

# Adaptive Human-Centric Video Compression for Humans and Machines

Wei Jiang Hyomin Choi Fabien Racapé  
Emerging Technologies Lab, InterDigital Inc.

{wei.jiang,hyomin.choi,fabien.racape}@interdigital.com

## Abstract

We propose a novel framework to compress human-centric videos for both human viewing and machine analytics. Our system uses three coding branches to combine the power of generic face-prior learning with data-dependent detail recovery. The generic branch embeds faces into a discrete code space described by a learned high-quality (HQ) codebook, to reconstruct an HQ baseline face. The domain-adaptive branch adjusts reconstruction to fit the current data domain by adding domain-specific information through a supplementary codebook. The task-adaptive branch derives assistive details from a low-quality (LQ) input to help machine analytics on the restored face. Adaptive weights are introduced to balance the use of domain-adaptive and task-adaptive features in reconstruction, driving trade-offs among criteria including perceptual quality, fidelity, bitrate, and task accuracy. Moreover, the proposed online learning mechanism automatically adjusts the adaptive weights according to the actual compression needs. By sharing the main generic branch, our framework can extend to multiple data domains and multiple tasks more flexibly compared to conventional coding schemes. Our experiments demonstrate that at very low bitrates we can restore faces with high perceptual quality for human viewing while maintaining high recognition accuracy for machine use.

## 1. Introduction

We investigate efficient compression of human-centric videos. Specifically, this work focuses on videos where human faces are the main regions of interest. Such “face-centric” videos can be largely found in video conferencing, video surveillance, *etc.* These applications require high performance compression, especially at low bitrates. Since human faces are the main focus, an optimized codec that exploits the highly structured characteristics of human faces should be more effective than off-the-shelf standard codecs [10, 12] or learned codecs using neural networks (NN) [6, 18, 19] that are designed as versatile methods for compressing general video content.

Depending on the application, the requirements of compressed faces may vary. For example, for human viewing the restored video should look realistic and perceptually pleasant. For content analytics such as detection and recognition, identity or fidelity cues should be restored to be analyzed by machine. Traditionally, videos are compressed for human viewing, leading to compression induced degradation that can severely affect machine analysis. Due to the complex relations among bitrate, distortion, and perceptual quality [2, 3], previous methods customize a video coding algorithm to optimize the compression efficiency either for human use or for machine use. For example, for video conferencing, NVIDIA proposed the Maxine solution based on face reenactment [23, 24]. For machine analytics, the recent MPEG Video Coding for Machine (VCM) [17] and JPEG AI [11] standardization activities optimize compression for detection, segmentation or tracking tasks. It is non-trivial to generalize the individually customized video coding methods for both human and machine use, or to perform multiple tasks. Furthermore, when the data domain changes, *e.g.*, from HD broadcast videos to webcam surveillance videos, a new set of model parameters are usually needed even for the same task, since compression models optimized for one dataset may not work well for another.

More often than not, restored human-centric videos in real-world applications are used for both human viewing and multiple analytic tasks, such as face recognition, emotion analysis, *etc.* In such cases, previous methods not only require to maintain multiple compression models in the system, but also need to compute and transmit multiple compressed video streams. The lack of generalizability and scalability has become an open issue.

In this work, we propose a novel framework for compressing human-centric videos to better accommodate multiple tasks and multiple data domains. Our system combines the power of generic face prior learning with data-dependent detail recovery to achieve robust face restoration with very low bitrates. As illustrated in Fig. 1, our framework comprises of three branches: a generic branch, a domain-adaptive branch, and a task-adaptive branch.

The generic branch takes advantage of the recent ad-

vances in face representation learning [8, 15, 29] and reconstructs a high-quality (HQ) generic face using a discrete HQ generic codebook-based representation. Since face is highly structured, the generic codebook learned from HQ face data captures the essential components for HQ face generation. With ultra-low transmission costs (*e.g.*, only 256 10-bit integers indicating codeword indices), the reconstructed generic face can achieve decent perceptual quality.

The domain-adaptive branch improves the perceptual authenticity of the restored face by using a discrete domain-specific codebook-based representation, with only a small increase of bitrate (*e.g.*, 256 integers). The domain-specific feature is weighted combined with the generic feature for domain-adaptive reconstruction. This provides supplementary information drawn from the current data distribution to fit the current data domain. The combining weight balances the reconstruction between having a higher perceptual quality or being more authentic to the current data domain. By sharing the main generic branch while maintaining an individual domain-adaptive branch, our method can flexibly scale to multiple data domains.

The task-adaptive branch provides detailed fidelity and expressive information to help analytic tasks. A low-bitrate low-quality (LQ) face input (highly compressed by a coding method like VVC [12] or learned image compression (LIC) [6]) is transmitted to the decoder, where detailed feature is derived and weighted combined with the generic codebook feature for task-oriented reconstruction. The task-adaptive branch balances the bitrate, the reconstruction fidelity, and the task performance, by adjusting the combining weight on the encoder side, according to the compression requirement of the current task. By sharing the main generic branch while maintaining an individual task-adaptive branch, the framework can flexibly scaled to multiple tasks.

We further propose a mechanism to automatically adjust the combining weights for the domain-adaptive branch and the task-adaptive branch at test time. Through online meta learning (OML) [13] with direct stochastic gradient decent (SGD), the combining weights are automatically tuned for the current test data according to the actual reconstruction need, *e.g.*, to improve perceptual quality, perceptual authenticity, or recognition accuracy.

Our main contributions are summarized as follows:

- A novel human-centric video compression framework based on robust face restoration, to accommodate compression needs for both humans and machines. The generic branch ensures baseline HQ face reconstruction using a highly effective discrete generic codebook-based representation. The domain-adaptive branch provides supplementary information using a domain-specific codebook, to adjust reconstruction for the current data domain. The task-adaptive branch derives additional detailed cues from an LQ input to as-

sist analytic tasks over the restored face. Our method can easily scale to multiple data domains and tasks.

- Flexible quality control at test time. The HQ generic feature and domain-adaptive feature are weighted combined to balance the perceptual quality and authenticity to the current data domain. Likewise, the HQ generic feature and task-adaptive LQ feature are combined using a combining weight that balances the bitrate and task performance.
- An OML mechanism that automatically adjusts the combining weights for the domain-adaptive feature and the task-adaptive feature, according to the current test data and the actual compression need.

In experiments, we evaluate the capabilities of domain and task adaptation by training the generic branch with high-quality faces from the FFHQ dataset [14], training the domain-adaptive branch and task-adaptive branch with the mediocre-quality internet faces from the CASIA-WebFace dataset [27], and performing evaluation over the real-world LFW dataset [9]. We also evaluate the flexibility of the codebook-based representation in working with either the off-the-shelf VVC [12] or NN-based LIC [6] for compressing LQ faces in the task-adaptive branch. Experimental results demonstrate the effectiveness of our method in terms of both restoration quality and recognition accuracy.

## 2. Related Work

### 2.1. Face reenactment & AI-based video conference

Face reenactment aims at transferring the facial motion of one driving face to another source face. It is widely used for applications like animated avatar manipulation or controllable face generation, by injecting 2D [23] or 3D [7] geometric pose and expression from the driving face to the identity and appearance generation of the target face. It is intuitive to use face reenactment for video conferencing. By transferring and storing a few HQ faces carrying identity and appearance cues of the subject, the decoder can synthesize the remaining frames of that subject based on new driving keypoints. Only keypoints need to be transferred for most successive frames, which reduces bitrates dramatically. However, when applied to real faces in the wild, they usually produce severe artifacts due to the difficulty in generating real hair, teeth, accessories, or any content that cannot be represented by facial keypoints. Also, the mismatch between the appearance of the source frame and the pose/expression of the driving frame makes such methods sensitive to changes in illuminations, poses, expressions, *etc.* Some artifacts can be reduced by using multiple source faces and constraining reenactment only to tight face regions [16, 20], but the innate instability remains.

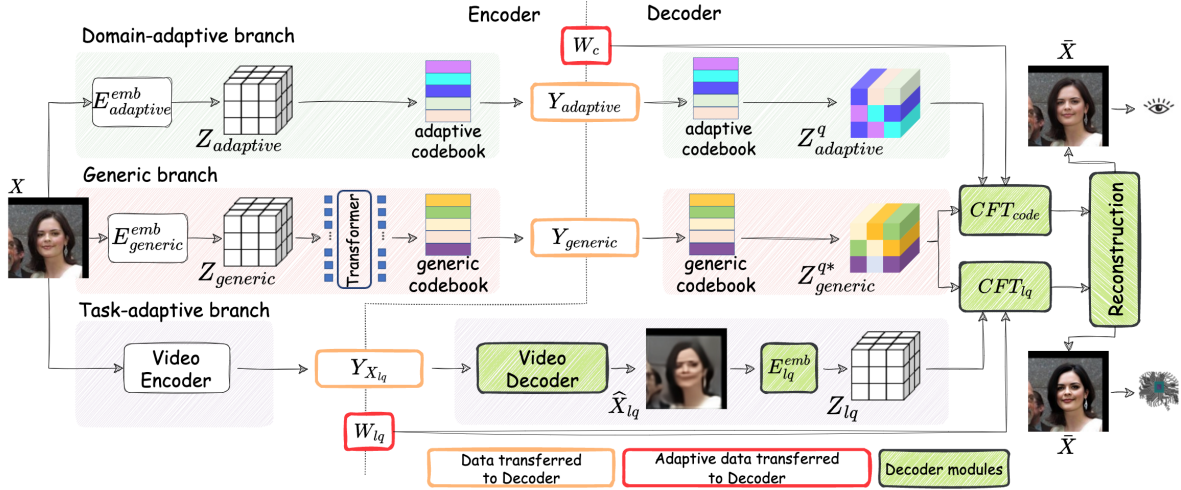


Figure 1. The overall framework of the proposed method. The generic branch reconstructs an HQ generic face. The domain-adaptive branch improves restoration authenticity. The task-adaptive branch provides fidelity and expressive details for analytic tasks.

## 2.2. Face restoration

Blind face restoration aims at recovering the HQ face from its LQ counterparts, which is degraded from the HQ version in an unknown way, such as low-resolution, noise, blur, compression, *etc.* Various face priors have been used to improve the LQ-to-HQ mapping in this ill-posed problem. Comparing to traditional geometry facial landmarks [5], the recent generative priors have shown great potential, which embed face priors into the encoder-decoder network structure and at the same time use the structural information from the input for high-quality and high-fidelity reconstruction, *e.g.*, GLEAN [4], GPEN [26], and GFPGAN [25].

On the other hand, sparse representation learning has been explored to model generic images. VQVAE [21] learns a highly compressed codebook by a vector-quantized autoencoder. VQGAN [8] further improves restoration quality by using GAN training with adversarial and perceptual loss. Such learned codebooks are optimized end-to-end (E2E), to balance efficiency and reconstruction quality. The latest CodeFormer [29] learns a discrete HQ codebook by predicting code sequences through global modeling. Faces can be restored using the HQ codebook without heavy dependency on feature fusion with LQ visual cues, which improves the generation robustness significantly.

## 2.3. Video coding for machine analysis

Traditionally videos are compressed for human viewing, and the degradation from compression can severely affect the performance of automated machine analysis, such as surveillance, diagnostics, *etc.* Standardization efforts such as MPEG VCM [17] and JPEG AI [11] have been launched to study video coding frameworks specialized for machine. Typically, a compression method is designed to optimize the compression efficiency for each task, such as object detection, segmentation, or tracking. Although significant bits

can be saved by such customization, these methods lack generalizability and scalability in reality. Optimized for one analytic model of one task, the compression method generally cannot work well for another task or even a different model of the same task. It is practically impossible to design and learn a compression method for each potential task. How to compress videos to support multiple tasks and to support both human and machine use remain open.

Fortunately, for videos focusing on human faces, the recent advances in face representation learning enable our solution to benefit from the powerful generic codebook-based restoration. Relying on a generic face prior for a baseline HQ reconstruction, domain-adaptive and task-adaptive details can be provided additionally to improve the perceptual quality and analysis accuracy for human and machine consumption, respectively. Compared to previous customized compression for each machine analysis task, our framework not only can better scale to different tasks and better generalize to different data domains, but can also provide control to balance perceptual quality, fidelity, and task performance.

## 3. Our Video Compression Framework

A general task-oriented video compression system that supports both human and machine use has an **Encoder** and a **Decoder**. Given input video frames, the **Encoder** compresses each frame to compute a latent representation, which is sent to the **Decoder** with much less bits than the original frame. Then the **Decoder** reconstructs the frame based on the latent representation. The goal is to minimize the loss of visual quality (*e.g.*, distortion like MSE, or perceptual quality loss like LPIPS [28]), minimize the bitrate, and maintain task performance (*e.g.*, recognition accuracy).

For human-centric videos where faces are the main focus, the **Encoder** starts with face detection. Face regions consisting of an extended bounding box containing mainly

the detected and aligned face are cropped and resized (*e.g.*, to  $512 \times 512$ ) as the input (denoted as  $X$ ) to the remaining processing modules. Correspondingly, the **Decoder** inverse-transforms and resizes the restored face (denoted as  $\bar{X}$ ), which is used to perform the end task and also blended back to the remaining part of the frame.

### 3.1. Generic branch

Aiming at using a highly compressed and HQ learned codebook for generating HQ faces, we use the latest CodeFormer restoration network [29] as the generic branch. Comparing with its ancestors like VQVAE [21] or VQGAN [8], CodeFormer focuses on learning an HQ codebook for restoring HQ face, and does not overly rely on feature fusion with LQ cues from the skip connections. Since faces in real-world videos generally have lower quality than benchmark HQ training faces, without heavy dependence on the LQ cues not only increases the system robustness, but also enables generating a face with high perceptual quality even when the quality of the original input is quite low.

Specifically, as shown in Fig. 1, following the recipe of VQGAN [8], the input face  $X \in \mathbb{R}^{w \times h \times 3}$  is first embedded as a generic latent feature  $Z_{generic} \in \mathbb{R}^{u \times v \times d}$  through a Generic Embedding network  $E_{generic}^{emd}$ , which is then mapped to a generic quantized feature  $Z_{generic}^q \in \mathbb{R}^{u \times v \times d}$ . Each “pixel”  $Z_{generic,l}^q$  ( $l=1, \dots, k, k=u \times v$ ) corresponds to a codeword  $c_{generic,l}$  in the learnable generic HQ codebook  $\mathcal{C}_{generic} = \{c_{generic,l} \in \mathbb{R}^d\}$  that is nearest to the corresponding latent feature  $Z_{generic,l}^q$  of the “pixel”. To provide rich and robust visual cues for HQ face generation, the HQ codebook is trained by using HQ faces as inputs and minimizing a joint loss:

$$\mathcal{L}_{generic} = \mathcal{L}_1(X, \bar{X}) + \mathcal{L}_{per}(X, \bar{X}) + \mathcal{L}_{code}(Z_{generic}, \mathcal{C}_{generic}) + \lambda_{ad} \mathcal{L}_{ad}(X, \bar{X}), \quad (1)$$

where  $\mathcal{L}_1(X, \bar{X})$ ,  $\mathcal{L}_{per}(X, \bar{X})$ , and  $\mathcal{L}_{ad}(X, \bar{X})$  are  $L_1$  loss, perceptual loss [28], and adversarial loss of a discriminator [8] respectively. The code-level loss  $\mathcal{L}_{code}(Z_{generic}, \mathcal{C}_{generic})$  [8] reduces the intermediate loss between codeword  $c_{generic,l}$  and embedded feature  $Z_{generic,l}^q$ :  $\|sg(Z_{generic,l}^q) - c_{generic,l}\|_2^2 + \alpha \|Z_{generic,l}^q - sg(c_{generic,l})\|_2^2$  ( $sg(\cdot)$  as stop-gradient), which regularizes codebook learning.

To reduce the influence of unknown quality of the input face that causes inaccurate codeword matching, a more accurate quantized feature  $Z_{generic}^{q*}$  is computed through a Transformer  $T$ . The key idea is to make use of the global interrelations of the codebook representation for better code prediction [29]. Specifically, the latent feature  $Z_{generic} \in \mathbb{R}^{u \times v \times d}$  is reshaped to a number of  $k = u \times v$  features of  $d$  dimensions. Then it is fed into the Transformer  $T$ , which predicts the code sequence  $p_1, \dots, p_k$  in the form of the probability of the  $M$ -way classification,

where  $p_l \in \{0, \dots, M-1\}$  and  $M$  is the size of the codebook. The predicted code sequence then retrieves the  $k$  respective code items from the learned codebook, forming the quantized feature  $Z_{generic}^{q*} \in \mathbb{R}^{u \times v \times d}$  to compute an HQ face through the same reconstruction network. In [29], the Transformer  $T$  is trained by fixing the above learned HQ codebook and reconstruction network, and then using the LQ face as input to minimize the code prediction loss:

$$\mathcal{L}_{pred} = \lambda_{token} \mathcal{L}_{token} + \mathcal{L}_{code}^{feat}, \quad (2)$$

where  $\mathcal{L}_{token} = \sum_l -p_l^{hq} \log(p_l)$  and  $p_l^{hq}$  is the ground-truth code sequence obtained with HQ input.  $\mathcal{L}_{code}^{feat} = \sum_l \|Z_{generic,l} - sg(c_{generic,l}^{hq})\|_2^2$  forces embedded feature  $Z_{generic,l}$  using LQ input to approach the HQ codewords  $c_{generic,l}^{hq}$  that are retrieved using  $p_l^{hq}$ .

Since  $Z_{generic}^{q*}$  can be represented by a  $k$ -dim vector  $Y_{generic}$  consisting of codeword indices  $p_1, \dots, p_k$ , it can be efficiently transmitted to the **Decoder** with very little bit consumption. The **Decoder** first recovers the generic quantized feature  $Z_{generic}^{q*}$  based on the received vector  $Y_{generic}$  using codebook  $\mathcal{C}_{generic}$ , and then reconstructs the HQ restored face  $\bar{X}$  based on  $Z_{generic}^{q*}$ .

### 3.2. Domain-adaptive branch

When applied to a specific domain of data, *e.g.*, videos from hand-held devices with motion blur, the restored HQ face from the generic branch may not be authentic to the true input, due to the large domain change. The domain-adaptive branch bridges such domain difference by using a domain-specific learned codebook to provide supplementary information drawn from the current data domain. Through a discrete domain-specific codebook-based representation, the reconstruction is tailored to fit the current data domain while maintaining a low bitrate. The framework is more flexible to scale to multiple data domains by sharing the generic branch, contrary to the traditional way of training a whole set of compression model for each data domain.

Similar to the generic branch, the input face  $X \in \mathbb{R}^{w \times h \times 3}$  is embedded as a domain-adaptive latent feature  $Z_{adaptive} \in \mathbb{R}^{u \times v \times d}$  through a Domain-Adaptive Embedding network  $E_{adaptive}^{emd}$ , which is then mapped to a domain-adaptive quantized feature  $Z_{adaptive}^q \in \mathbb{R}^{u \times v \times d}$ . Each “pixel”  $Z_{adaptive,l}^q$  ( $l=1, \dots, k, k=u \times v$ ) corresponds to a codeword  $c_{adaptive,l}$  in the learnable domain-adaptive codebook  $\mathcal{C}_{adaptive} = \{c_{adaptive,l} \in \mathbb{R}^d\}$  that is nearest to the corresponding latent feature  $Z_{adaptive,l}^q$  of the “pixel”. Since  $Z_{adaptive}^q$  can be represented by a  $k$ -dim vector  $Y_{adaptive}$  consisting of codeword indices similar to the generic branch, it can also be efficiently transmitted to the **Decoder** with very little bit consumption.

In **Decoder**, the domain-adaptive quantized feature  $Z_{adaptive}^q$  can be retrieved based on the received vector



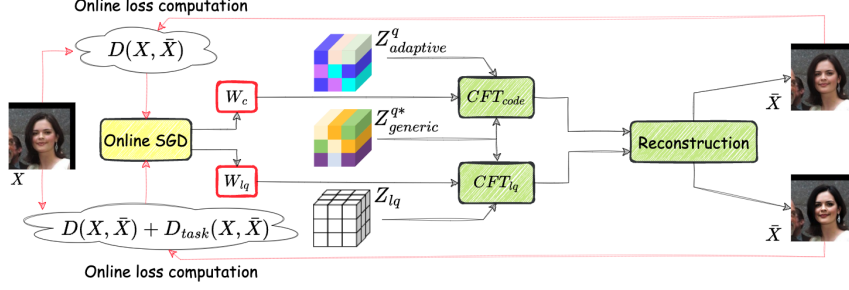


Figure 2. Framework of online adaptive learning. Combining weights for the domain-adaptive feature and task-adaptive feature are automatically adjusted through online SGD, according to the actual compression need that is defined by the online loss.

$Y_{adaptive}$  using codebook  $\mathcal{C}_{adaptive}$ . Then the generic quantized feature  $Z^{q*}_{generic}$  and the domain-adaptive quantized feature  $Z^q_{adaptive}$  are weighted combined to reconstruct the restored face  $\bar{X}$ . Specifically, the Controllable Feature Transformation (CFT) module from [29] is used. Let  $\Theta_c$  denote the parameters of the CFT module  $CFT_{code}$  to combine codebook-based features  $Z^q_{adaptive}$  and  $Z^{q*}_{generic}$ . Actually  $Z^q_{adaptive}$  tunes the generic feature  $Z^{q*}_{generic}$  into  $Z^{code} = Z^{q*}_{generic} + w_c * (\beta_c * Z^{q*}_{generic} + \gamma_c)$ , where  $\beta_c, \gamma_c$  are affine parameters  $\beta_c, \gamma_c = \Theta_c(\text{con}(Z^{q*}_{generic}, Z^q_{adaptive}))$ , and  $\text{con}(\cdot)$  is the concatenation operation.

To learn the domain-adaptive codebook  $\mathcal{C}_{adaptive}$  and module  $CFT_{code}$ , the training faces from the current data domain are used to minimize a joint loss similar to Eqn. (1):

$$\mathcal{L}_{adaptive} = \mathcal{L}_1(X, \bar{X}) + \mathcal{L}_{per}(X, \bar{X}) + \mathcal{L}_{code}(Z_{adaptive}, \mathcal{C}_{adaptive}) + \lambda_{ad}\mathcal{L}_{ad}(X, \bar{X}).$$

$w_c = 1$  during the training stage for all inputs.

### 3.3. Task-adaptive branch

The restored faces from codebook-based representations are perceptually pleasant for human eyes, but can be suboptimal for machine analysis. The facial details may be altered and the identity information may be lost, which may cause severe performance drop for tasks relying on such information and details, *e.g.*, face recognition, emotion analysis *etc.*

The task-adaptive branch provides supplementary identity and expressiveness details to the restored face to assist analytic tasks. A highly-compressed string  $Y_{X_{lq}}$  is computed from the input  $X$  in **Encoder** and is transmitted to the **Decoder** with very low bitrate. Then the **Decoder** decodes an LQ input  $\hat{X}_{lq}$  and computes an LQ embedded feature  $Z_{lq}$  from  $\hat{X}_{lq}$  using the LQ embedding network  $E_{lq}^{emd}$ . The LQ feature  $Z_{lq}$  and the generic quantized feature  $Z^{q*}_{generic}$  are weighted combined to reconstruct the task-oriented face  $\bar{X}$ . Similar to the domain-adaptive branch, a CFT module  $CFT_{lq}$  is used for this combination. Let  $\Theta_{lq}$  denote the parameters of  $CFT_{lq}$ .  $Z_{lq}$  tunes the generic  $Z^{q*}_{generic}$  into  $Z^{task} = Z^{q*}_{generic} + w_{lq} * (\beta_{lq} * Z^{q*}_{generic} + \gamma_{lq})$ .  $\beta_{lq}, \gamma_{lq}$  are affine parameters  $\beta_{lq}, \gamma_{lq} = \Theta_{lq}(\text{con}(Z^{q*}_{generic}, Z_{lq}))$ . The framework is more flexible to scale to multiple tasks by sharing

the generic branch, in comparison to training a whole customized compression system for each task.

It is worth mentioning that the video encoder/decoder in the task-adaptive branch for computing  $Y_{X_{lq}}$  and recovering  $\hat{X}_{lq}$  can be arbitrary video compression methods, including both NN-based LIC methods like [6] or off-the-shelf tools like VVC [12]. We evaluate both choices in Section 4.

The CFT module  $CFT_{lq}$  is trained with finetuning the embedding network  $E_{generic}^{emd}$  into the LQ embedding network  $E_{lq}^{emd}$  at the same time, using the training data of the current analytic task. The training loss includes the L1 loss  $L_1$ , the perceptual loss  $\mathcal{L}_{per}$ , the code-level loss  $\mathcal{L}_{code}$ , and the adversarial loss  $\mathcal{L}_{ad}$  from Eqn. (1), and the code feature loss  $\mathcal{L}_{code}^{feat}$  from Eqn. (2). In addition, the task loss  $\mathcal{L}_{task}$  is also used, *e.g.*, the triplet loss of the embedded face feature through FaceNet [22] for face recognition in our experiments.  $w_{lq} = 1$  during the training stage for all inputs.

### 3.4. Online adaptation

Our compression framework provides flexibility to control the restoration according to the current input  $X$ . For human viewing, the domain-adaptive branch is used with weight  $w_c$ , to balance the perceptual quality and authenticity to the current data domain. For machine analysis, the task-adaptive branch is used with weight  $w_{lq}$ , to balance the bitrate and task performance.

At test time, given an online compression goal, *i.e.*, a loss function  $\mathcal{L}_{online}(X, \bar{X})$ , its reconstruction task can be seen as drawn from a task distribution and weights  $w_c$  and  $w_{lq}$  are meta variables of the distribution. As described in Fig. 2, the OML method [13] can be used to adaptively tune  $w_c$  and  $w_{lq}$  through direct SGD by back-propagating the gradients of  $\mathcal{L}_{online}(X, \bar{X})$ . Instead of online tuning NN parameters, tuning the weights is much more stable. Also, only the tuned weight  $w_c$  or  $w_{lq}$  needs to be additionally sent to **Decoder** with almost no transmission overhead.

The online loss can be flexible. For human viewing,  $\mathcal{L}_{online}$  is distortion  $D(X, \bar{X})$ , *e.g.*, combination of MSE and LPIPS. For machine analysis,  $\mathcal{L}_{online} = D(X, \bar{X}) + \lambda_{task}\mathcal{D}_{task}(X, \bar{X})$ , which includes task-oriented distortion  $\mathcal{D}_{task}(X, \bar{X})$ . For example,  $\mathcal{D}_{task}(X, \bar{X})$  can be MSE be-

tween output features from a face embedding network like FaceNet [22], using  $X$  or  $\bar{X}$  as input.

During online learning, the generic quantized feature  $Z_{generic}^{q*}$ , the domain-adaptive quantized feature  $Z_{adaptive}^q$  and the LQ embedded feature  $Z_{lq}$  are kept unchanged. That is, the transmission bitrate remains unchanged since the generic  $Y_{generic}$ , the adaptive  $Y_{adaptive}$  and the LQ  $Y_{X_{lq}}$  remain the same. The inference time in **Decoder** remains unchanged. We only need to perform multiple inference iterations in **Encoder** over the  $CFT_{code}/CFT_{lq}$  and reconstruction module, and only 5 SGD iterations are taken in total. The additional time complexity in **Encoder** is small.

## 4. Experiments

To evaluate domain and task adaptation performance, the generic branch uses the pre-trained model from [29], which is trained with the FFHQ dataset [14] consisting of 70,000 HQ Flickr images at  $1024 \times 1024$  resolution; the domain-adaptive branch is trained with the CASIA-WebFace dataset [27], consisting of 455,594 images from 10,575 identities crawled from webpages with mediocre quality; and evaluation is over the real-world LFW dataset [9], which consists of 13,233 images from 5,749 identities and is one of the most challenging datasets for face recognition.

We evaluate PSNR, SSIM, LPIPS for human use and recognition accuracy for machine analysis. Several methods are tested, *i.e.*, “code only” using only codebook-based features targeting at human use, where the generic  $Z_{generic}^{q*}$  and domain-adaptive  $Z_{adaptive}^q$  are combined for reconstruction, and the task-adaptive methods “t-adapt” and “t-adapt oml” where the generic  $Z_{generic}^{q*}$  and task-adaptive  $Z_{lq}$  are used, with or without online learning, respectively.

We also test the flexibility of using different encoder/decoder to compress the LQ face in the task-adaptive branch, including off-the-shelf VVC [12] and NN-based LIC [6]. The bitrate of the encoder/decoder determines the quality of the LQ face, which further influences reconstruction. So we test different bitrate configurations too. For each image, only the largest face in the center are considered. Each detected face is extended and cropped to include twice the area of the detection box using the facelib library, and then resized to  $512 \times 512$ . For LIC, we use pre-trained models from the CompressAI library [1]. For face recognition, the FaceNet from PyTorch model zoo embeds the restored face into a feature vector for similarity matching, and the reported accuracy is over 10-fold cross-validation.

To present the final result, we use our algorithm to compress the extended face area, which is then merged back to the VVC/LIC compressed image. For human viewing, the codebook-based solution (“code only”) reconstructs face using  $Z_{generic}^{q*}$  and  $Z_{adaptive}^q$  from the generic and domain-adaptive branches. The bit costs include the sparse  $Y_{generic}$  and  $Y_{adaptive}$  for the extended face area and the bit counts

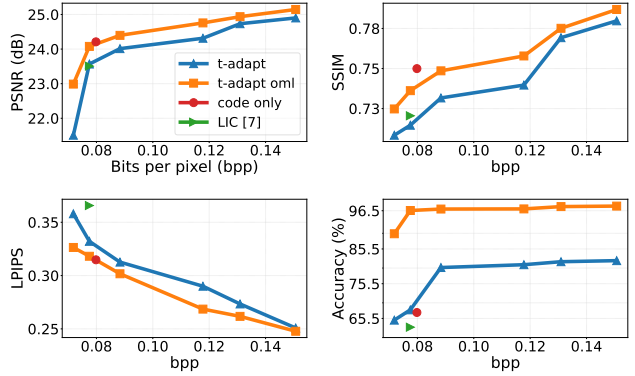


Figure 3. Performance with LIC [6] as video encoder/decoder for task-adaptive branch.  $q = 2$  gives a balanced trade-off among bitrate, visual quality, and recognition accuracy.

for the remaining pixels. For machine analysis the task-adaptive solution (“t-adapt”) uses the generic  $Z_{generic}^{q*}$  and task-adaptive  $Z_{lq}$  for face reconstruction. The bit costs include  $Y_{generic}$  and the entire VVC/LIC encoded image. An image-based approach is taken following the setup in standard evaluation like VCM [17]. That is, the all-intra mode is used for VVC to compress images with intra-prediction.

### 4.1. Performance pairing with LIC

Using the NN-based LIC [6] as the video encoder/decoder in the adaptive branch, Fig. 3 shows the performance comparison with 6 pre-trained models from CompressAI corresponding to 6 target bitrates. To get ultra-low bitrates, the original  $X$  is downsampled  $4\times$  (*i.e.*  $16\times$  resolution reduction) before compressed by LIC, and the decoded image from LIC is upsampled  $4\times$  to the original size. The bicubic filter is used for resizing.

From the results, without online learning, using only codebook-based features, “code only” gives good overall visual quality for PSNR, LPIPS and SSIM, but bad accuracy. Here the bitrate is computed by using the codebook-based representation for the face area while using LIC compression for remaining pixels. With the help from the LQ face input, the “t-adapt” method restores faces that are better for recognition, but with reduced visual quality. The better the compression quality of the LQ face, the better the reconstruction quality, and the larger the bit consumption. With online learning, using adaptive combining weights  $w_{lq}$  the “t-adapt oml” method largely improves the recognition accuracy, with only little drop of the visual quality. The compression model of  $q = 2$  gives a balanced performance overall, where the 0.078 bpp is fairly evenly distributed to generic  $Y_{generic}$  (0.041 bpp) and  $Y_{X_{lq}}$  (0.037 bpp) and the recognition accuracy reaches 0.966 with reasonable perceptual quality. After  $q = 2$ , the accuracy saturates. This shows that reasonably good details have been captured by the low-bitrate LQ face to perform recognition.

Table 1. Performance for human viewing using VVC as video encoder/decoder for task-adaptive branch. The domain-adaptive “code only” gives the best visual quality for both configurations.

QP=30	psnr	ssim	lpips	bpp
vvc	26.35	0.807	0.341	0.0999
code only	<b>27.52</b>	<b>0.824</b>	<b>0.231</b>	0.0995
QP=42	psnr	ssim	lpips	bpp
vvc	23.93	0.704	0.406	0.0320
code only	<b>24.85</b>	<b>0.743</b>	<b>0.329</b>	0.0450

## 4.2. Performance paring with VVC

We test two bitrate configurations for VVC as the video encoder/decoder in the task-adaptive branch:  $QP = 30$  and  $QP = 42$ . Same as Section 4.1, the original  $X$  goes through  $4\times$  downsampling before VVC encoding and  $4\times$  upsampling after VVC decoding to get ultra-low bitrates. The bicubic filter is used for resizing. Table 1 and Table 2 give the performance for human viewing and machine analysis, respectively. Fig. 4 gives some reconstruction examples.

### 4.2.1 For human viewing

From Table 1, the “code only” method for human viewing largely outperforms VVC in both configurations with 44%/38% PSNR improvement and 48%/23% LPIPS improvement for  $QP=30/42$ . From Fig 4, with similar overall bitrates using  $QP=30$  for “code only” (0.0995 bpp) and VVC (0.0999 bpp), our reconstructed faces are clearly more visually pleasing than the VVC compressed version. When bitrate is decreased from  $QP=30$  to  $QP=42$ , the visual quality of VVC drops significantly. As for “code only”, quality of faces reconstructed by codebook is the same, and the performance drop comes from the remaining pixels.

### 4.2.2 For machine analytics

For face recognition, as shown in Table 2, by using the generic  $Z_{generic}^{q*}$  and task-adaptive  $Z_{lq}$ , our “t-adapt oml” method largely outperforms VVC for both configurations, *i.e.*, 37%/28% PSNR, 44%/22% LPIPS, and 17%/41% accuracy improvements for  $QP=30/42$ . Also, our reconstructed faces look much more pleasant than the VVC compressed ones as shown in Fig.4. Often our results look even better than the original ground-truth, thanks to the powerful HQ face prior learned by the HQ generic codebook, exceeding the quality of the actual input.

Without online adaptation, the “t-adapt” method gives reasonably good reconstruction using  $QP=30$  with 0.957 accuracy, and “t-adapt oml” further boosts the performance by online learning. For  $QP=42$  where the decoded LQ face has very bad quality, online learning plays an important role by per-datum adaptation, which improves both visual quality and recognition accuracy significantly.

### 4.2.3 More discussions

We can flexibly configure our method according to different compression needs. Considering Table 1, Table 2,

Table 2. Performance for machine analytics using VVC as video encoder/decoder for task-adaptive branch. The “t-adapt” method outperforms VVC, and “t-adapt oml” further boosts the performance, especially at low bitrates.

QP=30	psnr	ssim	lpips	accuracy	bpp
vvc	26.35	0.807	0.341	0.826	0.100
t-adapt	27.11	0.811	0.239	0.957	0.140
t-adapt oml	<b>27.33</b>	<b>0.815</b>	<b>0.236</b>	<b>0.976</b>	0.140
QP=42	psnr	ssim	lpips	accuracy	bpp
vvc	23.93	0.704	0.406	0.680	0.032
t-adapt	23.91	0.695	0.360	0.857	0.073
t-adapt oml	<b>24.61</b>	<b>0.726</b>	<b>0.334</b>	<b>0.962</b>	0.073

and Fig. 4 together, for human viewing we recommend “code only” using  $Z_{generic}^{q*}$  and  $Z_{adaptive}^q$  with  $QP=30$ . For machine analytics, we recommend “t-adapt oml” using  $Z_{generic}^{q*}$  and  $Z_{lq}$  with  $QP=42$ . In such cases, we can achieve good perceptual quality or accuracy with 0.1 bpp. When the reconstruction is used for both human and machine, we recommend “t-adapt oml” with  $QP=30$ , where results are both visually pleasing and good for recognition, still having a reasonably low bitrate of 0.14 bpp.

From another perspective, the task-adaptive solution can be seen as an augmentation to off-the-shelf codecs like VVC. That is, when using VVC at very low bitrates, by transmitting an additional discrete codebook-based representation  $Y_{generic}$  (2560 bits/frame), we can improve the task accuracy from 0.680 to 0.962 for  $QP=42$  and from 0.826 to 0.976 for  $QP=30$ . The perceptual quality is also largely improved in general.

We notice that it is hard to balance task loss and visual quality in training the task-adaptive branch. It is probably because of the limited regularization power of the discriminative task loss, similar to the common problem in GAN training. Online learning can mitigate this issue and largely improve the performance with task-oriented online loss. Also, online tuning of  $w_c$  only brings marginal gain over the preset  $w_c = 1$ . This can be caused by the same distortion loss used for both online and offline training of the domain-adaptive branch. The network is already optimized for that loss, leaving little room for online improvement.

### 4.2.4 Limitations

Our approach uses the structural characteristics of human faces to achieve high compression efficiency. It can potentially be used on other types of content with highly structured features (*e.g.*, human body) where a generic sparse neural representation can be learned. It may not be effective when being applied to general video content.

Similar to the VCM evaluation, our method compresses each frame individually. This is due to the lack of large-scale labeled video sets and video-oriented models trained for recognition tasks. As a compression method, we should



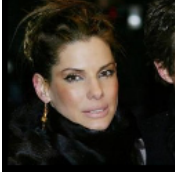
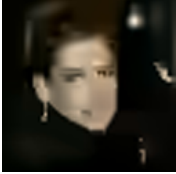

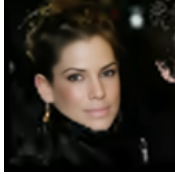
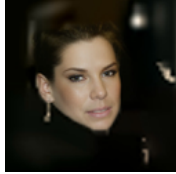


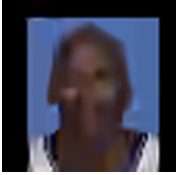
















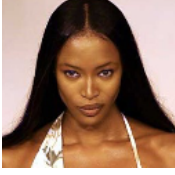
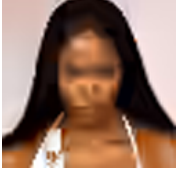
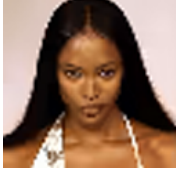
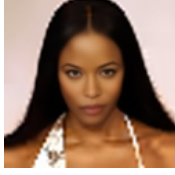
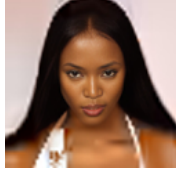
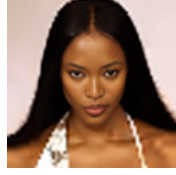
Ground truth	VVC (QP=42)	VVC (QP=30)	code only	t-adapt	t-adapt om1
					
	0.413 25.17 0.612 0.023	0.300 28.25 0.754 0.077	0.257  <b>28.80</b>   <b>0.773</b>  0.081	0.304 25.95 0.663 0.064	<b>0.255</b>  28.35 0.761 0.118
					
	0.364 24.27 0.712 0.024	0.261 26.02  <b>0.817</b>  0.068	0.208  <b>26.09</b>  0.816 0.071	0.245 27.07 0.774 0.065	<b>0.205</b>  26.00 0.814 0.109
					
	0.309 21.48 0.774 0.031	0.194 22.96 0.862 0.094	<b>0.131</b>   <b>23.10</b>   <b>0.867</b>  0.092	0.204 22.16 0.821 0.075	0.168  <b>23.10</b>  0.865 0.135
					
	0.388 22.90 0.662 0.036	0.228 26.08 0.830 0.108	0.198  <b>26.34</b>   <b>0.833</b>  0.110	0.312 23.42 0.683 0.077	<b>0.192</b>  26.10 0.826 0.149
					
	0.355 23.54 0.711 0.031	0.244 25.43 0.811 0.093	0.199 25.83 0.827 0.094	0.233 24.66 0.765 0.072	<b>0.187</b>   <b>26.13</b>   <b>0.834</b>  0.134

Figure 4. Examples using VVC as video encoder/decoder in task-adaptive branch. Numbers under each reconstruction result are “LIPIS|PSNR|SSIM|bpp”. In general, the domain-adaptive “code only” performs well objectively, and the task adaptive “t-adapt om1” gives the best recognition accuracy with good perceptual quality. For both VVC configurations our approaches subjectively outperform the VVC compressed counterparts significantly.

not restrict and retrain machine task models, and the image-based method is therefore used to be plugged into the existing machine task models. One naive way to extend to video is to compress I-frames only. However, as shown in experiments the reconstructed I-frames then may have higher quality than the original inputs, and the traditional inter-prediction may not be efficient anymore. One future work is to investigate effective temporal prediction methods to improve video compression efficiency.

## 5. Conclusions

We proposed a robust framework for human-centric video compression to accommodate both human viewing and machine analytics. The generic branch used the

highly efficient generic codebook-based representation to ensure face reconstruction with high perceptual quality. The domain-adaptive and task-adaptive details were provided in addition to improve, respectively, the visual authenticity to the current data domain for human use and the task performance for machine analysis. The combining weights of the generic, domain-adaptive and task-adaptive features were online adjusted to fit different compression needs. Experiments demonstrated superior perceptual quality and task accuracy with very low bitrate. Comparing to conventional coding methods, our framework can be flexibly configured and can better scale to multiple data domains and tasks.



## References

- [1] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja. CompressAI: a PyTorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020. 6
- [2] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *Proc. IEEE CVPR*, pages 6228–6237, 2018. 1
- [3] Y. Blau and T. Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. In *Proc. ICML*, pages 675–685. PMLR, 2019. 1
- [4] K. Chan, X. Wang, X. Xu, J. Gu, and C. Loy. GLEAN: Generative latent bank for large-factor image super-resolution. In *Proc. IEEE CVPR*, 2021. 3
- [5] Y. Chen, Y. Tai, X. Liu, C. Shen, and J. Yang. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proc. IEEE CVPR*, 2018. 3
- [6] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proc. IEEE CVPR*, 2020. 1, 2, 5, 6
- [7] Y. Deng, J. Yang, D. Chen, F. Fang Wen, and X. Tong. Disentangled and controllable face image generation via 3d imitative-contrastive learning. In *Proc. IEEE CVPR*, 2020. 2
- [8] P. Esser, R. Rombach, and B. Ommer. Taming transformers for high-resolution image synthesis. In *Proc. IEEE CVPR*, 2021. 2, 3, 4
- [9] G. Huang, M. Mattar, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In *Workshop on faces in 'RealLife' Images: detection, alignment, and recognition*, 2008. 2, 6
- [10] 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. 1
- [11] ISO/IEC JTC 1/SC29/WG1. Report on the jpeg ai call for proposals results. In *ISO/IEC JTC1/SC29 WG1, N100250*, July 2022. 1, 3
- [12] ITU-T and ISO. Versatile video coding. Rec. ITU-T H.266 and ISO/IEC 23090-3, 2020. 1, 2, 5, 6
- [13] W. Jiang, W. Wang, S. Li, and S. Liu. Online meta adaptation for variable-rate learned image compression. In *NTIRE CVPRWs*, pages 498–506, June 2022. 2, 5
- [14] T. Karras, S. Laine, and T. Aila. A style-based generator architecture for generative adversarial networks. In *Proc. IEEE CVPR*, 2019. 2, 6
- [15] X. Li, C. Chen, S. Zhou, X. Lin, W. Zuo, and L. Zhang. Blind face restoration via deep multi-scale component dictionaries. In *Proc. ECCV*, 2020. 2
- [16] X. Li, W. Li, D. Ren, H. Zhang, M. Wang, and W. Zuo. Enhanced blind face restoration with multi-exemplar images and adaptive spatial feature fusion. In *Proc. IEEE CVPR*, 2020. 2
- [17] S. Liu, H. Zhang, and C. Rosewarne. Common test conditions for video coding for machines. In *ISO/IEC JTC1/SC29 WG4, wg04n311*, Jan. 2023. 1, 3, 6
- [18] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao. DVC: An end-to-end deep video compression framework. In *Proc. IEEE CVPR*, pages 11006–11015, 2019. 1
- [19] D. Minnen, J. Ballé, and G. Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *NeurIPS*, pages 10794–10803, 2018. 1
- [20] Y. Nirkin, Y. Keller, and T. Hassner. FSGAN: Subject agnostic face swapping and reenactment. In *Proc. IEEE ICCV*, 2019. 2
- [21] A. Oord, O. Vinyals, and K. Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017. 3, 4
- [22] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proc. IEEE CVPR*, 2015. 5, 6
- [23] T. Wang, M. Liu, A. Tao, G. Liu, A. Tao, G. Liu, J. Kautz, and B. Catanzaro. Few-shot video-to-video synthesis. In *NeurIPS*, 2019. 1, 2
- [24] T. Wang, A. Mallya, and M. Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proc. IEEE CVPR*, 2021. 1
- [25] X. Wang, Y. Li, H. Zhang, and Y. Shan. Towards real-world blind face restoration with generative facial prior. In *Proc. IEEE CVPR*, 2021. 3
- [26] T. Yang, P. Ren, X. Xie, and L. Zhang. GAN prior embedded network for blind face restoration in the wild. In *Proc. IEEE CVPR*, 2021. 3
- [27] D. Yi, Z. Lei, S. Liao, and S. Li. Learning face representation from scratch. In *arXiv preprint arXiv:1411.7923*, 2014. 2, 6
- [28] R. Zhang, P. Isola, A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE CVPR*, 2018. 3, 4
- [29] S. Zhou, K. Chan, C. Li, and C. Loy. Towards robust blind face restoration with codebook lookup transformer. In *NeurIPS*, 2022. 2, 3, 4, 5, 6