Análise do Impacto da Precisão de Quantização no Decodificador de Hyperprior do SSF

Ruhan Conceição[1,2], Érick Radmann[1], Bruno Zatt[1], Wen-Hsiao Peng[2], Marcelo Porto[1], Luciano Agostini[1]

[1]Universidade Federal de Pelotas (UFPel), Brasil

[2]National Yang Ming Chiao Tung University (NYCU), Taiwan

radconceicao@inf.ufpel.edu.br

Abstract. *This work presents a study on the impact of quantization in the* hyperprior *decoder of the SSF codec, with an exclusive focus on analyzing the precision of weights and activations. We consider only scenarios in which weights and activations share the same bit precision (W4A4, W8A8, W12A12, W16A16). Results on the HEVC-B and UVG datasets demonstrate that: (i) INT4 severely degrades efficiency (BD-rate >200%); (ii) INT8 provides an attractive* trade-off*, with an average penalty of ≤7% in terms of BD-rate; (iii) INT12 achieves near parity with FP32 (loss ≈0.2%); and (iv) INT16 does not yield significant additional gains. These findings indicate that quantizing the hyperprior decoder not only ensures cross-platform consistency but also enables efficient hardware deployment.*

Resumo. *Este trabalho apresenta um estudo sobre o impacto da quantização no decodificador de* hyperprior *do codec SSF, com foco exclusivo na análise da precisão dos pesos e ativações. Consideramos apenas cenários em que pesos e ativações possuem a mesma precisão de bits (W4A4, W8A8, W12A12, W16A16. Resultados nos datasets HEVC-B e UVG demonstram que: (i) INT4 degrada severamente a eficiência (BD-rate >200%); (ii) INT8 oferece* trade-off *atrativo, com penalidade média ≤7% em termos de BD-rate; (iii) INT12 atinge praticamente a paridade com FP32 (perda ≈0.2%); e (iv) INT16 não agrega ganhos relevantes. Esses achados indicam que a quantização do decodificador de hyperprior não apenas garante consistência entre plataformas, mas também viabiliza implantação eficiente em hardware.*

1. Introdução

Nos últimos anos, codecs neurais surgiram como uma alternativa promissora aos métodos tradicionais de codificação de vídeo, explorando arquiteturas baseadas em autoencoders e modelos probabilísticos aprendidos. Em particular, a introdução do *hyperprior* [Ballé et al. 2018] representou um avanço significativo, pois permite modelar de forma mais precisa as distribuições do código latente e, consequentemente, reduzir a entropia da informação transmitida. Esse recurso foi incorporado em arquiteturas modernas como o SSF [Agustsson et al. 2020], que atingem desempenho comparável aos codecs convencionais em diversas condições. No entanto, apesar desse progresso em termos de eficiência de compressão, a implementação prática desses sistemas ainda enfrenta desafios relacionados a custo computacional, consumo de energia e inconsistências entre

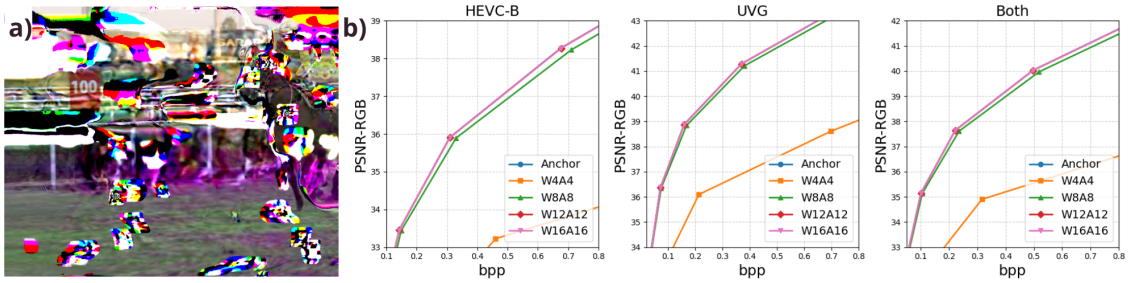

Figura 1. a) Exemplo de resíduo gerado na reconstrução da imagem devido às inconsistências entre plataformas. b) Gráficos de taxa de bits vs. qualidade para o trabalho desenvolvido.

plataformas, especialmente no módulo de *hyperprior*, que é altamente sensível a erros de quantização.

No entanto, a implantação prática desses modelos enfrenta desafios importantes. Em primeiro lugar, a maior parte dos frameworks de compressão neural é treinada e avaliada em *floating point* (FP32/FP16), o que implica elevado consumo de memória e energia, limitando o uso em dispositivos embarcados. Além disso, diferenças de arredondamento entre implementações de ponto flutuante em CPU, GPU ou DSP podem gerar inconsistências entre plataformas, comprometendo a reprodutibilidade e até a decodificação correta [Ballé et al. 2019, Conceição et al. 2025] (veja a Figura 1(a)).

Uma solução viável para esses desafios é a quantização pós-treinamento (*post-training quantization* – PTQ), que converte pesos e ativações de modelos neurais para representações inteiras de menor precisão [Pyt 2024]. Essa técnica reduz custos de armazenamento e processamento, além de garantir aritmética determinística, eliminando diferenças entre plataformas. Ferramentas como a AIMET [Qualcomm Innovation Center, Inc. 2024] facilitam esse processo, oferecendo estratégias de calibração baseadas em amostras de dados.

Neste resumo expandido, analisamos os impactos da quantização no decodificador de hyperprior do codec SSF, reconhecido como um dos módulos mais sensíveis a erros numéricos. Para isso, consideramos quatro cenários distintos de precisão, nos quais pesos e ativações compartilham o mesmo número de bits: W4A4, W8A8, W12A12 e W16A16, onde N em WNAN indica o número de bits inteiros utilizados para representar os pesos (*weights*) e ativações (*activations*).

2. Metodologia

Neste trabalho, adota-se o codec SSF, cujo modelo treinado é disponbilizado na biblioteca CompressAI [Begaint et al. 2020], e quantizado empregando a biblioteca AIMET. A calibração dos parâmetros de quantização neste trabalho empregou 1000 sequências do Vimeo90k, recortadas para a dimensão de 256×256, utilizando um modo de definição dos parâmetros de quantização orientado pela métrica MSE. Neste trabalho, foram avaliados quatro cenários: W4A4, W8A8, W12A12 e W16A16.

Os testes foram realizados com os *datasets* de vídeo HEVC-B [Bossen et al. 2013] e UVG [Mercat et al. 2020], ajustados para 1920×1024, sendo codificados em sequências de 96 quadros. A codificação utilizou um quadro intra inicial seguido de P-frames.

O desempenho em eficiência de compressão foi avaliado através da métrica BD-rate [Bjøntegaard 2001], que estima a variação percentual média na taxa de bits necessária para alcançar a mesma qualidade em curvas RD, permitindo assim comparar diferentes configurações de forma objetiva. Para estabilidade, apenas os níveis de qualidade ímpares (1, 3, 5, 7, 9) foram considerados, utilizando FP32 como âncora.

3. Resultados

A Tabela 1 apresenta os valores médios de BD-rate para os cenários avaliados, obtidos a partir das curvas de taxa-distorção mostradas na Figura 1(b). Nota-se que a perda de eficiência cresce à medida que a precisão é reduzida, embora de forma não linear: enquanto a configuração W8A8 mantém desempenho competitivo, o caso W4A4 mostra-se inviável.

Tabela 1. BD-rate (%) utilizando diferentes precisões. FP32 como âncora.

W4A4	W8A8	W12A12	W16A16
254,0	5,7	0,2	0,1

Os resultados evidenciam que a quantização extrema em W4A4 degrada severamente a eficiência de compressão, resultando em aumento de BD-rate acima de 250%, o que inviabiliza sua adoção. Já a configuração W8A8 mostrou-se um ponto de equilíbrio, com degradação limitada a $\leq$7%, sendo adequada para aplicações em tempo real e sistemas embarcados [Conceição et al. 2025]. A precisão W12A12 apresentou desempenho praticamente indistinguível do FP32, com perdas médias de apenas 0.2%, enquanto W16A16 não trouxe ganhos relevantes adicionais em relação ao W12A12, podendo ser considerado redundante.

4. Discussão

A análise confirma que o decodificador de hyperprior é altamente sensível a quantizações agressivas, com perdas significativas observadas no caso W4A4. Por outro lado, as configurações W8A8 e W12A12 mostraram-se adequadas para manter a eficiência de compressão mesmo em cadeias longas de 96 quadros, indicando que não há acúmulo relevante de erro ao longo da sequência.

Do ponto de vista prático, o uso de W8A8 oferece uma solução eficiente em termos de custo computacional e consumo de energia, aproveitando instruções otimizadas para inteiros em hardware moderno. Já a configuração W12A12 garante fidelidade quase total ao modelo FP32, sendo recomendada para cenários que exigem alta qualidade ou aplicações críticas.

Adicionalmente, a quantização do decodificador de *hyperprior* contribui para eliminar erros decorrentes de inconsistências de arredondamento entre plataformas (*cross-platform round-off errors*) [Ballé et al. 2019, Conceição et al. 2025], garantindo maior robustez na sincronização entre codificador e decodificador em implementações heterogêneas.

5. Conclusões

Este estudo focado na precisão do decodificador de *hyperprior* do SSF mostrou que:

1. W4A4 é inviável (BD-rate $>200\%$).
2. W8A8 representa uma eficiência de compressão aceitável.
3. W12A12 elimina quase totalmente a lacuna frente ao FP32, com perdas $\leq 0.2\%$.
4. W16A16 não traz ganhos práticos adicionais.

Conclui-se que, para implantações reais, recomenda-se W8A8 quando eficiência for prioritária e W12A12 quando fidelidade for imprescindível. A quantização, além de resolver inconsistências entre plataformas, reduz custos de memória e processamento, tornando a adoção de codecs neurais mais viável em dispositivos restritos.

Agradecimentos

Agradecemos às agências FAPERGS, CNPq e CAPES pelo apoio financeiro.

Referências

(2024). Quantization. `https://pytorch.org/docs/stable/quantization.html`. PyTorch Documentation.

Agustsson, E., Minnen, D., Johnston, N., Ballé, J., Hwang, S. J., and Toderici, G. (2020). Scale-space flow for end-to-end optimized video compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA.

Ballé, J., Johnston, N., and Minnen, D. (2019). Integer networks for data compression with latent-variable models. In *Proc. Int. Conf. Learn. Representations (ICLR)*, New Orleans, LA, USA.

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. (2018). Variational image compression with a scale hyperprior. `https://arxiv.org/abs/1802.01436`. arXiv preprint arXiv:1802.01436.

Begaint, F., Mentzer, F., Agustsson, E., and Van Gool, L. (2020). Compressai: A pytorch library and evaluation platform for end-to-end compression research. arXiv preprint arXiv:2011.03029.

Bjøntegaard, G. (2001). Calculation of average psnr differences between rd-curves. ITU-T VCEG-M33.

Bossen, F. et al. (2013). Common test conditions and software reference configurations. *JCTVC-L1100*, 12(7).

Conceição, R., Porto, M., Peng, W.-H., and Agostini, L. (2025). Cross-platform neural video coding: A case study. In *2025 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5.

Mercat, J., Viitanen, V., and Vanne, J. (2020). Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In *Proc. ACM Multimedia Syst. Conf. (MMSys)*, pages 297–302, Istanbul, Turkey.

Qualcomm Innovation Center, Inc. (2024). Ai model efficiency toolkit (aimet). `https://github.com/quic/aimet`. Accessed: 2025-08-12.