Image and Video Compression using Generative Sparse Representation with Fidelity Controls

Lebin Zhou, Wei Wang, and Wei Jiang

Futurewei Technologies Inc. San Jose, CA 95131, USA
{lzhou, rickweiwang, wjiang}@futurewei.com

Abstract. We propose a framework for learned image and video compression using the generative sparse visual representation (SVR) guided by fidelity-preserving controls. By embedding inputs into a discrete latent space spanned by learned visual codebooks, SVR-based compression transmits integer codeword indices, which is efficient and cross-platform robust. However, high-quality (HQ) reconstruction in the decoder relies on intermediate feature inputs from the encoder via direct connections. Due to the prohibitively high transmission costs, previous SVR-based compression methods remove such feature links, resulting in largely degraded reconstruction quality. In this work, we treat the intermediate features as fidelity-preserving control signals that guide the conditioned generative reconstruction in the decoder. Instead of discarding or directly transferring such signals, we draw them from a low-quality (LQ) fidelity-preserving alternative input that is sent to the decoder with very low bitrate. These control signals provide complementary fidelity cues to improve reconstruction, and their quality is determined by the compression rate of the LQ alternative, which can be tuned to trade off bitrate, fidelity and perceptual quality. Our framework can be conveniently used for both learned image compression (LIC) and learned video compression (LVC). Since SVR is robust against input perturbations, a large portion of codeword indices between adjacent frames can be the same. By only transferring different indices, SVR-based LIC and LVC can share a similar processing pipeline. Experiments over standard image and video compression benchmarks demonstrate the effectiveness of our approach.

Keywords: learned image compression · learned video compression · sparse visual representation · generative controls

1 Introduction

Image and video compression has been a decades-long research topic, and great success has been achieved recently by using neural networks (NN) for both learned image compression (LIC) [9, 22] and learned video compression (LVC) [14, 33]. Most existing LIC methods [9, 18, 22, 30, 40] use a hyperprior framework [3], which combines classical entropy coding with NN-based representation learning in a Variational AutoEncoder (VAE) structure. An entropy model is used to encode the quantized latent feature for easy transmission. Most existing

LVC methods follow the pipeline of traditional video coding [16,19], while replacing processing modules like motion estimation, motion compensation, residue coding *etc.* by learned NNs.

In this paper, we explore a different compression pipeline for both LIC and LVC, based on controlled generative modeling using the Sparse Visual Representation (SVR) (as shown in Fig. 1 and Fig. 2). We learn discrete generative priors as visual codebooks, and embed images into a discrete latent space spanned by the codebooks. By sharing the learned codebooks between the encoder and decoder, images can be mapped to integer codeword indices in the encoder, and the decoder can use these indices to retrieve the corresponding codewords' latent features for reconstruction.

The SVR-based compression has several benefits. (1) Transferring integer indices is very robust to heterogeneous platforms. One caveat of the hyperprior framework is the extreme sensitivity to small differences between the encoder and decoder in calculating the hyperpriors [4]. Even perturbations caused by floating round-off error can lead to catastrophic error propagation in the decoded latent feature. By encoding codeword indices instead of latent features, SVR-based compression does not suffer from such sensitivity. (2) Transferring indices gives the freedom of expanding latent feature dimension (often associated with better representation power for better reconstruction) without increasing bitrate, in comparison to transferring latent features or residues. (3) Generative SVR increases robustness to input degradations. Realistic and rich textures can be generated using hiqh-quality (HQ) codebooks even for low-quality (LQ) inputs.

However, SVR-based HQ restoration [7, 8, 25] relies on the dense connection of multi-scale features between the embedding network (encoder) and reconstruction network (decoder). Such intermediate features are too large to transfer, defeating the purpose of the compression task. As a result, previous SVR-based compression methods remove such direct feature links. By applying to specific content like human faces [20, 42], a code transformer is used to recover an aligned structured code sequence for HQ face restoration without intermediate features. For general images, M-AdaCode [21] compensates the performance loss of removed feature links by using data-adaptive weights to combine multiple semantic-class-dependent codebooks and uses weight masking to reduce transmitted weight parameters. However, although the restored images may look okay perceptually, important fidelity details are usually lost. As shown in Fig. 4, without direct feature links images generated by M-AdaCode often lack rich details.

In our opinion, the generative SVR-based reconstruction aims at high perceptual quality, and the multi-scale intermediate features provide complementary fidelity details to the reconstruction. Such details should NOT be ignored for applications like compression. Therefore, we focus on how to obtain effective and transmission-friendly fidelity information to balance bitrate and quality.

Our work is inspired by the success of ControlNet [38] where conditioning controls are used to guide image generation. We view the multi-scale intermediate features as fidelity-preserving control signals that guide the conditioned reconstruction in the decoder. As control conditions, such signals do NOT have

to come from the original input. Instead, we draw these control signals from an LQ alternative of the original input in decoder. This LQ alternative is computed in decoder based on highly compressed easy-to-transmit fidelity-preserving information, which is generated by fidelity-preserving methods like the previous NN-based or traditional image and video compression methods.

Based on this idea, we propose a framework (Fig. 2) that combines generative SVR-based restoration with fidelity-preserving compression. A highly compressed LQ alternative is transmitted with efficient bits, from which LQ control conditions are extracted to guide the reconstruction process. A conditioned generation network with weighted feature modulation is used to combine the SVR-based latent features with the LQ control features. The quality of the LQ control features is determined by the bitrate of the LQ alternative. The strength of the LQ control features in the conditioned generation process balances the importance between the HQ codebook and LQ fidelity details, which can be tuned based on the current reconstruction target to pursue high perceptual quality or high fidelity. As shown in Fig. 4, with our LQ control features, the restored results have largely improved fidelity with rich details.

In addition, we extend the SVR-based LIC framework into an effective LVC framework. Since SVR is robust against input degradation and small perturbations, a substantial amount of codeword indices between adjacent frames can be the same (47% in our experiments). We only need to transfer different indices for most frames. No motion estimation or motion compensation is involved and there is no error propagation. Comparing to previous LIC and LVC, our SVR-based LIC and LVC share a similar processing pipeline, which makes it possible to simplify industrial productive optimization.

We evaluate our approach using benchmark datasets for image and video standardization. Specifically, SVR-based LIC is tested over the JPEG-AI dataset [2]. SVR-base LVC is tested over a combined dataset comprising of video sequences from AOM [1], MPEG [17], JVET [34], and AVS [15]. Also, we evaluate the performance of different SVR-based restoration methods, based on a single codebook [8] or multiple codebooks [25]. Experimental results demonstrate the effectiveness of our method.

2 Related Works

2.1 Sparse Visual Representation Learning

Discrete generative priors have shown impressive performance in image restoration tasks like super-resolution [8], denoising [11] *etc.* By embedding images into a discrete latent space spanned by learned visual codebooks, SVR improves robustness to various degradations. For instance, VQ-VAE [28] learns a highly compressed codebook by a vector-quantized VAE. VQGAN [11] further improves restoration quality by using GAN with adversarial and perceptual loss. In general, natural images have very complicated content, and it is difficult to learn a single class-agnostic codebook for all image categories. Therefore, most methods

focus on specific categories. In particular, great success has been achieved in face generation due to the highly structured characteristics of human faces [37, 42].

For general images, the recent AdaCode [25] uses image-adaptive codebook learning. Instead of learning a single codebook for all categories of images, a set of basis codebooks are learned, each corresponding to a semantic partition of the latent space. A weight map to combine such basis codebooks is adaptively determined for each input image. By learning the semantic-class-guided codebooks, the semantic-class-agnostic restoration performance can be largely improved.

2.2 Learned Image Compression

There are two main research topics for LIC: how to learn a latent representation, and how to quantize and encode the latent representation. One most popular framework is based on hyperpriors [3], where the image is transformed into a dense latent feature, and an entropy model encodes/decodes the quantized latent feature for efficient transmission. Many improvements have been made to improve the transformation for computing the latent [9, 27, 43], the entropy model [13, 27, 29], or the quantization strategy [30, 40].

One vital issue of the hyperprior framework is the extreme sensitivity to small differences between the encoder and decoder in calculating the hyperpriors [4]. Most works simply assume homogeneous platforms and deterministic CPU calculation. Some work uses integer NN to prevent non-deterministic GPU computation [4] or designs special NN module that is computational friendly to CPU [41]. However, such solutions cannot be easily generalized to arbitrary network architectures. Also, it is well known that there are complex relations among bitrate, distortion, and perceptual quality [5, 6], and it is difficult to pursue high perceptual quality and high pixel-level fidelity at the same time.

2.3 Learned Video Compression

Existing LVC methods [14, 26, 31, 33] follow the traditional video coding pipeline by replacing processing modules like motion estimation, motion compensation, post-enhancement by NNs. Generally the independent (I) frames in a GoP (group of pictures) are compressed as images, and the predictive (P) frames and the bidirectional predictive (B) frames are compressed based on motion estimation and residue coding. This pipeline is not designed for LVC, resulting in error accumulation from different modules. Also the computation cost is generally very high due to the complicated framework.

2.4 SVR-based Compression

SVR is intuitively suitable for compression, since the integer codeword indices are easy to transfer and are robust to small computation differences in heterogeneous hardware and software platforms. However, HQ SVR-based restoration relies on direct links of multi-scale features between the encoder and decoder. Such

features are too expensive to transfer, which often cost more bits than the original input. To make SVR-based compression feasible, previous approaches remove such feature connections. For example, when applied to specific categories like aligned human faces [20,36,42], it is possible to predict a cohesive code sequence for HQ restoration without direct feature links. However, for general images the reconstruction quality is severely impacted without such features. As a result, most methods focus on very low-bitrate scenarios [10], where reconstruction with low fidelity yet good perceptual quality is tolerated. The recent M-AdaCode [25] compensates the performance loss of the removed feature links by using data-adaptive weights to combine multiple semantic-class-dependent codebooks and trades off bitrate and distortion by weight masking to reduce transmitted weight parameters. Unfortunately, for general image content, without the feature connections the restoration quality is overall unsatisfactory.

3 SVR-based Compression with Conditional Controls

The general framework of SVR-based restoration can be summarized in Fig. 1. An input image $X \in \mathbb{R}^{w \times h \times c}$ is embedded into a latent space as latent feature $Y \in \mathbb{R}^{u \times v \times d}$ by an embedding network E^{emb}. Using a learned codebook $\mathcal{C} = \{c_l \in \mathbb{R}^d\}$, the latent Y is further mapped into a discrete quantized latent feature $Y^q \in \mathbb{R}^{u \times v \times d}$. Each super-pixel $y^q(l)$ $(l = 1, \ldots, u \times v)$ in Y^q corresponds to a codeword $c_l \in \mathcal{C}$ that is closest to the corresponding latent feature $y(l)$ in Y:

$$y^q(l) = c_l = argmin_{c_i \in \mathcal{C}} D(c_i, y(l))).$$

$y^q(l)$ can be represented by the index z_l of codeword c_l, and the entire Y^q can be mapped to an n-dim vector Z of integers, $n = u \times v$. Based on indices Z, the quantized feature Y^q can be retrieved from the codebook $\mathcal{C}$. Also, multi-scale features F are computed from several downsampling blocks in E^{emb}, which are fed to the corresponding upsampling blocks in a reconstruction network E^{rec} as residual inputs. E^{rec} then reconstructs the output image $\hat{x}$ based on the quantized latent Y^q and features F.

To improve the performance for general image restoration, instead of using one codebook as in [24, 42], AdaCode [25] learns a set of basis codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_K$, each corresponding to a semantic partition of the latent space. A weight map $W \in \mathbb{R}^{u \times v \times K}$ is computed to combine the basis codebooks for adaptive restoration. The quantized latent Y^q is a weight combination of individual quantized latents $Y_1^q, \ldots, Y_K^q$ using each of the basis codebooks:

$$y^q(l) = \sum\nolimits_{j=1}^{K} w_j(l) y_j^q(l), \tag{1}$$

and $w_j(l)$ is the weight of the j-th codebook for the l-th super-pixel in W.

For the purpose of compression, previous methods [20, 21] remove the direct skip connections of the multi-scale features F that are too heavy to transfer. Only the indices $Z_1, \ldots, Z_K$ are sent to the decoder to retrieve the quantized latent

Y^q for reconstruction. However, the multi-scale features F provide important fidelity details of the input, and without F the reconstructed result may lack details and may not be consistent with the original input.

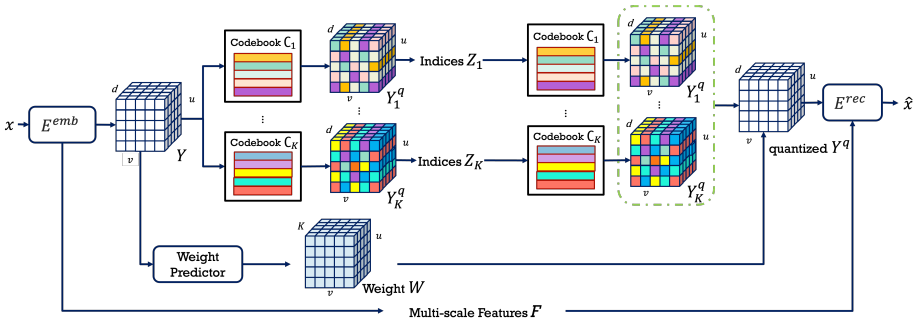
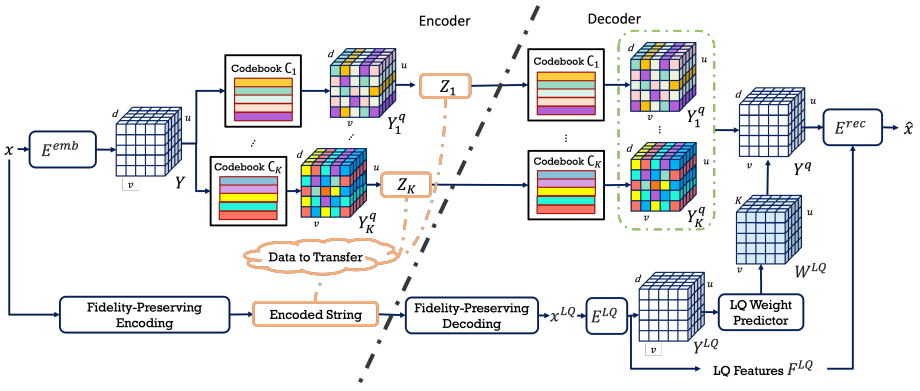


Fig. 1: The general workflow of SVR-based restoration. Discarding the multi-scale features F will sacrifice restoration quality significantly.

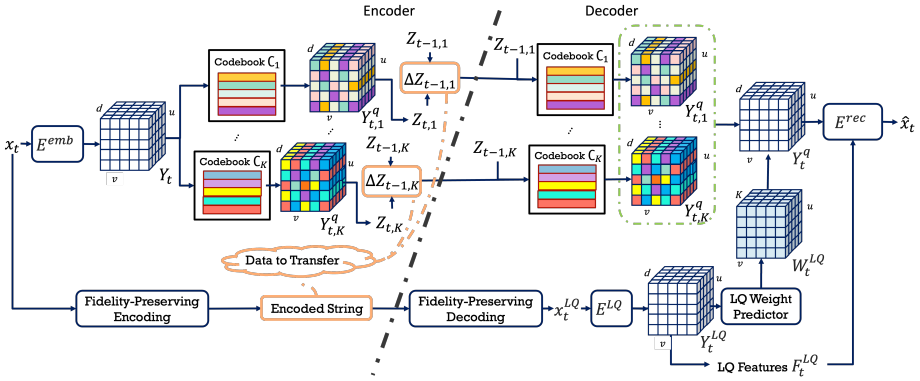
3.1 SVR-based Compression using LQ Control Conditions

In the realm of image generation, ControlNet [38] has been developed to enable different levels of control over generated results. The key idea is to provide data-specific conditions to a pre-trained generative model to control the generation process. This is analogous to SVR-based compression, where the reconstruction network E^{rec} generates the output based on quantized feature Y^q, and the multi-scale features F provide additional control conditions drawn from the current input. This perspective motivates our compression framework in Fig. 2. When used as control conditions, the multi-scale features do not have to come from the original input, and therefore we can avoid transmitting the heavy F. Instead, they can be drawn from an LQ substitute of the input x^{LQ} in the decoder, and the LQ substitute can be computed in decoder based on fidelity-preserving information calculated by existing compression methods like [19, 22] with high compression rates and low bitrate. With the help of additional controls, this framework not only improves the restoration fidelity and quality, but also enables flexible quality control. By tuning the bitrate of the LQ substitute, we can tune the quality of the LQ substitute and change the quality of the control condition.

SVR-based LIC As shown in Fig. 2 (a), in the encoder, the input image x is embedded into the latent space as latent feature Y, which is further quantized into $Y_1^q, \ldots, Y_K^q$ with associated codeword indices $Z_1, \ldots, Z_K$, by using codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_K$, respectively. At the same time, x is encoded into a highly compressed string with low bitrate using a fidelity-preserving image compression method (*e.g.*, an existing LIC method [10]), which is transferred to the



(a) Image compression pipeline



(b)Video compression pipeline

Fig. 2: The proposed SVR-based compression framework using LQ control conditions.

decoder together with indices $Z_1, \ldots, Z_K$. Then in the decoder, the quantized latents $Y_1^q, \ldots, Y_K^q$ are retrieved from the corresponding codebooks using the codeword indices, and the LQ substitute x^{LQ} is decoded from the corresponding image compression method. Albeit low quality, x^{LQ} carries important fidelity information about the original x to guide reconstruction.

In the decoder, x^{LQ} is fed into an LQ embedding network E^{LQ} to compute the multi-scale LQ features F^{LQ} from the multiple downsampling blocks in E^{LQ}, where F^{LQ} has the same size as the original multi-scale features F (if computed from the original E^{emb}). When $K > 1$, an LQ weight map $W^{LQ} \in \mathbb{R}^{u \times v \times K}$ is also computed by an LQ weight predictor. Both F^{LQ} and W^{LQ} are control conditions drawn from the fidelity-preserving LQ substitute x^{LQ}, which are used by the reconstruction network E^{rec} to guide the reconstruction process. Since F^{LQ} and W^{LQ} are calculated in decoder using x^{LQ}, they do not increase bitrate.

In detail, to reconstruct the output, when $K > 1$, the final quantized latent Y^q is a weighted combination of $Y_1^q, \ldots, Y_K^q$ similar to Eqn. (1), using LQ weight

W^{LQ}. When $K=1$, $Y_1^q = Y^q$. Then the multi-scale LQ feature F^{LQ} is used as modulating conditions to the multiple upsampling blocks in E^{rec} to guide the reconstruction from Y^q. In this work, we use the Controllable Feature Transformation (CFT) module from [42] to apply modulating conditions. Let Θ_c denote the parameters of the CFT module CFT_{code} to combine the codebook-based quantized feature Y^q and the LQ control feature F^{LQ}. F^{LQ} tunes Y^q into a modulated $Y^{mod} = Y^q + \alpha_c * (\beta_c * Y^q + \gamma_c)$, where β_c, γ_c are affine parameters $\beta_c, \gamma_c = \Theta_c(concat(Y^q, F^{LQ}))$, and $concat(\cdot)$ is the concatenation operation. α_c determines the strength of the control feature F^{LQ} in conditioning the codebook-based feature Y^q, which can be flexibly set according to the actual compression needs, *i.e.*, to pursue high perceptual quality or high fidelity.

One advantage of our SVR-based LIC method is the flexibility in accommodating different scenarios. For homogeneous computing platforms, our method combines the strength of the fidelity cue from existing fidelity-preserving image compression methods (classic or learning-based) and the perceptual cue from SVR-based restoration, enables bitrate control by tuning the bitrate of the LQ substitute, and allows tradeoff between perceptual quality and fidelity. For heterogeneous computing platforms where previous LIC methods may have difficulty to apply, our method can still give a decent low-bitrate baseline reconstruction with good perceptual quality using SVR-based restoration alone, or can pair with classic compression methods for improved reconstruction.

SVR-based LVC The above SVR-based LIC method can be easily extended to an effective SVR-based LVC method, whose workflow is shown in Fig. 2 (b). Since SVR is robust to input degradation and perturbations, for most frames in a video, a large portion of the codeword indices can be the same between adjacent frames. Therefore, for a video frame x_t at time stamp $t > 1$, only the different indices $\Delta Z_{t,1}, \ldots, \Delta Z_{t,K}$ from the previous frame need to be transmitted for the decoder to restore the quantized latent Y_t^q for SVR-based reconstruction. Actually our experiments show that 47% of the codeword indices remain unchanged on average, leading to effective bit reduction for LVC. The remaining processing modules are similar to the LIC method, with the difference that the LQ substitute x_t^{LQ} of frame x_t comes from a fidelity-preserving video compression method (*e.g.*, classic VVC [19] or learning based DVC [26]) or a fidelity-preserving image compression method.

Similar to SVR-based LIC, our SVR-based LVC method provides flexibility to accommodate different scenarios, where we can choose different methods to generate the LQ substitutes by considering different factors like computation and transmission requirements, reconstruction targets, *etc.* In addition, there is no error propagation in recovering the codeword-based quantized feature for every frame, and a decent low-bitrate baseline with good perceptual quality can be mostly guaranteed. Furthermore, in comparison to previous LIC and LVC methods that usually have completely different processing pipelines, the SVR-based LIC and LVC have a similar workflow with similar processing modules, making it possible to simplify industrial productive optimization.

It is worth mentioning that in our implementation the reconstruction network E^{rec} has the same architecture for both LIC and LVC. A video-oriented network like C3D [35] can be used for LVC to better ensure temporal consistency. We found it unnecessary in experiments as the result is quite consistent temporally due to the deterministic generation and temporal-consistent fidelity control.

Complexity Our approach is very efficient in computation. For SVR-based LIC, our encoding time includes inference through E^{emb} with time $\mathcal{T}(E^{emb})$ and compressing for X^{LQ} with time $\mathcal{T}(Enc(X^{LQ}))$. In comparison, previous LIC methods compress for the original X, and the encoding time $\mathcal{T}(Enc(X))$ includes both the embedding inference $\mathcal{T}(E^{emb})$ and the hyperprior coding time $\mathcal{T}(Enc(X_{hyper}))$. Usually $\mathcal{T}(Enc(X_{hyper}))$ is much larger than $\mathcal{T}(E^{emb})$ due to the expensive autoregressive process using CPU. Since our X^{LQ} can be highly compressed, our $\mathcal{T}(Enc(X^{LQ}))$ is much less than $\mathcal{T}(Enc(X_{hyper}))$ (*e.g.*, by using a much smaller embedding latent and hyperprior to encode to largely reduce bitrate and computation).

Similarly, our decoding time mainly includes inference through E^{rec} with $\mathcal{T}(E^{rec})$, decoding for X^{LQ} with $\mathcal{T}(Dec(X^{LQ}))$, and inference through E^{LQ} with $\mathcal{T}(E^{LQ})$. In comparison, previous LIC methods needs to decode for the original X where the decoding time includes $\mathcal{T}(E^{rec})$ and $\mathcal{T}(Dec(X))$. Again, due to the expensive hyperprior decoding process, $\mathcal{T}(Dec(X))$ is usually much larger than $\mathcal{T}(Dec(X^{LQ}))$ and $\mathcal{T}(E^{LQ})$ combined.

For SVR-based LVC, our computation is basically a linear extension of the computation for SVR-based LIC according to the number of frames. This is much more efficient than previous LVC methods, which not only require multiple inference processes through multiple networks, but also require the entire decoding computation in the encoder to obtain residues.

3.2 Training strategy

Stage 1: Pretrain The embedding network E^{emb}, codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_K$, reconstruction network E^{rec}, and weight predictor (for $K>1$) are pretrained for single-codebook-based restoration [8] or multi-codebook-based restoration [25].
Stage 2: Train SVR-based LIC The embedding network E^{emb} and codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_K$ are fixed, and we train the LQ embedding network E^{LQ}, the CFT module CFT_{code}, the LQ weight predictor (for $K>1$), the reconstruction network E^{rec}, and the GAN discriminator for the LIC pipeline. The training loss comprises of pixel-level L_1 loss and SSIM, the perceptual loss [23], LPIPS [39], and the GAN adversarial loss [12]. The straight-through gradient estimation is used for back-propagation through the non-differentiable vector quantization process during training. The strength of control is set as $\alpha_c=1$ for all inputs.
Stage 3: Train SVR-based LVC The embedding network E^{emb}, the codebooks $\mathcal{C}_1, \ldots, \mathcal{C}_K$, and the reconstruction network E^{rec} are fixed, and we train the LQ embedding network E^{LQ}, the CFT module CFT_{code}, the LQ weight predictor (when $K>1$), and the GAN discriminator by finetuning from the corresponding LIC version in the previous stage. One benefit of funetuning from

the LIC counterparts is to benefit from the large variety of image training content to avoid overfitting, due to the limited amount of training videos from the standardization community.

3.3 Bit reduction for integer codeword indices

To transfer codeword indices, naively we need $b(Z_k) = u \times v \times \text{floor}(\log_2 n_k)$ bits for each codebook $\mathcal{C}_k$ of size n_k. This number can be further reduced to save bit consumption of the whole system. We propose an effective arithmetic coding method that can losslessly compress the integer indices by 5× on average. For natural images, codewords normally show up with different frequencies. For instance, codewords of natural scenes may be used more frequently than those of human faces. We can assign less bits to more frequently used indices to reduce the total bitrate. Specifically, we first calculate the frequency of codewords' usage in training data and reorder the codewords in descending order. Then for each particular indices string of each datum, we convert each odd index ix to a negative integer as $ix^* = -(ix+1)/2$ and rescale even indices by $1/2$. Such operations transform the indices distribution to a Gaussian style bell shape, which can be efficiently encoded by Gaussian Mixture-based arithmetic coding [32].

4 Experiments

Datasets We tested the proposed SVR-based LIC and LVC method, respectively, over the JPEG-AI dataset [2,18] and a mixed video dataset combining test video sequences from several video compression standards including AOM [1], MPEG [17], JVET [34], and AVS [15]. The JPEG-AI dataset had 5664 images with a large variety of visual content and resolutions up to 8K. The training, validation, and test set had 5264, 350, and 50 images, respectively. The mixed video set contained 150 videos, which were used as test sequences by the standardization community. We removed the duplicate sequences, *e.g.*, the same sequences used by different standards, or the same sequences resized to different resolutions, where we kept videos with different resolutions ranging from 240×400 to 4K. 134 and 16 video sequences were used for training and test respectively.

The training patches was 256×256, randomly cropped from randomly resized training images or video frames, augmented by random flipping and rotation. For evaluation, the maximum inference tiles was 1080×1080. For training SVR-based LVC modules, video frames were randomly sampled from videos, and were used as images in the same way as training SVR-based LIC modules. For all tested methods, each training stage had 500K iterations with Adam optimizer and a batch size of 32, using 8 NVIDIA Tesla V100 GPUs. The learning rate for the generator and discriminator were fixed as 1e-4 and 4e-4, respectively.

Evaluation Metrics For reconstruction distortion, we measured PSNR and SSIM, as well as the perceptual LPIPS [39]. The bitrate was measured by bpp (bit-per-pixel): $bpp = B/(h \times w)$, and the overall bits $B = b_c + b_{LQ}$ consisted of b_c for

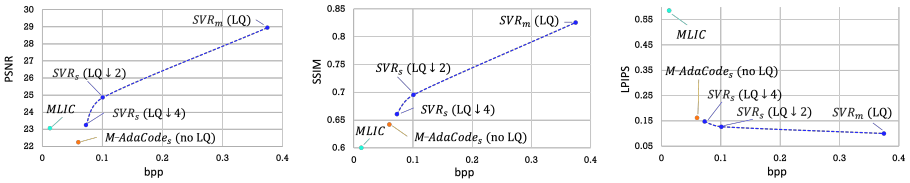


Fig. 3: Rate-distortion performance for image compression.

sending codebook indices and b_{LQ} for sending the encoded string to compute the LQ substitute x^{LQ} using previous image/video compression methods. In detail, for LIC $b_c = \sum_{k=1}^{K} b(Z_k)$, and for LVC $b_c = \sum_{k=1}^{K}[b(Z_{1,k}) + \sum_{t=2}^{T} b(\Delta Z_{t,k})]/T$. $b(Z_k)$ is computed based on the indices reduction method described in Sec. 3.3. In terms of b_{LQ}, it determined the quality of the LQ substitute x^{LQ}. We chose a low b_{LQ} (< 0.1 bpp) to roughly match b_c.

Evaluated Methods We evaluated two configurations for our approach using SOTA SVR-based restoration algorithms: the single-codebook-based FeMaSR [8] and the multi-codebook-based AdaCode [25]. With only a single codebook, FeMaSR gave very low bitrate. Using multiple codebooks, AdaCode gave improved restoration quality but consumed more bits.

To generate fidelity-preserving LQ substitute x^{LQ}, for SVR-based LIC we used previous SOTA LIC method MLIC [22]. The pre-trained MLIC model with the lowest available bitrate setting was used, which corresponded to the quality-1 model in [22]. In order to get lower bitrate for b_{LQ} to match b_c, we first downsampled the input x by 2× or 4×, and then used MLIC to encode the downsampled input and then upsampled the decoded x^{LQ} back to the original size. The bicubic filter was used for downsampling/upsampling. For SVR-based LVC we used the SOTA VVC video compression method [19] with $qp = 42$ to generate x^{LQ}, which gave reasonable low-bitrate reconstruction in general.

4.1 LIC Results

Fig. 3 gives the rate-distortion performance for image compression. For our SVR-based LIC, we tested 3 different settings: single-codebook SVR with 2× and 4× downsampled-upsampled x^{LQ} as "SVR$_s$(LQ↓2×)" and "SVR$_s$(LQ↓4×)", and multi-codebook SVR with x^{LQ} without downsampling-upsampling SVR$_m(LQ)$. We also compared with M-AdaCode without x^{LQ} with the 1-codebook setting [21] ("M-AdaCode$_s$(no LQ)") and compared with MLIC [22] generated x^{LQ}. From the figures, "SVR$_s$(LQ↓4×)" outperformed M-AdaCode with $1dB$, 2.3% and 8.9% improvements over PSNR, SSIM and LPIPS, respectively, using only

a 0.013*bpp* increase. Compared to MLIC, "SVR$_s$(LQ↓4×)" improved LPIPS and SSIM by 74.8%, and 10.1% respectively. Among methods having <0.1*bpp*, our SVR-based LIC gave balanced results with good fidelity and perceptual quality.

Fig. 4 gives some restoration examples, which clearly show the strength of our method. With the help of x^{LQ}, our SVR-based LIC largely improved the reconstruction fidelity and perceptual quality with rich visually pleasing details, compared to M-AdaCode$_s$(no LQ)". Also, our framework can be flexibly configured to different settings to tradeoff bitrate and reconstruction quality.

4.2 LVC Results

For video compression, we tested the single-codebook SVR$_s$. The LQ substitute x^{LQ} was generated by the VVC standard [19] using $qp=42$ over the original resolution. This is basically the lowest bpp configuration of VVC ($bpp=0.06$) with a reasonable reconstruction quality for x^{LQ}. Tab. 1 gives the rate-distortion performance, and Fig. 5 gives some restoration examples. Our "SVR$_s$(LQ)" achieved much better perceptual quality with a 64.8% improvement over LPIPS given only 0.035*bpp* increase. As expected, improvements over PSNR and SSIM are less significant as VVC is tailored to optimize such pixel-level distortions. With overall $bpp<0.1$, the SVR-based LVC can generate rich visually pleasing details compared to the overly smoothed results from VVC. On average, 47% of codeword indices remain unchanged, verifying the effectiveness of using the similar pipeline for both SVR-based LIC and LVC.

Table 1: SVR-based LVC performance

	PSNR	SSIM	LPIPS	bpp
SVR$_s$(LQ)	28.15	0.812	0.109	0.095
VVC (x^{LQ})	28.09	0.806	0.310	0.06

5 Conclusions

We proposed a general SVR-based compression framework for both LIC and LVC. Based on the idea of guided image generation with conditional controls, our method drew fidelity cues as control signals from a low-bitrate LQ version of the original input to guide reconstruction. Compared with previous approaches that relied on SVR-based generation alone, the fidelity cues largely improved the reconstruction quality. By tuning the bitrate of the LQ input, we could trade off bitrate, reconstruction fidelity and perceptual quality. By transferring the difference of codeword indices between adjacent frames, a similar processing pipeline was used for both SVR-based LIC and LVC. Experimental results showed improved performance over SOTA image and video compression methods.

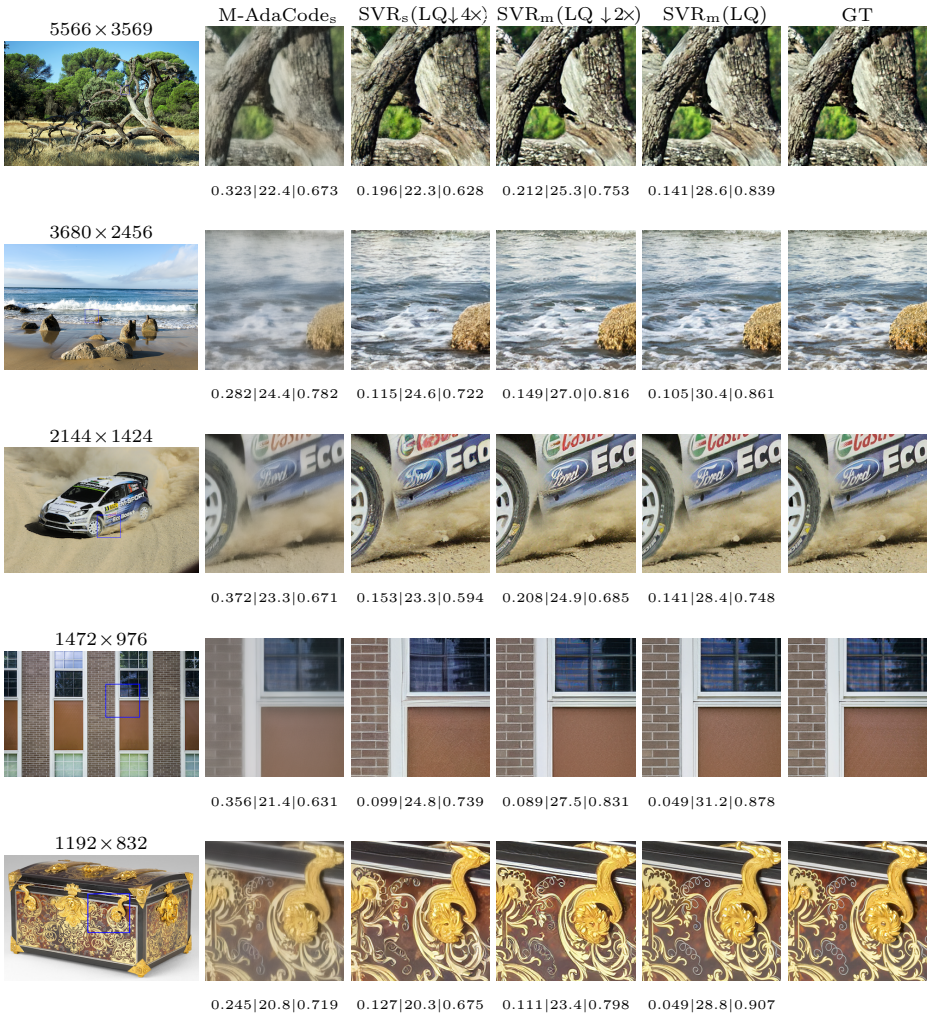


Fig. 4: "LPIPS|PSNR|SSIM" under each result. "SVR$_m$"/ "SVR$_s$": multi-codebook-based/single-codebook-based SVR. "LQ"/"LQ↓2×"/"LQ↓4×": x^{LQ} of original size/x^{LQ} with 2× downsampling-upsampling /x^{LQ} with 4× downsampling-upsampling.

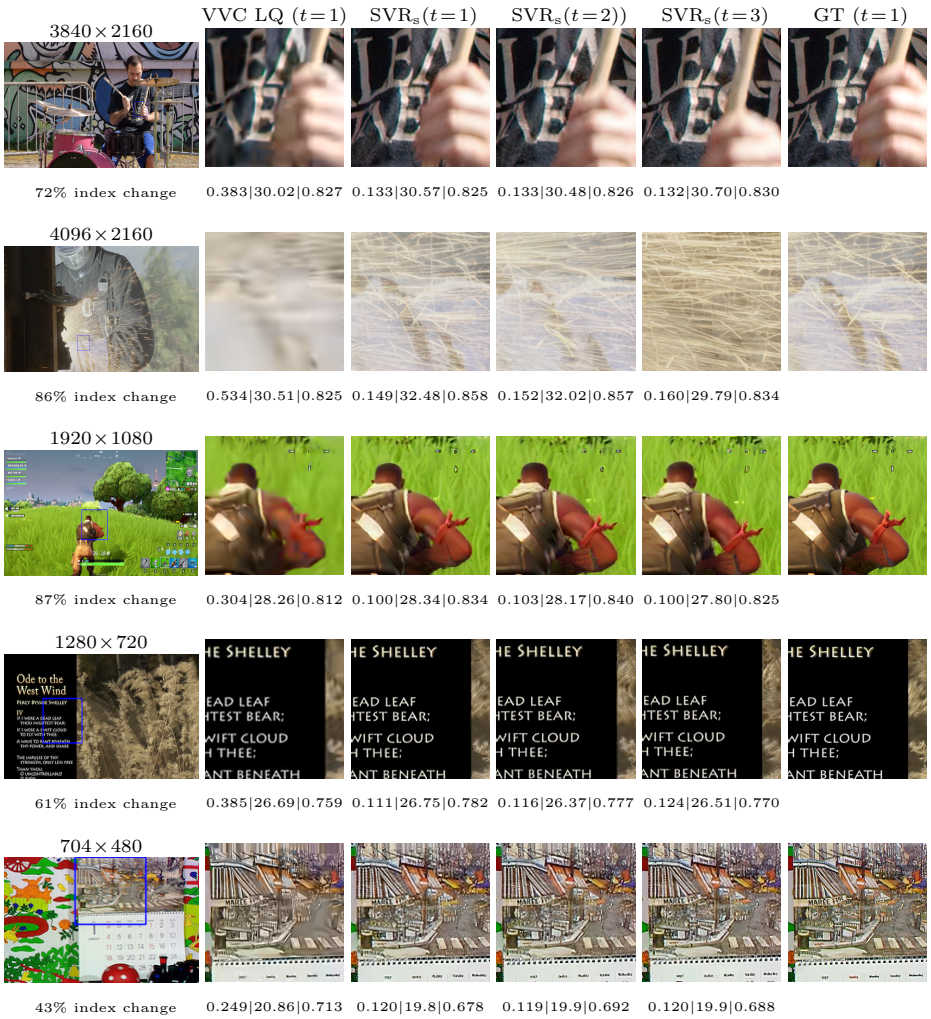


Fig. 5: "LPIPS|PSNR|SSIM" under each result. Single-codebook SVR was used. x^{LQ} was computed by VVC with $qp = 42$. The average $b_{LQ} = 0.06$ and $b_c = 0.035$. Our SVR-based LVC can largely improve reconstruction fidelity and perceptual quality.

References

1. Alliance for open media, press Release: http://aomedia.org/press-release/, Git repositories on aomedia: https://aomedia.googlesource.com/
2. Ascenso, J., Akyazi, P., Pereira, F., Ebrahimi, T.: Learning-based image coding: early solutions reviewing and subjective quality evaluation. SPIE Photonics Europe - Optics, Photonics and Digital Technologies for Imaging Applications VI (2020)
3. Balle, J., Minnen, D., Singh, S., Hwang, S., Johnston, N.: Variational image compression with a scale hyperprior. ICLR (2018)
4. Ballé, J., Johnston, N., Minne, D.: Integer networks fro data compression with latent-variable models. ICLR (2019)
5. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. CVPR pp. 6228–6237 (2018)
6. Blau, Y., Michaeli, T.: Rethinking lossy compression: The rate-distortion-perception tradeoff. ICML pp. 675–685 (2019)
7. Chan, K., Wang, X., Xu, X., Gu, J., Loy, C.: GLEAN: Generative latent bank for large-factor image super-resolution. CVPR (2021)
8. Chen, C., Shi, X., Qin, Y., Li, X., Han, X., Yang, T., Guo, S.: Real-rorld blind super-resolution via feature matching with implicit high-resolution priors. ACM Multimedia (2022)
9. Cheng, Z., Sun, H., Takeuchi, M., Katto, J.: Learned image compression with discretized gaussian mixture likelihoods and attention modules. CVPR (2020)
10. El-Nouby, A., Muckle, M., Ullrich, K., Laptev, I., Verbeek, J., Jegou, H.: Image compression with product quantized masked image modeling. arXiv preprint: arXiv:2212.07372 (2022)
11. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. CVPR (2021)
12. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS (2014)
13. He, D., Zheng, Y., Sun, B., Wang, Y., Qin, H.: Checkerboard context model for efficient learned image compression. CVPR (2021)
14. Hu, Z., Lu, G., Xu, D.: Fvc: A new framework towards deep video compression in feature space. CVPR (2021)
15. IEEE: Audio video coding standard. IEEE Standard for Second-Generation IEEE 1857 Video Coding, ISBN 978-1-5044-5461-2 (2019)
16. Int. Telecommun. Union-Telecommun. (ITU-T) and Int. Standards Org./Int/Electrotech. Commun. (ISO/IEC JTC 1): High efficiency video coding, rec. ITU-T H.265 and ISO/IEC 23008-2, 2019
17. ISO/IEC JTC 1/SC 29: Moving picture experts group (mpeg)
18. ISO/IEC JTC 1/SC29/WG1: Report on the jpeg ai call for proposals results. ISO/IEC JTC1/SC29 WG1, N100250 (July 2022)
19. ITU-T and ISO: Versatile video coding, rec. ITU-T H.266 and ISO/IEC 23090-3, 2020
20. Jiang, W., Choi, H., Racapé, F.: Adaptive human-centric video compression for humans and machines. CVPRW on NTIRE (June 2023)
21. Jiang, W., Wang, W., Chen, Y.: Neural image compression using masked sparse visual representation. WACV (2024)
22. Jiang, W., Yang, J., Zhai, Y., Ning, P., Gao, F., Wang, R.: Mlic: Multi-reference entropy model for learned image compression. ACM Multimedia (2023)

23. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. ECCV (2016)
24. Li, T., Chang, H., Mishra, S., Zhang, H., Katabi, D., Krishna, D.: Mage: Masked generative encoder to unify representation learning and image synthesis. CVPR (2023)
25. Liu, K., Jiang, Y., Choi, I., Gu, J.: Learning image-adaptive codebooks for class-agnostic image restoration. ICCV (2023)
26. Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., Z.Gao: DVC: An end-to-end deep video compression framework. CVPR pp. 11006–11015 (2019)
27. Mentzer, F., Toderici, G., Minnen, D., Hwang, S., Caelles, S., Lucic, M., Agustsson, E.: Vct: A video compression transformer. arXiv preprint: arXiv:2206.07307 (2022)
28. Oord, A., Vinyals, O., Kavukcuoglu, K.: Neural discrete representation learning. NeurIPS (2017)
29. Qian, Y., Lin, M., Sun, X., Jin, Z.T.R.: Entroformer: A transformer-based entropy model for learned image compression. arXiv preprint: arXiv:2202.05492 (2022)
30. R. Feng, Z. Guo, W.L.Z.C.: Nvtc: Nonlinear vector transform coding. CVPR (2023)
31. Rippel, O., Nair, S., Lew, C., Branson, S., Anderson, A., Bourdev, L.: Learned video compression. ICCV (2019)
32. Rissanen, J., Langdon, G.: Arithmetic coding. IBM Journal of research and development **23**(2) (1979), https://doi.org/10.1147/rd.232.0149
33. Shi, Y., Ge, Y., Wang, J., , Mao, J.: Alphavc: high-performance and efficient learned video compression. ECCV (2022)
34. Suehring, K., Li, X.: Jvet common test conditions and software reference configurations. document JVET-B1010 of JVET (2016)
35. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. ICCV (2015)
36. Wang, T., Mallya, A., Liu, M.: One-shot free-view neural talking-head synthesis for video conferencingr. CVPR (2021)
37. Wang, Z., Zhang, J., Chen, R., Wang, W., Luo., P.: Restoreformer: High-quality blind face restoration from undegraded key-value pairs. CVPR (2022)
38. Zhang, L., Rao, A., Agrawalau, M.: Adding conditional control to text-to-image diffusion models. ArXiv, arXiv:2302.05543 (2023)
39. Zhang, R., Isola, P., Efros, A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. CVPR (2018)
40. Zhang, X., Wu, X.: Lvqac: Lattice vector quantization coupled with spatially adaptive companding for efficient learned image compression. CVPR (2023)
41. Zheng, Z., Wang, X., Lin, X., Lv, S.: Get the best of the three worlds: Real-time neural image compression in a non-gpu environment. ACM Multimedia (2021)
42. Zhou, S., Chan, K., Li, C., Loy, C.: Towards robust blind face restoration with codebook lookup transformer. NeurIPS (2022)
43. Zou, R., Song, C., Zhang, Z.: The devil is in the details: Window-based attention for image compression. CVPR (2022)