Towards Backward-Compatible Continual Learning of Image Compression

Supplementary Material

7. Appendix: Model Architecture Details

In the main paper, Sec. 4.2 provides a high-level overview of the proposed model architecture. This section provides more details about the model architecture, such as the number of channels and stride sizes for each layer.

The detailed model architecture is shown in Fig. 7, where the model components are marked in the same way as in Sec. 4.2. Our model contains four phases, all of which have the same structure, while only different in (1) the number of feature channels, (2) the number of ConvNeXt blocks, and (3) the first phase starts from bias e_0 and r_0 instead of the feature maps from the previous phase.

The spatial dimensions (height and width) in the figure are for an input image with 256×256 pixels. Since the model is fully convolutional, the spatial dimensions of intermediate layer outputs scales accordingly with the input image size. Both initial bias features e_0 and r_0 have a shape of $1 \times 1 \times 128$, and they are repeated spatially to match the spatial dimensions of z_1.

8. Appendix: Training and Fine-tuning Details

Tab. 10 lists the pre-training and fine-tuning hyperparameters used in our experiments. For a fair comparison, we use the same hyperparameters for training all models, including our proposed model and the baseline models (i.e., *MSH-VR* and *GMA-VR*). Note that the fine-tuning dataset varies for different sets of experiments. For data-incremental learning, we use CelebA-HQ [30], and for rate-incremental learning, we use COCO [32], which is the same as the pre-training dataset.

	Pre-training	Fine-tuning
Data augmentation	Crop, h-flip	Crop, h-flip
Input size	256x256	256x256
Optimizer	Adam	Adam
Learning rate	2×10^{-4}	1×10^{-4}
LR schedule	Constant + cosine	Cosine
Weight decay	0.0	0.0
Batch size	32	32
# iterations	500K	100K
# images seen	16M	3.2M
Gradient clip	2.0	2.0
EMA	0.9999	-
GPU	$1 \times$ RTX 3090	$1 \times$ A40
Time	$\approx$ 51 hours	$\approx$ 11 hours

Table 10. Training Hyperparameters. The GPU time is for training our proposed model, and all other hyperparameters are the same for all models.

9. Appendix: Variable-Rate Baseline Models

In the main paper (Sec. 5.2), we mentioned that we construct variable-rate versions of the two baseline models (*i.e.*, *MSH-VR* and *GMA-VR*) in order to use them in the rate-incremental learning experiment. Fig. 11 shows the rate-distortion performance of the variable-rate versions compared to the original ones. As shown in the figure, the variable-rate versions achieve similar performance as the original ones, which validates the our experimental setting.

10. Appendix: Experimental Results

10.1. PSNR-Bpp curves for the main experiments

Due to the space constraint, we show only BD-rate results without PSNR-bpp curves in the main paper. This section provides the PSNR-bpp curves for the main experiments (Sec. 5.3).

Fig. 8 shows the PSNR-bpp curves for data-incremental learning experiments, which includes the backward compatibility experiment (Fig. 8a) and the new-data performance experiment (Fig. 8b). For backward compatibility, it is clear that models with fine-tuned encoder and decoder suffer a significant performance drop on the old bitstreams, while other fine-tuned models obtain comparable performance as the pre-trained models. Among them, our proposed knowledge replay strategy achieves even better performance than using the pre-trained model directly. For new-data performance, our method achieves comparable performance as the models with fine-tuned encoder and decoder (which are not backward compatible), and outperforms the pre-trained models by a clear margin. These observations are consistent with what we have observed in the BD-rate results in the main paper.

We show the PSNR-bpp curves for rate-incremental learning experiments, including the *low-to-high* experiment (Fig. 9) and the *high-to-low* experiment (Fig. 10). The results are consistent with previous observations: (1) Fine-tuning the encoder and decoder does not preserve backward compatibility, while our approach does; and (2) Our approach even outperforms all other methods in terms of new-rate performance.

10.2. Fine-tuning the encoder does not generalize the model to new rates

We mentioned in Sec. 5.3 that fine-tuning the encoder alone cannot effectively extend the rate range of the pre-trained models. We provide an example for showing this in Fig. 12, where we show the rate-incremental learning (low $\rightarrow$ high)

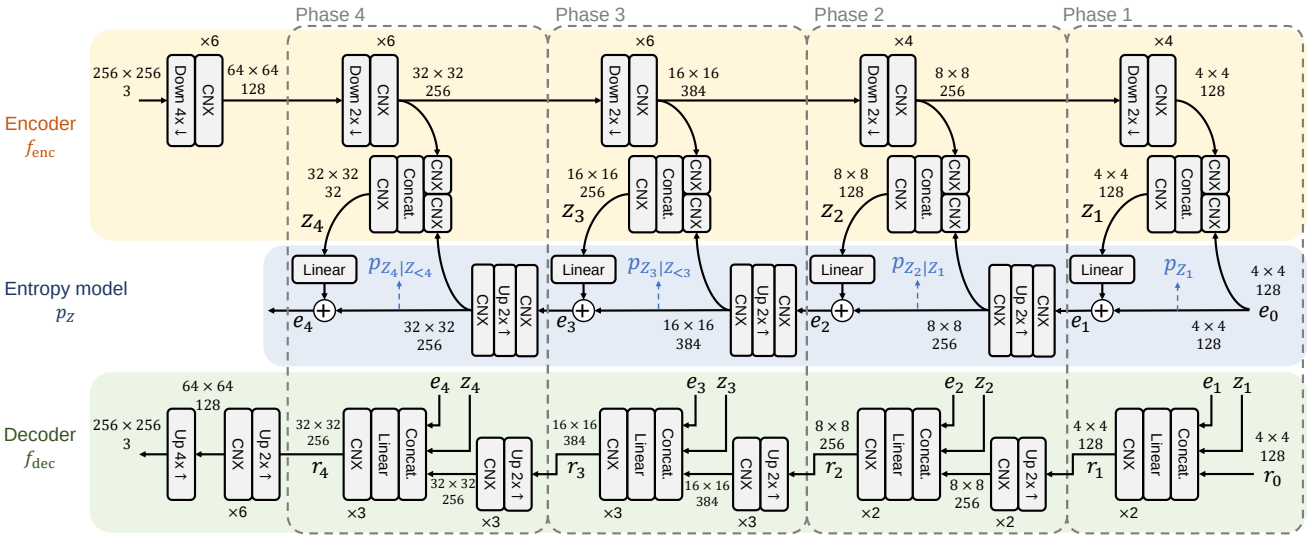
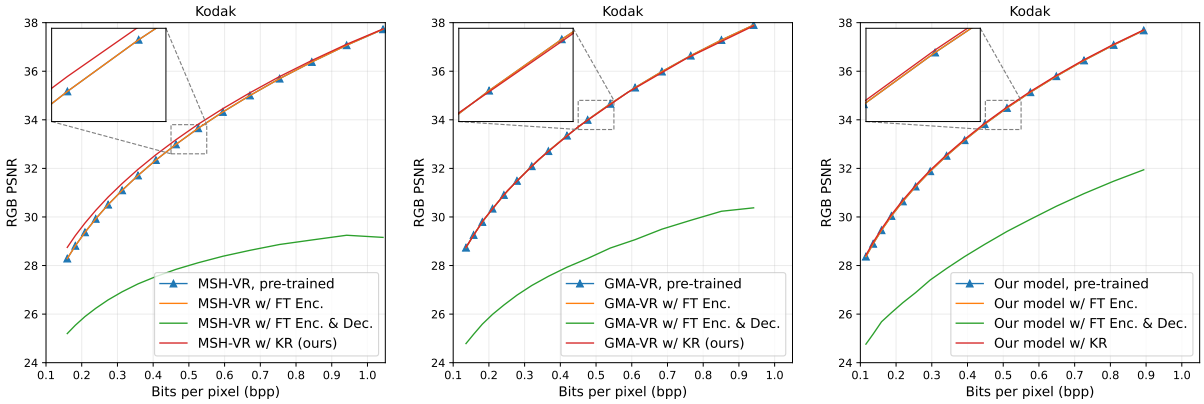
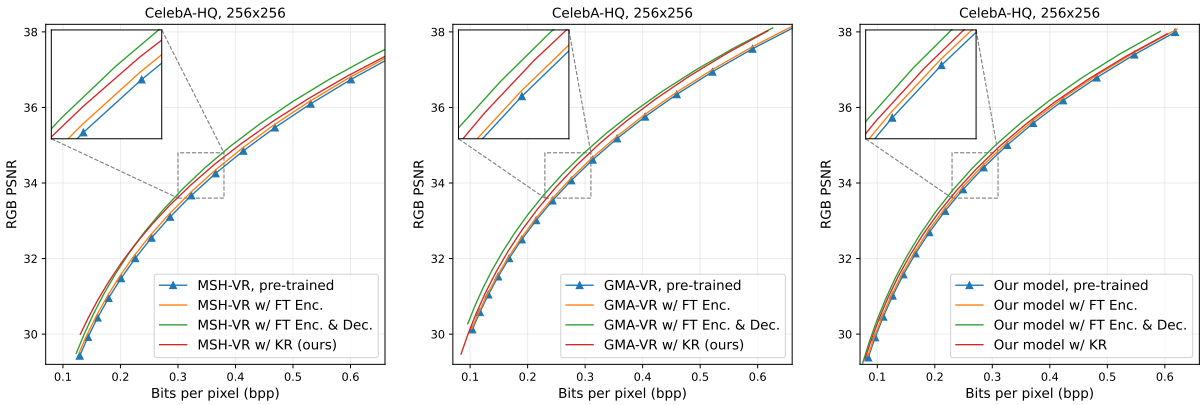


Figure 7. Detailed architecture of the proposed model. In the figure, *CNX* denotes a ConvNeXt block [35] conditioned on lagrange multiplier λ, as described in Fig. 6. Dimensionality of the layer outputs are shown in the format of *height* $\times$ *width* and *channels*, where the spatial dimensions (height and width) are for a 256×256 input image, and they scales linearly with the input image size.

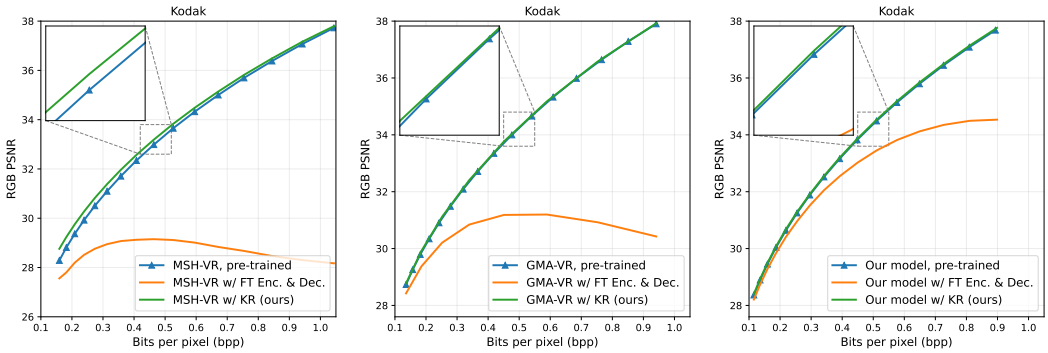


(a) Performance on the old bitsreams of Kodak (backward compatibility). Each subfigure shows the performance of a different model.

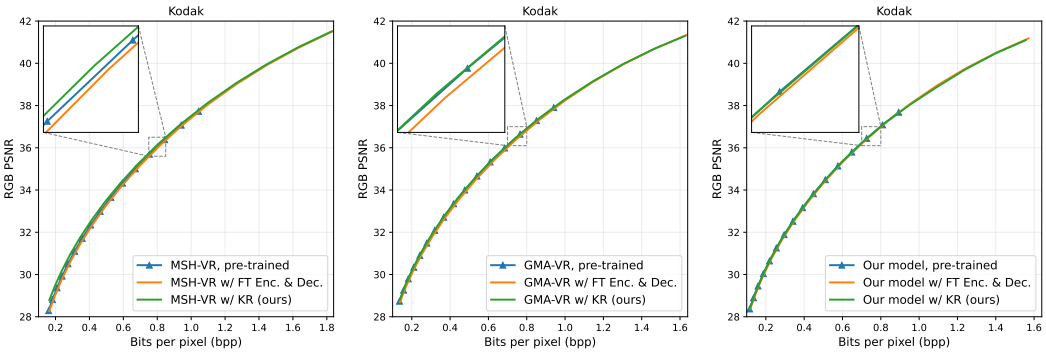


(b) Performance on CelebA-HQ (new-data performance). Each subfigure shows the performance of a different model.

Figure 8. PSNR-Bpp curves for **data-incremental learning** experiments. In figure (a), the *"models, pre-trained"* curves overlap with the *"models w/ FT Enc."* curves because their decoder are the same.

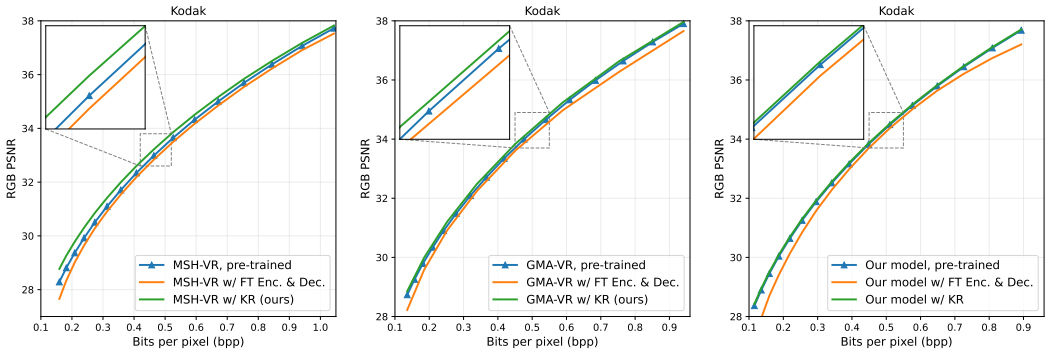


(a) Backward compatibility (bpp range is around $[0.1, 0.9]$). Each subfigure shows the performance of a different model.

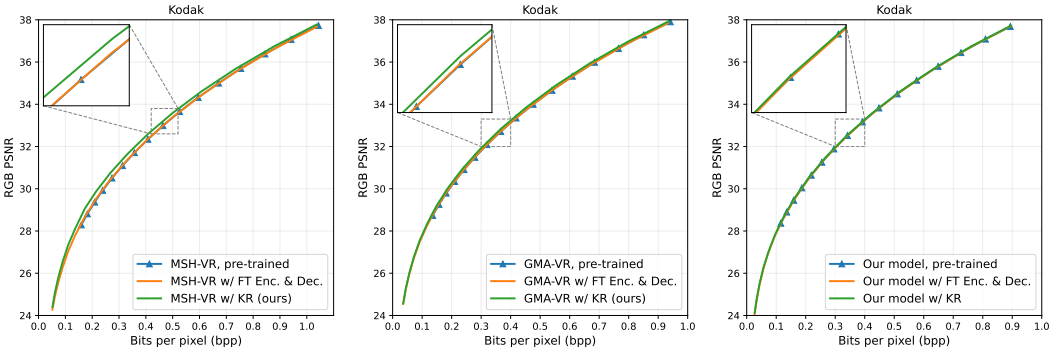


(b) New-rate performance (bpp range is around $[0.1, 1.6]$). Each subfigure shows the performance of a different model.

Figure 9. PSNR-Bpp curves for **rate-incremental learning (low → high)** experiments.



(a) Backward compatibility (bpp range is around $[0.1, 0.9]$). Each subfigure shows the performance of a different model.



(b) New-rate performance (bpp range is around $[0.03, 0.9]$). Each subfigure shows the performance of a different model.

Figure 10. PSNR-Bpp curves for **rate-incremental learning (high → low)** experiments.

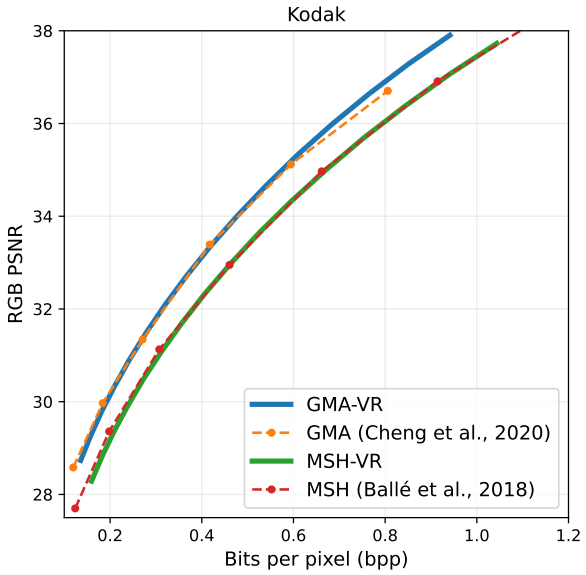


Figure 11. The variable-rate version of the baseline models that we constructed (MSH-VR and GMA-VR) are comparable to the original ones (MSH [41] and GMA [11]) in terms of PNSR-bpp performance on Kodak.

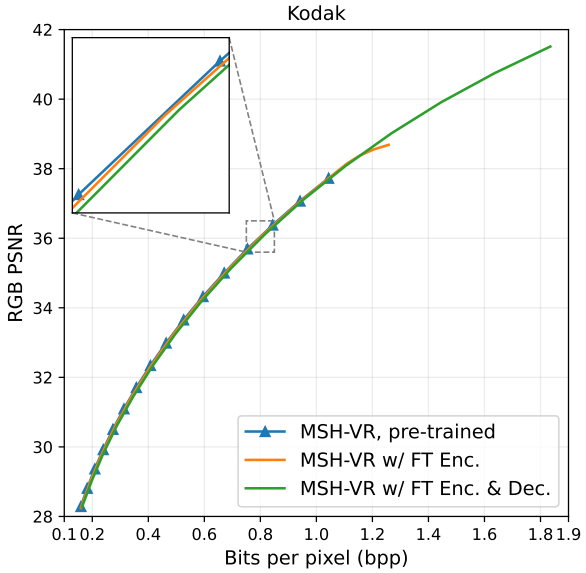


Figure 12. Fine-tuning the encoder does not effectively generalize the pre-train model (MSG-VR, for example) to new rates.

performance of the pre-trained MSH-VR, the one with fine-tuned encoder (*MSH-VR w/ FT Enc.*), and the one with fine-tuned encoder and decoder (*MSH-VR w/ FT Enc. & Dec.*). As shown in the figure, fine-tuning the encoder marginally extends the rate range of the pre-trained model, and the PSNR drops visibly when the rate is higher than maximum rate of the pre-trained model. Thus, we do not use this strategy in our rate-incremental learning experiments.