

Module-wise Quantization for Transformer-based Neural Video Coding

Matheus Carvalho, Ricardo Queiroz, Ruhan Conceição, Marcelo Porto, Luciano Agostini, and Wen-Hsiao Peng

Abstract—Transformer-based neural video codecs achieve strong rate-distortion performance, but their high complexity and reliance on floating-point arithmetic hinder practical deployment. In this work, we propose a module-wise post-training quantization (PTQ) pipeline for the MaskCRT codec, where each module is wrapped to interface with a quantization framework, enabling input tracing, calibration, and export of reusable encodings without retraining or modifying the codec core. Experimental results show that moderate precisions (12-16 bits) preserve near-anchor performance, whereas 8-bit activations cause severe degradation and unstable rate-distortion behavior. Single-module analysis identifies *Conditional Motion* and *Residual* as the most precision-sensitive components. Guided by this analysis, an RD-BOP-guided allocation policy greedily assigns bit-widths to maximize rate-distortion recovery per unit of BOP cost. At the same BOP budget as uniform 12-bit integer precision $W_{12}A_{12}$, the proposed policy reduces aggregate BD-rate degradation from +3.59% to +2.20% over HEVC-B and UVG. To the best of our knowledge, this is the first systematic PTQ study for a Transformer-based neural video codec.

Keywords—Neural video compression, post-training quantization, mixed-precision, transformers

I. INTRODUCTION

Neural video coding (NVC) has demonstrated competitive rate-distortion performance compared to traditional standards [1], [2], [3], [4], [5], [6]. Early NVC frameworks largely followed the residual paradigm adopted by conventional codecs, encoding the difference between the target frame x_t and a temporal prediction x_c (the *residual*). As an evolution of residual coding (RC), subsequent works explored conditional coding (CC), in which the latent representation is modeled conditionally on the predictor. In general, CC can be more compression-efficient than RC because conditioning may reduce uncertainty in the representation. However, CC may suffer from an *information bottleneck*, motivating the conditional residual coding framework (CRC), which combines both strategies [7].

Motivated by this limitation, masked conditional residual coding (MCR) was introduced to combine the strengths of CC and CRC by learning a pixel-wise soft mask that adaptively balances the two coding modes. A representative work in this family is the *Masked Conditional Residual Transformer for Learned Video Compression* (MaskCRT) [5], which employs

a transformer-based conditional autoencoder and can surpass the state-of-the-art traditional standard, Versatile Video Coding (VVC) [8].

Despite these advances, MaskCRT still faces major barriers to practical deployment that are common to NVC methods, particularly high computational complexity and a strong reliance on floating-point arithmetic. Complexity in NVC systems is often quantified in terms of multiply-accumulate operations (MACs), reflecting the compute required by analysis/synthesis transforms, motion estimation, and entropy models. Moreover, operating predominantly in 32-bit floating point can increase energy consumption in system-on-chip designs and may introduce cross-platform round-off mismatches [9], [10].

In this context, neural network quantization is a promising approach to mitigate these limitations by enabling integer-arithmetic inference with reduced precision, which can improve cross-platform consistency and lower hardware cost. The present study evaluates the rate-distortion impact of quantization through simulation; hardware deployment is not addressed. Building on recent efforts toward practical learned video compression, this work investigates Post-Training Quantization (PTQ) of MaskCRT by (i) evaluating mixed-precision weight and activation settings, (ii) analyzing per-module sensitivity to quantization, and (iii) proposing an RD-BOP-guided module-wise allocation policy that assigns per-module bit-widths to minimize rate-distortion degradation under a fixed compute budget.

II. BACKGROUND

A. Related Works

Quantization has become a common strategy in learned data compression to reduce computational cost and model size, improve cross-platform determinism, and maintain rate-distortion performance close to floating point. Accordingly, a large body of work has investigated quantization for learned image compression [9], [11], [12], [13], [14], [15], [16]. In contrast, quantization for NVC remains relatively underexplored, with only a few works explicitly targeting practical deployment constraints. Conceição *et al.* [10] applied 8-bit post-training quantization to the hyperprior decoding path of the Scale-Space Flow (SSF) codec to improve cross-platform compatibility. Jia *et al.* [17] presented a real-time NVC design that supports fixed 16-bit integer inference. MobileNVC [18] introduced an 8-bit integer codec design for real-time operation on edge devices. Notably, most prior quantization studies focus on image compression, and the few works on video do

Matheus Carvalho and Ricardo L. de Queiroz are with the Department of Electrical Engineering, Universidade de Brasília (UnB). Emails: carvalho-matheus.mc@aluno.unb.br and queiroz@ieee.org. Ruhan Conceição, Marcelo Porto, and Luciano Agostini are with the Universidade Federal de Pelotas (UFPEL). Emails: radconceicao@inf.ufpel.edu.br, porto@inf.ufpel.edu.br, and agostini@inf.ufpel.edu.br. Wen-Hsiao Peng is with National Yang Ming Chiao Tung University (NYCU). Email: wpeng@cs.nctu.edu.tw.

not specifically address transformer-based NVC architectures, which is the focus of this work.

Mixed-precision quantization has been widely investigated as a means of improving the efficiency–accuracy trade-off of neural networks. Rakka et al. [19] provide a broad survey of mixed-precision neural networks, highlighting the difficulty of selecting suitable bit-widths across models, datasets, and deployment constraints. Van Baelen and Karsmakers [20] proposed Constraint Guided Model Quantization (CGMQ), a QAT-based approach that jointly learns bit-widths under an explicit BOP constraint, without manual hyperparameter tuning. Sensitivity-driven PTQ has also been explored at the parameter level: Zheng et al. [21] use Hessian-based sensitivity to guide the ordering of parameter quantization and accelerate OBQ-style methods. In contrast, our work targets a different setting: post-training quantization of a Transformer-based neural video codec, where bit-widths are assigned at the codec-module level using empirical rate-distortion sensitivity, measured by BD-rate, under a fixed normalized BOP budget.

B. MaskCRT

MaskCRT is an NVC that incorporates Swin-Transformer blocks within both the conditional motion codec and the conditional inter-frame codec. Following the MCR framework, given the current frame x_t and previously decoded reference frames, the codec estimates and encodes optical flow in the conditional motion codec to generate a temporal predictor x_c . To adaptively combine CC and CRC, MaskCRT learns a pixel-wise soft mask m and forms the masked input shown in

$$x_t - m \odot x_c = (1 - m) \odot x_t + m \odot (x_t - x_c), \quad (1)$$

where $\odot$ denotes element-wise multiplication. When $m = 0$, the formulation reduces to conditional coding, whereas when $m = 1$ it reduces to conditional residual coding, enabling a spatially adaptive trade-off between the two regimes.

The codec comprises eight modules: *MENet* (motion estimation) estimates optical flow between the reference and current frames; *MWNet* (motion extrapolation) extrapolates motion from previously decoded flows; *Motion* (motion coder) compresses the estimated motion representation; *CondMotion* (conditional motion coder) codes motion conditioned on a temporal prior; *MCNet* (motion compensation) refines the motion-compensated prediction to form the temporal predictor x_c ; *MaskGenerator* (soft mask generation) generates the soft mask m that blends CC and CRC; *Residual* (conditional inter-frame coder) codes the masked signal $x_t - m \odot x_c$; and *Intra Frame Coder* (I-Frame coder) handles I-frames when no temporal predictor is available. Both *CondMotion* and *Residual* employ Swin-Transformer-based architectures.

C. Neural Network Quantization and AIMET

Neural network quantization reduces numerical precision by mapping floating-point computations to lower bit-width fixed-point (integer) representations [22]. In practice, quantization methods are commonly categorized into quantization-aware training (QAT) and PTQ. QAT simulates quantization effects during training or fine-tuning to recover accuracy, but it is

more costly due to the need for retraining and often requires careful hyperparameter tuning. PTQ, in contrast, applies quantization after training and is therefore attractive when retraining is impractical or prohibitively expensive, as is often the case for NVC.

PTQ is applied to a pretrained model and typically relies on a calibration dataset to estimate activation statistics. A standard PTQ pipeline consists of selecting target bit-widths for weights and activations, inserting quantizers into a model simulation, and performing calibration to determine the quantization parameters. The calibration output is commonly stored as *encodings*, which contain the quantization parameters, such as scales and zero-points (for asymmetric quantization), used to map floating-point values to integer values.

The AI Model Efficiency Toolkit (AIMET) [23] is an open-source toolkit developed by Qualcomm for model optimization. It supports PTQ in *PyTorch* and provides a quantization simulation workflow to assess accuracy impact prior to deployment. Specifically, AIMET inserts quantizers into the model, computes encodings from calibration data, and exports quantization parameters that can be reused during evaluation and deployment.

III. METHODOLOGY

This section details the three pillars of the proposed approach: the experimental setup (Section III-A), the module-wise PTQ pipeline (Section III-B), and the RD-BOP-guided mixed-precision allocation (Section III-C).

A. Experimental Setup

We implement PTQ using AIMET with symmetric quantization for weights and asymmetric quantization for activations, following common practice in the literature. For activation calibration, we use the Vimeo90k dataset [24]. Specifically, we sample 1,000 sequences, each containing 7 frames at a resolution of 448×256 , and run MaskCRT inference to trace per-module inputs. For the residual (inter-frame) module, we adopt a larger calibration set of 2,000 clips due to the increased sensitivity of this module and the numerical instability observed. Both sizes were selected empirically and were not optimized through a dedicated ablation.

To evaluate the compression efficiency effects on MaskCRT coding performance, we report the aggregate results on the UVG [25] and HEVC Common Test Conditions Class B (HEVC-B) datasets [26]. Test sequences are converted from YUV420 to RGB444 using ITU-R BT.709 coefficients. We use an intra period of 32 frames and encode the first 96 frames of each sequence. Compression efficiency performance is evaluated using Bjøntegaard Delta Rate (BD-rate) computed from peak signal-to-noise ratio (PSNR), with rate measured in bits per pixel (bpp) [27]. BD-rate compares the average bitrate between two RD curves at equal distortion. Negative values indicate bitrate savings relative to the anchor, while positive values indicate a bitrate increase. The anchor in the experiments is the full-precision MaskCRT model in 32-bit floating point (FP32). For this calculation, we used the four

operating points (Q2-Q5) available in the original MaskCRT implementation.

For quantized inference, we consider weight and activation bit-widths of 8, 12, and 16 bits and evaluate both uniform and mixed-precision configurations, denoted as $W_n A_m$ (weights at n bits and activations at m bits). These settings allow the wrapper-based PTQ study to reflect deployment-relevant trade-offs across uniform and mixed precisions.

All quantized results are obtained using AIMET quantization simulation, in which quantization is emulated in floating point through the insertion of quantizers. Integer-only hardware inference is not evaluated; latency, energy consumption, and cross-platform determinism are therefore outside the scope of these experiments.

B. Module-wise PTQ Pipeline

Fig. 1 illustrates the proposed module-wise PTQ pipeline, which consists of two independent phases: calibration (left) and inference (right).

Calibration phase: For each of the eight MaskCRT modules M , a lightweight wrapper exposes a forward signature compatible with AIMET’s *QuantizationSimModel* and performs any required pre/post-processing so the module can be calibrated in isolation. During a standard MaskCRT forward pass, the wrapper traces and stores module inputs, producing a reusable per-module calibration set without modifying the codec core. These stored inputs are replayed under *QuantizationSimModel* to collect activation statistics and derive scales and zero-points (quantization parameters), which are exported and stored for reuse. The loop is repeated for each module M at its assigned bit-width $W_n A_m$.

Inference phase: Each module M is replaced by its quantized counterpart, the previously exported encodings are loaded, and the full codec is evaluated on FullHD video sequences to produce the rate-distortion results reported in Section IV. Because calibration inputs are cached at module boundaries, the target bit-width can be changed without re-running the full codec on the calibration set, substantially reducing exploration cost across precision settings.

C. RD-BOP-guided Module-wise Allocation

Let M denote the set of codec modules presented in Table I and $T = \{(8, 12), (12, 12), (12, 16), (16, 16)\}$ the discrete set of precision tiers $\tau = (W, A)$, where an $A \geq 12$ floor is enforced because $A = 8$ configurations caused unstable RD behavior in the single-module study.

For each module $m \in M$ and precision tier $\tau \in T$, we define two quantities, as follows. Firstly, the *sensitivity score*

$$S(m, \tau), \quad (2)$$

is the BD-rate penalty (in %) observed when only module m is quantized to τ , with all others at FP32. The allocation reported in Section IV uses sensitivity estimates derived from the symmetric configurations of Table I ($W_8 A_8$, $W_{12} A_{12}$, $W_{16} A_{16}$), with the asymmetric tiers (8,12) and (12,16) estimated by linear interpolation in the product $W \cdot A$; this keeps the single-module ablation cost independent of $|T|$. A

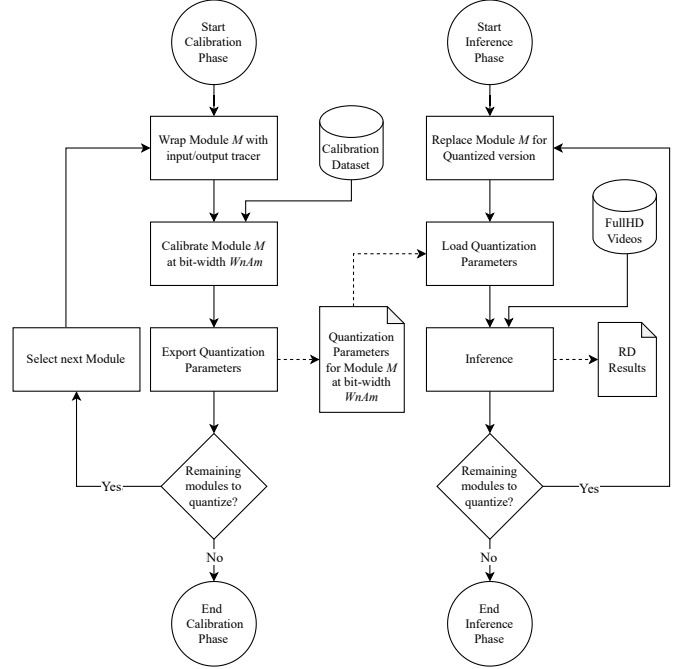


Fig. 1. Module-wise PTQ pipeline: per-module calibration (left) and quantized inference (right), iterated independently for each module M in MaskCRT at assigned bit-width $W_n A_m$.

higher score indicates greater sensitivity to quantization at that tier. The *per-module compute cost* in bit-operations (BOP) is

$$C(m, \tau) = W \cdot A \cdot \kappa_m, \quad (3)$$

where κ_m is the complexity of module m in kMAC/pixel, obtained from its multiply–accumulate count normalized by the number of input pixels and reported in Table I (κ column). Because C scales with κ_m , upgrading a large module costs proportionally more BOPs than the same tier upgrade on a small one.

The goal is to find an assignment $\pi: M \rightarrow T$ that maps each module to a precision tier so as to minimize the total RD degradation under a fixed BOP budget B , set equal to the total BOP of uniform $W_{12} A_{12}$. Formally, the desired allocation is

$$\begin{aligned} \pi^* = \arg \min_{\pi: M \rightarrow T} \sum_{m \in M} S(m, \pi(m)) \\ \text{s.t.} \quad \sum_{m \in M} C(m, \pi(m)) \leq B. \end{aligned} \quad (4)$$

We approximate this combinatorial problem with a greedy heuristic: all modules start at the lowest tier (8,12), and at each step the upgrade $(m, \pi(m) \rightarrow \tau_{\text{next}})$ that yields the highest sensitivity reduction per unit BOP spent,

$$\frac{S(m, \pi(m)) - S(m, \tau_{\text{next}})}{C(m, \tau_{\text{next}}) - C(m, \pi(m))}, \quad (5)$$

is applied, until no further upgrade fits within B . This marginal efficiency criterion naturally favors sensitive modules that are also computationally cheap to protect. The heuristic uses single-module sensitivities as independent marginal estimates and therefore neglects cross-module interaction effects; the final selected allocation is nevertheless evaluated end-to-end.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

We first analyze per-module sensitivity, then compare the proposed RD-BOP-guided allocation against uniform precision baselines.

A. Single-Module Quantization Analysis

Table I reports the single-module sensitivity study: only one module is quantized at a time while all others remain in FP32. Under the aggressive W_8A_8 setting, *Residual* dominates with a +33.86% BD-rate increase, confirming it is the most sensitive component. *MCNet* and *CondMotion* also show notable penalties (+7.99% and +8.32%, respectively). Among these, *CondMotion* and *Residual* employ Swin-Transformer-based architectures, whose attention mechanisms are known to be sensitive to fixed-point rounding, while *MCNet* operates directly on the motion-compensated prediction and is tightly coupled to the residual signal.

TABLE I

PER-MODULE COMPLEXITY κ (KMAC/PIXEL) AND SINGLE-MODULE BD-RATE ($\downarrow$ %) AVERAGED OVER UVG AND HEVC-B. FOR THE BD-RATE COLUMNS, ONLY THE INDICATED MODULE IS QUANTIZED; ALL OTHERS REMAIN IN FP32.

Module	κ	W_8A_8	W_8A_{12}	$W_{12}A_{12}$	$W_{12}A_{16}$	$W_{16}A_{16}$
Residual	534.4	+33.86	+2.38	+0.62	+0.25	+0.22
CondMotion	116.8	+8.32	+1.80	+2.90	+1.19	+1.21
MCNet	236.5	+7.99	+3.70	+0.07	+0.03	0.00
I-Frame	549.1	+5.36	+0.41	+0.05	+0.01	0.00
MaskGenerator	56.2	+2.81	+0.05	+0.01	+0.01	0.00
MWNet	38.9	+0.36	+0.11	+0.02	0.00	+0.01
Motion	69.0	+0.34	+0.14	+0.05	+0.10	+0.03
MENet	319.6	+0.19	+0.21	+0.01	+0.14	+0.15

Under $W_{12}A_{12}$, most modules recover to negligible RD loss ($\leq +0.07\%$), but *CondMotion* remains uniquely problematic at +2.90%, and *Residual* retains mild sensitivity (+0.62%). This asymmetry is the key observation that motivates the mixed-precision policy of Section III-C: *Residual* is catastrophic at W_8A_8 but well-behaved at $W_{12}A_{12}$, while *CondMotion* is the only module that continues to hurt the codec at moderate precision. Moving to $W_{16}A_{16}$, all modules except *CondMotion* (+1.21%) converge to near-zero BD-rate penalty, confirming that 16-bit precision is effectively transparent for most components.

B. Mixed-Precision Policy Results

Table II reports representative uniform baselines and the proposed allocation, while Fig. 2 shows the full set of evaluated RD curves. Uniform W_8A_8 provides the largest BOP reduction (Norm. BOP = 0.06) but is not a practical operating point, degrading aggregate BD-rate by +86.22% over the HEVC-B and UVG average. Increasing precision to $W_{12}A_{12}$ substantially recovers performance (+3.59%), and $W_{16}A_{16}$ approaches the anchor (+0.95%) at nearly twice the BOP cost.

The RD-BOP-guided allocation converts the single-module RD sensitivity analysis into a concrete mixed-precision decision under a fixed compute budget, following Eqs. (4)-(5).

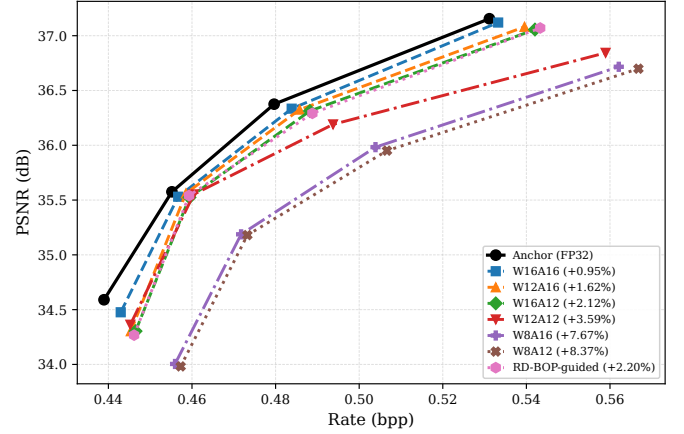


Fig. 2. Aggregate rate-distortion curve (PSNR vs. bpp) averaged over HEVC-B and UVG for the FP32 anchor, uniform PTQ, and the RD-BOP-guided mixed-precision allocation. Values in parentheses denote BD-rate relative to the anchor.

TABLE II

BD-RATE (%) VS. NORMALIZED BOP FOR UNIFORM PRECISION AND THE RD-BOP-GUIDED MIXED-PRECISION ALLOCATION AVERAGED OVER HEVC-B AND UVG (ANCHOR: FP32 MASKCMT).

Config	BD-rate (%)	Norm. BOP
W_8A_8	+86.22	0.06
$W_{12}A_{12}$	+3.59	0.14
$W_{16}A_{16}$	+0.95	0.25
RD-BOP-guided	+2.20	0.14

At the same normalized BOP as uniform $W_{12}A_{12}$ (0.14), it improves aggregate BD-rate from +3.59% to +2.20%, a gain of 1.39 percentage points. The RD curves in Fig. 2 confirm a consistent aggregate improvement over $W_{12}A_{12}$, with the policy yielding lower BD-rate across the evaluated rate range. This result validates that the sensitivity study is not merely diagnostic: directing precision to *CondMotion*, the module with the highest marginal efficiency ratio under the estimates used by the policy, yields consistently better rate-distortion performance under the same compute budget.

The resulting allocation assigns $W_{16}A_{16}$ to *CondMotion*, $W_{12}A_{16}$ to *MWNet*, W_8A_{12} to *MENet*, and $W_{12}A_{12}$ to the remaining modules (*Residual*, *MCNet*, *MaskGenerator*, *Motion*, and *I-Frame* coder). The key decision is the interplay between *CondMotion* and *MENet*: although *Residual* has the highest raw sensitivity at W_8A_8 (+33.86%), it recovers well at $W_{12}A_{12}$ (+0.62%), so upgrading it further offers little gain. *CondMotion*, by contrast, remains the most degraded module under $W_{12}A_{12}$ (+2.90%) and has a relatively small kMAC footprint ($\kappa = 116.8$, Table I), making its upgrade to $W_{16}A_{16}$ cost-efficient. The budget for this upgrade is partly funded by *MENet*, whose low sensitivity (+0.19% at W_8A_8) and large footprint ($\kappa = 319.6$) make it an ideal candidate to remain at the minimum W_8A_{12} tier. This trade-off follows the interpolated sensitivity estimates used by the policy (Section III-C): spend precision where sensitivity is high and the BOP cost is low.

The complete measurements in Table I, including the asym-

metric tiers, were obtained after the allocation had been fixed, and show that $S(m, \tau)$ is not monotone in $W \cdot A$. For *CondMotion*, the measured penalty at W_8A_{12} (+1.80%) is lower than at $W_{12}A_{12}$ (+2.90%), and $W_{12}A_{16}$ (+1.19%) is essentially tied with $W_{16}A_{16}$ (+1.21%) at lower cost; the interpolated estimates therefore overestimate its sensitivity at the lower tiers, and re-running the allocation on the fully measured grid would assign $W_{12}A_{16}$ to this module. The reported allocation is nevertheless validated end-to-end and remains the best evaluated configuration at Norm. BOP = 0.14; searching directly over non-monotone sensitivities is left for future work.

The 1.39 percentage-point gain is modest in absolute terms, but it is obtained without retraining and under the same normalized BOP budget. The module-wise pipeline itself applies to other modular neural video codecs, although sensitivity scores and the resulting allocation must be re-estimated for each architecture.

V. CONCLUSIONS

This paper presented a module-wise PTQ pipeline for MaskCRT, a Transformer-based NVC, enabling per-module calibration and mixed-precision evaluation without retraining or modifying the codec core. Single-module experiments (Table I) revealed a strong sensitivity hierarchy. *Residual* is catastrophic at W_8A_8 but recovers at $W_{12}A_{12}$, while *CondMotion* remains the most degraded module at moderate precision, making it the primary target for precision protection.

Building on this analysis, the proposed RD-BOP-guided mixed-precision allocation (Eqs. 4-5) greedily selects bit-width upgrades according to the highest marginal rate-distortion recovery per unit BOP cost. At the same normalized BOP as uniform $W_{12}A_{12}$, the policy improves aggregate BD-rate from +3.59% to +2.20% over HEVC-B and UVG (Fig. 2, Table II), demonstrating that module-level sensitivity can be converted into a practical mixed-precision allocation under the same normalized BOP budget. Future work includes extending the module-wise sensitivity analysis to other Transformer-based neural video codecs, combining PTQ with structured pruning under a joint BOP budget, and applying quantization-aware training selectively to the modules identified as most precision-critical by the RD-BOP analysis.

REFERENCES

- [1] J. Li, B. Li, and Y. Lu, "Deep Contextual Video Compression," *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.
- [2] X. Sheng, J. Li, B. Li, L. Li, D. Liu, and Y. Lu, "Temporal context mining for learned video compression," *IEEE Trans. Multimedia*, vol. 25, pp. 7311–7322, 2022.
- [3] J. Li, B. Li, and Y. Lu, "Hybrid Spatial-Temporal Entropy Modelling for Neural Video Compression," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022.
- [4] —, "Neural Video Compression with Diverse Contexts," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Vancouver, Canada, June 2023.
- [5] 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," 2024. [Online]. Available: <https://arxiv.org/abs/2312.15829>
- [6] 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)*, 2025.
- [7] 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," 2024. [Online]. Available: <https://arxiv.org/abs/2410.03898>
- [8] ITU-T and I. J. T. C. S. 29, "Versatile Video Coding (VVC)," International Telecommunication Union (ITU), Tech. Rep. H.266, 2020. [Online]. Available: <https://www.itu.int/rec/T-REC-H.266>
- [9] J. Ballé, N. Johnston, and D. Minnen, "Integer Networks for Data Compression with Latent-Variable Models," in *Proc. Int. Conf. Learn. Representations (ICLR)*, New Orleans, LA, USA, 5 2019.
- [10] R. Conceição, M. Porto, W. H. Peng, and L. Agostini, "Cross-Platform Neural Video Coding: A Case Study," in *Proceedings - IEEE International Symposium on Circuits and Systems*. Institute of Electrical and Electronics Engineers Inc., 2025.
- [11] H. Sun, L. Yu, and J. Katto, "End-to-End Learned Image Compression with Quantized Weights and Activations," 2021.
- [12] —, "Q-lic: Quantizing learned image compression with channel splitting," 2022. [Online]. Available: <https://arxiv.org/abs/2205.14510>
- [13] X. Hang, C. Jia, S. Ma, and W. Gao, "A near lossless learned image coding network quantization approach for cross-platform inference," in *2023 IEEE International Conference on Visual Communications and Image Processing (VCIP)*. IEEE, 2023, pp. 1–5.
- [14] E. Koyuncu, T. Solovyev, E. Alshina, and A. Kaup, "Device interoperability for learned image compression with weights and activations quantization," in *2022 Picture Coding Symposium (PCS)*. IEEE, 2022, pp. 151–155.
- [15] W. Hong, T. Chen, M. Lu, S. Pu, and Z. Ma, "Efficient neural image decoding via fixed-point inference," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 9, pp. 3618–3630, 2021.
- [16] J. Shi, M. Lu, and Z. Ma, "Rate-distortion optimized post-training quantization for learned image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 34, no. 5, pp. 3082–3094, 2024. [Online]. Available: <https://doi.org/10.1109/TCSVT.2023.3323015>
- [17] Z. Jia, B. Li, J. Li, W. Xie, L. Qi, H. Li, and Y. Lu, "Towards Practical Real-Time Neural Video Compression," *arXiv preprint arXiv:2502.20762*, 2025. [Online]. Available: <https://github.com/microsoft/DCVC>
- [18] T. van Rozaendaal, T. Singhal, H. Le, G. Sautiere, A. Said, K. Buska, A. Raha, D. Kalatzis, H. Mehta, F. Mayer, L. Zhang, M. Nagel, and A. Wiggers, "Mobilenvc: Real-time 1080p neural video compression on a mobile device," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2024.
- [19] M. Rakka, M. E. Fouda, P. Khargonekar, and F. Kurdahi, "Mixed-precision neural networks: A survey," 2022. [Online]. Available: <https://arxiv.org/abs/2208.06064>
- [20] Q. Van Baelen and P. Karsmakers, "Constraint guided model quantization of neural networks," 2025. [Online]. Available: <https://arxiv.org/abs/2409.20138>
- [21] Z. Zheng, H. Li, Y. Chen, M. Tan, and Q. Du, "Sensitivity-aware post-training quantization for deep neural networks," 2025. [Online]. Available: <https://arxiv.org/abs/2509.05576>
- [22] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," 2021. [Online]. Available: <https://arxiv.org/abs/2106.08295>
- [23] S. Siddegowda, M. Fournarakis, M. Nagel, T. Blankevoort, C. Patel, and A. Khobare, "Neural Network Quantization with AI Model Efficiency Toolkit (AIMET)," 2022. [Online]. Available: <https://arxiv.org/abs/2201.08442>
- [24] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video Enhancement with Task-Oriented Flow," *International Journal of Computer Vision (IJCV)*, vol. 127, pp. 1106–1125, 2019.
- [25] J. Mercat, V. Viitanen, and J. Vanné, "UVG dataset: 50/120fps 4K sequences for video codec analysis and development," in *Proc. ACM Multimedia Syst. Conf. (MMSys)*, Istanbul, Turkey, 6 2020, pp. 297–302.
- [26] F. Bossen and others, "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, no. 7, 2013.
- [27] N. Barman, M. G. Martini, and Y. Reznik, "Bjontegaard Delta (BD): A Tutorial Overview of the Metric, Evolution, Challenges, and Recommendations," 1 2024. [Online]. Available: <http://arxiv.org/abs/2401.04039>