, causing hybrid buffering to underperform compared

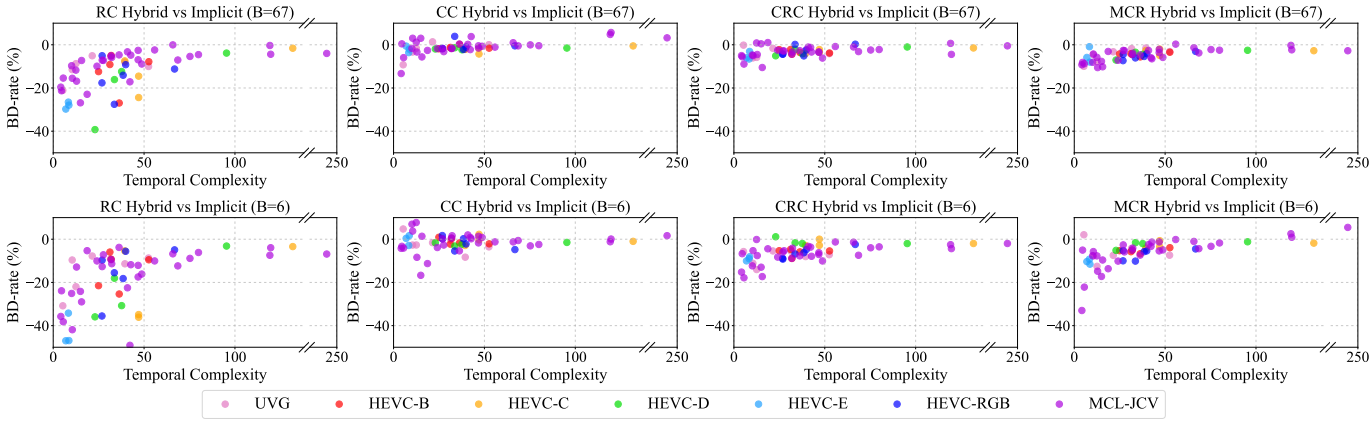

Fig. 3. Per-sequence BD-rate versus temporal complexity analysis across four coding frameworks under large (B=67) and small (B=6) buffer settings. Each point represents the BD-rate of hybrid buffering using each framework's implicit buffering variant serving as the anchor. Temporal complexity is computed using the video complexity analyzer from [23]. Negative BD-rates suggest bitrate savings. The lower the BD-rate, the better the compression performance.

to using explicit buffering alone. In contrast, although CRC and MCR also encode residues, their use of conditional signals enriches $\tilde{F}_t$ with contextual information, enabling hybrid and implicit buffering to surpass explicit buffering.

(2) *Consistent superiority of hybrid buffering and performance hierarchy among coding frameworks*: As expected, hybrid buffering consistently outperforms implicit buffering across all coding frameworks and buffer sizes. Moreover, a clear performance hierarchy of coding frameworks is observed across buffering mechanisms, with MCR outperforming CRC, CRC outperforming CC, and CC outperforming RC. It is important to note that, as shown in Table I, prior works [6]–[10] conduct cross-framework studies only on subsets of frameworks, and [6]–[8] limit the analysis to explicit buffering. None of these prior works offers a comprehensive comparison across both coding frameworks and buffering mechanisms as this paper. Compared to the traditional codec HM [13], MCR hybrid buffering variants perform closer to HM, though a performance gap remains. This gap mainly stems from using a simple but typical learned codec as the common code base for systematic analysis. Adopting more efficient video compression architectures can further improve performance.

(3) *Hybrid buffering is more robust to buffer size reduction; CC degrades the least among implicit variants*: The performance degradation when reducing buffer size is greater in implicit buffering than in hybrid buffering, with BD-rate increases of 11.7%, 2.1%, 4.1%, and 3.5% for implicit buffering, compared to 0.4%, 2.0%, 2.0%, and 2.2% for hybrid buffering in RC, CC, CRC, and MCR, respectively. This highlights the robustness of hybrid buffering under constrained memory. Notably, conditional coding (CC) exhibits the smallest degradation (2.1%) among all implicit buffering variants, likely because CC encodes the intra-frame x_t, which contains abundant content; therefore, the intermediate features $\tilde{F}_t$ during decoding naturally retain richer contextual information than those in frameworks encoding only residue ($x_t - x_c$ or $x_t - m \odot x_c$), making CC less sensitive to buffer size reduction.

(4) *Increasing advantage of hybrid buffering in advanced coding frameworks*: Interestingly, as the framework transitions from CC to CRC and then MCR, the performance gap between hybrid buffering and implicit buffering becomes progressively larger. Under both large and small buffer settings, the BD-rate improvement of hybrid over implicit buffering increases from 1.1-1.2% in CC to 2.4-4.5% in CRC, and to 3.2-4.5% in MCR. These results indicate that more advanced coding frameworks derive greater benefit from combining explicit and implicit temporal references, regardless of buffer size. This is because CRC (or MCR) encode the residue $x_t - x_c$ (or $x_t - m \odot x_c$), so the extracted features $\tilde{F}_t$ carry less contextual information than those derived from encoding the intra-frame x_t in CC, leading to weaker performance in their implicit variants.

C. Per-Sequence Comparison

Fig. 3 shows how the BD-rate savings of hybrid buffering over implicit buffering across coding frameworks correlate with the temporal complexity of each test sequence. Hybrid buffering outperforms implicit buffering across most sequences and frameworks. However, some cases in CC exhibit BD-rate values greater than zero, meaning implicit buffering performs better than hybrid buffering. This may be because, as discussed in Section III-B, the propagated feature $\tilde{F}_t$ in CC retains richer contextual information than those in RC, CRC, and MCR. When implicit and explicit information are not well fused, due to the use of a simple 1×1 convolution in Adapter$_P$, CC's implicit buffering alone can provide sufficient information. Nonetheless, in most CC cases, hybrid buffering still yields better results. Moreover, Table II shows that CC is not the best framework, as more advanced frameworks combined with hybrid buffering achieve superior coding performance.

In addition, hybrid buffering shows greater BD-rate savings on sequences with lower temporal complexity, likely because it uses the previously decoded frame $\hat{x}_{t-1}$, which has the highest correlation with the current coding frame x_t, offering a good temporal reference in lower temporal complexity scenarios.

D. Complexity Analysis

Table III presents a detailed comparison of the BD-rate and three platform-independent complexity metrics: encoding kMACs/pixel, decoding kMACs/pixel, and model size,

TABLE III
COMPARISON OF THE BD-RATE AND COMPLEXITY IN TERMS OF THE ENCODING/DECODING MACS AND MODEL SIZE.

Variants (Buffer size)	BD-rate (%)	Enc. / Dec. kMACs/pixel	Model Size (M)
RC, Explicit (3+0)	0	1029 / 661	7.266
RC, Implicit (0+67)	22.3	1040 / 672	7.279
RC, Hybrid (3+64)	1.5	1040 / 672	7.278
RC, Implicit (0+6)	34.0	1029 / 661	7.267
RC, Hybrid (3+3)	1.9	1029 / 661	7.267
CC, Explicit (3+0)	-11.1	1162 / 768	7.944
CC, Implicit (0+67)	-22.5	1169 / 775	7.953
CC, Hybrid (3+64)	-23.6	1169 / 775	7.953
CC, Implicit (0+6)	-20.4	1162 / 768	7.945
CC, Hybrid (3+3)	-21.6	1162 / 768	7.945
CRC, Explicit (3+0)	-13.5	1164 / 770	7.946
CRC, Implicit (0+67)	-26.2	1171 / 777	7.955
CRC, Hybrid (3+64)	-28.6	1171 / 777	7.955
CRC, Implicit (0+6)	-22.1	1164 / 770	7.947
CRC, Hybrid (3+3)	-26.6	1164 / 770	7.947
MCR, Explicit (3+0)	-19.7	1221 / 827	8.169
MCR, Implicit (0+67)	-26.9	1228 / 834	8.177
MCR, Hybrid (3+64)	-30.1	1228 / 833	8.177
MCR, Implicit (0+6)	-23.4	1221 / 827	8.169
MCR, Hybrid (3+3)	-27.9	1221 / 827	8.169

for different buffering strategies across four inter-frame coding frameworks. Across all coding frameworks, hybrid buffering maintains nearly the same encoding and decoding kMACs/pixel and model size to implicit buffering, with increases typically within 1% in MACs and 0.1% in model size compared to the explicit baseline. This minimal overhead is primarily due to the lightweight design of hybrid buffering modules, which introduces only a 1×1 convolution to adapt the buffer channel size, keeping the additional computational cost low. Notably, under small-buffer setting, hybrid buffering achieves substantial BD-rate improvements of 10.5%/1.2% in CC, 13.1%/4.5% in CRC, and 8.2%/4.5% in MCR over their respective explicit/implicit variants, while maintaining nearly the same MACs and model size. This favorable performance–complexity trade-off highlights the practicality of hybrid buffering, especially in resource-constrained scenarios.

IV. CONCLUSION

This paper systematically evaluates explicit, implicit, and hybrid temporal buffering strategies across four inter-frame coding frameworks under unified settings. Results show hybrid buffering consistently outperforms explicit buffering with minimal added complexity. Advanced frameworks like conditional residual coding and masked conditional residual coding benefit most from hybrid buffering through effective use of explicit and implicit information. These findings offer clear insights into how buffering strategies affect coding performance and complexity under unified settings, supporting informed design choices for future learned video codecs, especially in memory-constrained scenarios. Expanding the analysis by adopting more advanced compression architectures in the unified codec to examine different coding frameworks and buffering strategies under similar model complexity is among our future work.

REFERENCES

[1] J. Li, B. Li, and Y. Lu, "Neural video compression with diverse contexts," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, Canada, June 18-22, 2023*, 2023.

[2] ——, "Neural video compression with feature modulation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 17-21, 2024*, 2024.

[3] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "Dvc: An end-to-end deep video compression framework," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 006–11 015.

[4] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, "Scale-space flow for end-to-end optimized video compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8503–8512.

[5] F. Brand, J. Seiler, and A. Kaup, "On benefits and challenges of conditional interframe video coding in light of information theory," in *2022 Picture Coding Symposium (PCS)*. IEEE, 2022, pp. 289–293.

[6] ——, "Conditional residual coding: A remedy for bottleneck problems in conditional inter frame coding," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, 2024.

[7] Y.-H. Chen, H.-S. Xie, C.-W. Chen, Z.-L. Gao, M. Benjak, W.-H. Peng, and J. Ostermann, "Maskcrt: Masked conditional residual transformer for learned video compression," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[8] Y.-H. Chen, K.-W. Ho, M. Benjak, J. Ostermann, and W.-H. Peng, "On the rate-distortion-complexity trade-offs of neural video coding," in *2024 IEEE 26th International Workshop on Multimedia Signal Processing*. IEEE, 2024.

[9] ——, "Conditional residual coding with explicit-implicit temporal buffering for learned video compression," in *2025 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2025.

[10] Y.-H. Chen, Y.-C. Yao, K.-W. Ho, C.-H. Wu, H.-T. Phung, M. Benjak, J. Ostermann, and W.-H. Peng, "Hytip: Hybrid temporal information propagation for masked conditional residual video coding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2025.

[11] J. Li, B. Li, and Y. Lu, "Deep contextual video compression," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[12] Y.-H. Ho, C.-P. Chang, P.-Y. Chen, A. Gnutti, and W.-H. Peng, "Canf-vc: Conditional augmented normalizing flows for video compression," *European Conference on Computer Vision*, 2022.

[13] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[14] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[15] A. Mercat, M. Viitanen, and J. Vanne, "Uvg dataset: 50/120fps 4k sequences for video codec analysis and development," in *Proceedings of the 11th ACM Multimedia Systems Conference*, 2020, pp. 297–302.

[16] F. Bossen *et al.*, "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, no. 7, 2013.

[17] D. Flynn *et al.*, "Common test conditions and software reference configurations for hevc range extensions," *JCVTC-n1006*, 2013.

[18] H. Wang, W. Gan, S. Hu, J. Y. Lin, L. Jin, L. Song, P. Wang, I. Katsavounidis, A. Aaron, and C.-C. J. Kuo, "Mcl-jcv: a jnd-based h. 264/avc video quality assessment dataset," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 1509–1513.

[19] F. Developers, "Ffmpeg," https://www.ffmpeg.org/, 2016, accessed: 2022-05-18.

[20] "Hstp-vid-wpom - working practices using objective metrics for evaluation of video coding efficiency experiments." iTU-T Technical Paper, 2020.

[21] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[22] "Ecm," https://vcgit.hhi.fraunhofer.de/ecm/ECM.

[23] V. V. Menon, C. Feldmann, H. Amirpour, M. Ghanbari, and C. Timmerer, "Vca: video complexity analyzer," in *Proceedings of the 13th ACM multimedia systems conference*, 2022.