A Compressive Prior Guided Mask Predictive Coding Approach for Video Analysis*

Zhimeng Huang[1][0000−0001−8026−9349], Chuanmin Jia[2][0000−0002−7418−6245], Shanshe Wang[1][0000−0002−7665−7434], and Siwei Ma[1][0000−0002−2731−5403]

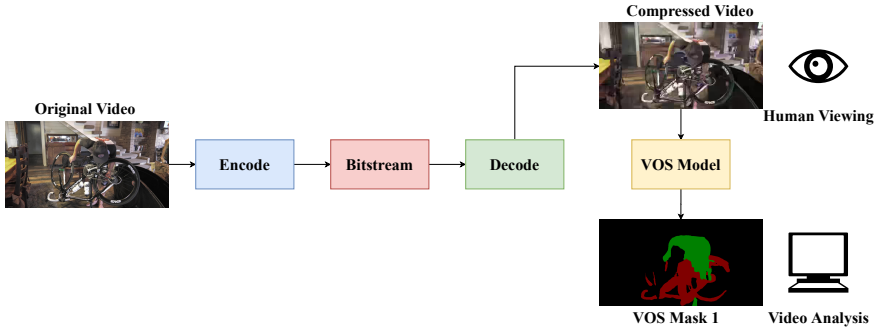
[1] National Engineering Research Center of Visual Technology, Peking University, Beijing 100871, China
[2] Wangxuan Institute of Computer Technology, Peking University, Beijing 100871, China

Abstract. In real-world scenarios, video analysis algorithms are conducted for visual signals after compression and transmission. Generally speaking, most codecs introduce irreversible distortion due to coarse quantization during compression. The distortion may lead to significant perception degradation in terms of video analysis performance. To tackle this problem, we propose an efficient plug-and-play approach to preserve the essential semantic information in video sequences explicitly. The proposed approach could boost the video analysis performance with a little extra bit cost. Specifically, we employ the proposed approach on an emerging video analysis task, video object segmentation(VOS). Massive experimental results prove that the our work outperforms the existing coding approaches over multiple VOS datasets. Concretely, it could improve the analysis performance by up to 13% at similar bitrates. Additional experiments also verifies the flexibility of our scheme because there is no dependency on any specific VOS model or encoding method. Essentially, the proposed approach provides novel insights for the emerging Video Coding for Machine (VCM) standard.

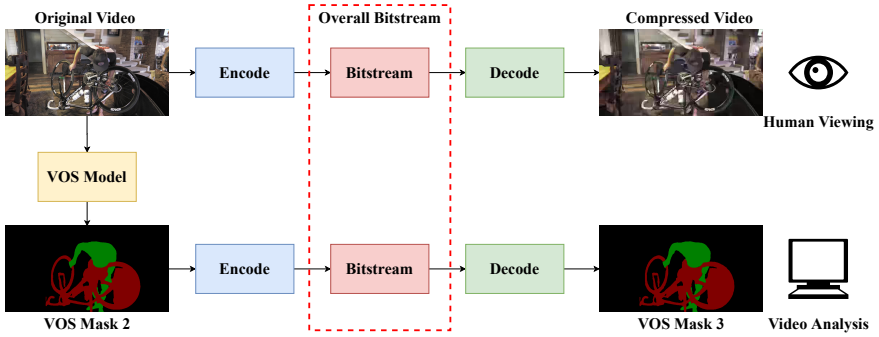
1 Introduction

In recent years, videos has become the dominant component of the internet traffic. Considering the data volume of video big data, it is necessary to develop high efficient video compression from analysis-friendly perspective. However, in earlier studies [1], the target of video compression is to simply keep the signal fidelity. Methods following this target tend to ignore other useful information in compressed domain, which may cause severe performance degradation in video analysis tasks. To discuss this problem, we regard video object segmentation,

* Supported in part by the National Natural Science Foundation of China under grant 62072008, 62025101, 61931014, 62101007, and in part by the High Performance Computing Platform of Peking University, which are gratefully acknowledged. *(Corresponding to Chuanmin Jia.)*



(a) Typical VOS Paradigm in Existing Solutions



(b) VOS Paradigm of the Proposed Approach

Fig. 1: The comparison between the paradigm of vision tasks in real-world application and that in our proposed approach. Our work uses an additional bitstream to significantly improve the VOS performance.

a popular video analysis task recently, as the representation of video analysis tasks. Semi-supervised VOS targets segmenting particular objects throughout the entire video sequence, given only the object mask of the first frame. Existing state-of-the-art (SOTA) solutions [2] [3] [4] could achieve high accuracy by fully utilizing the semantic information of the input videos. Nevertheless, the performance of these high efficient methods sharply decrease when dealing with video sequences with compression distortions, especially at low bitrate. Therefore it is critical to boost the performance of the existing video analysis methods on compressed videos.

For better comprehension, we summarize two kinds of paradigms for VOS. As shown in Fig. 1a, the VOS algorithm are conducted after signal capture, signal encoding, bitstream transmitting, and bitstream decoding. In other words, VOS is more like a downstream task for video compression. To avoid analytical performance degradation caused by video compression, we propose a compressive prior guided mask predictive coding approach. The proposed scheme is shown in 1b. Different from the traditional one, the proposed work could improve the analysis performance by an additional bitstream. Following the proposed ap-

proach, the VOS methods could be deployed on the raw videos[3] directly rather than compressed video sequences with severe semantic distortion.

Specifically, our framework could be divided into three parts, Motion Estimation (ME), Feature Compression (FC), and Motion Comprehension (MC). For the first part, we utilize the existing masks as a reference to extract a motion feature of the current mask. Then the motion feature is compressed by an end-to-end autoencoder for transmission. Finally, the compressed motion feature and the reference masks are fed into a convolution neural network (CNN) based comprehension network to generate the predicted masks. Experiments are deployed on two VOS baseline models and three standard VOS datasets. The consistently superior performances to the baseline codec demonstrate the effectiveness and generality of the proposed framework. The contributions are summarized as follows:

- We propose a novel approach to improve VOS performance on compressed video sequences. To the best of our knowledge, this is the first work who jointly considers the bitrate and the corresponding VOS performance.
- The proposed framework is high-efficient, generalizable, and flexible. It could be transferred to any VOS methods or datasets without any fine-tuning.
- Experimental results prove that the proposed framework outperforms traditional codecs over two VOS models and three VOS datasets. Exhaustive experiments also demonstrate the robustness of our framework. Our method provides novel possibilities the Video Coding for Machine (VCM) research.

2 Related Works

2.1 Video Object Segmentation (VOS)

VOS has two sub-tasks, semi-supervised VOS and unsupervised VOS. The difference between them is whether an initial mask is provided. In this paper, we mainly consider the former one. Current semi-supervised VOS methods fall into one of two categories. Following the first category, VOS method [5–7] generates the masks by fine-tuning the provided initial mask. Regarding the second category, these approaches [8–10] adopt propagating the mask from the previous frame using optical flow and then refining these estimates using a fully convolutional network. However, both VOS methods are facing difficulties when dealing with compressed video sequences due to the discriminative feature damage caused by quantization error.

Moreover, some VOS approaches utilize compressed videos as an extra supplementary information for better accuracy or efficiency. To increase VOS performance, several algorithms [11, 12] convert the input video sequences into the compressed domain. In [13], the sparse motion vector are utilized for object

[3] Note that we omit the encoding distortions caused by signal capturing tools such as cameras. Namely, we assume that the video sequences pre-processed in the dataset are all pristine videos.

segmentation. To realize high efficient processing, many researches [14] use bitstream of compressed video sequences to accelerate existing VOS methods. A plug-and-play acceleration framework is proposed in [15] by propagating the motion vectors extracted from the HEVC [16] bitstream. However, existing approaches for accuracy and efficiency pay little attention to the critical problem, the bitrate analysis of the compressed videos and their associated VOS performance. For example, if the bandwidth for compressed videos is unlimited, the common codecs allow almost lossless compression at an extremely high bitrate, which is unrealistic in practical scenarios. Therefore in this paper, we regard bitrate as another dimension of discussions, which is different from works mentioned in this subsection.

2.2 Image/Video Compression for Vision Tasks

Most of the existing compression frameworks for visual tasks aim at image tasks such as image classification, object detection, and semantic segmentation. Therefore we will introduce some image compression methods which could improve analysis performance for vision tasks. Benefited from emerging neural image/video compression approaches [17–19], most existing machine vision oriented methods mainly utilize a task-related learning objective to optimize the entire framework [20–23]. Chamain *et al.* [24] formulate a detection loss to optimize existing end-to-end image compression framework. Moreover, a content-adaptive end-to-end compression approach [25] is proposed for instance segmentation task. However, these methods rely heavily on analytical models and datasets. Therefore, they are less effective in practical application scenarios. To overcome this problem, we aim to design a framework that is effective for different datasets and analytical algorithms.

3 Methods

The proposed method is elaborated in this section. We first demonstrate the schematic illustration of our method with terminology definition. Subsequently, the detailed descriptions of each module are provided and analyzed. The notations and preliminary concepts are shown in Table 1.

3.1 Overview

The overall flowchart of the proposed approach in illustrated in Fig. 2. Denote $\mathcal{X}$ as the original video sequence with length T. For each frame $X_t \in \mathcal{X}$ at time step t, we generate the corresponding VOS mask $M_t = F(X_t)$ by VOS model F. After the generation of all of the masks, we get the mask sequence $\mathcal{M} = \{M_1, M_2, ...M_t, ...\}$. Note that the first mask M_1 is provided by the self-supervised VOS task. For other masks M_t at time step t ($t \in [2, T]$), we deconstruct it into N_t binary masks, in which N_t indicates the number of objects

Table 1: Notations and descriptions of the proposed scheme

Notations	Descriptions
$\mathcal{X} = \{X_1, X_2, \dots, X_t, \dots\}$	a sequence of video frames with timestep t
$\mathcal{M} = \{M_1, M_2, \dots, M_t, \dots\}$	a sequence of masks
M_t	original mask at t
N_t	number of objects at t
$m_{t,k}$	binary mask of object k at t
$v_{t,k}$	motion feature of object k at t
$\hat{v}_{t,k}$	compressed motion feature of object k at t
$\hat{m}_{t,k}$	the prediction of $m_{t,k}$
P_t	the prediction of M_t
$z_{t,k}$	bitstream of $v_{t,k}$
$b_{t,k}$	bitrates of compressed $v_{t,k}$
B_t	bitrates of compressed motion features at t
F	baseline VOS method
Θ_τ	trainable parameters of module τ

in M_t. Denote the binary map of the k^{th} object in M_t as $m_{t,k}$, which could be formulated as,

$$m_{t,k}(i,j) = \begin{cases} 1 & M_t(i,j) = k \\ 0 & others, \end{cases} \tag{1}$$

in which (i, j) represents the coordinate of each pixel. The reason for doing so lies in that objects with different labels will essentially disturb the motion estimation between each other. The related analysis will be conduced in Section 4.4. Note that the $m_{t-1,k}$ is not available at the decoder, thus for each binary mask $m_{t,k}$, we utilize the previous reconstructed binary mask $\hat{m}_{t-1,k}$ as the reference mask to ensure encoder-decoder consistency. When $t = 2$, the reference mask is $m_{1,k}$ given by the labeled data. Then the binary mask $m_{t,k}$ and its reference is fed into ME module to estimate the changes from time step $t-1$ to t. The changes are represented by a motion feature $v_{t,k} = ME(m_{t,k}, \hat{m}_{t-1,k})$. After that, we encode the motion feature into a more compact representation $z_{t,k}$ by **FC** for transmission. We utilize $b_{t,k}$, the size of $z_{t,k}$, to evaluate the bit cost to compress each motion feature. Moreover, given all of the $b_{t,k}$ at time step t, the bit cost for the entire mask B_t is calculated by,

$$B_t = \sum_{i=1}^{N_t} b_{t,k}. \tag{2}$$

The compressed motion feature $\hat{v}_{t,k}$, together with the reference mask $\hat{m}_{t-1,k}$, are utilized to generate the predicted binary mask $\hat{m}_{t,k}$ by MC. Specifically, MC module consists of two parts. Firstly, $\hat{v}_{t,k}$ is warped on $\hat{m}_{t-1,k}$ to obtain a coarse prediction for $m_{t,k}$. Then the warped mask is further refined by a CNN-based network. Finally, we merge all of the predicted binary masks into the predicted

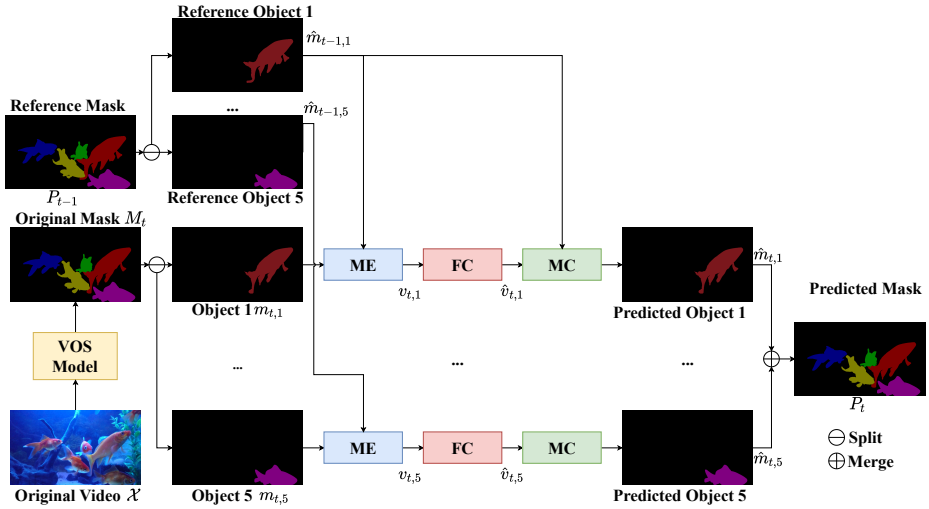


Fig. 2: The overall flowchart of the proposed approach. **ME**, **FC**, **MC** respectively denote the *Motion Estimation*, *Feature Compression*, and *Motion Comprehension* module. Firstly, the mask M_t is generated by a VOS model. Then the masks are split into several binary masks. For each binary mask $m_{t,k}$, we utilize the responding predicted mask $\hat{m}_{t-1,k}$ as the reference mask to extract the motion feature by ME. Then the motion feature $v_{t,k}$ is compressed by the FC module for transmission and decompression. After decompression, the compressed motion feature $\hat{v}_{t,k}$, together with $\hat{m}_{t-1,k}$, are utilized to predict $\hat{m}_{t,k}$, the predicted binary mask by MC. Finally, all of the binary masks are merged to the final reconstructed mask P_t. Note that we color some binary masks (including $m_{t,k}$, $\hat{m}_{t-1,k}$ and $\hat{m}_{t,k}$) for better comprehension.

mask P_t at time step t.

$$P_t(i,j) = \begin{cases} k & \hat{m}_{t,k} = 1 \\ 0 & others. \end{cases} \tag{3}$$

3.2 Detailed Architecture

ME. In our proposed framework, we employ a masked CNN-based optical flow estimation approach [26] to estimate the motion between the temporal adjacent binary mask. The visualization of ME module is shown in Fig 3. The splitted binary mask $m_{t,k} \in \{0,1\}^{W \times H}$ is fed into the optical flow estimation model to generate $o_{t,k} \in \mathbb{R}^{W \times H \times 2}$, which denotes the motion displacement between $\hat{m}_{t-1,k}$ and $m_{t,k}$. Then $o_{t,k}$ is element-wise multiplied with $\hat{m}_{t-1,k}$ to generate the motion feature $v_{t,k} \in \mathbb{R}^{W \times H \times 2}$. Instead of directly deploying the estimated flow, the element-wise multiplication have two advantages. One is that the multiplication could remove the disturbs of other regions. Note that the optical flow responding of the black region are the noises of the estimation. The other is the

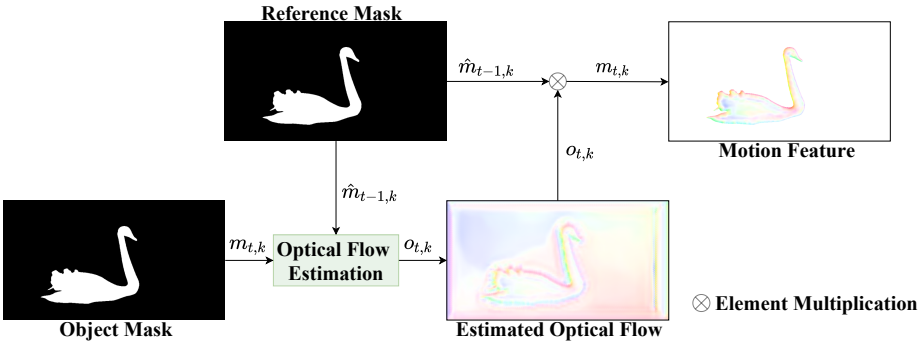


Fig. 3: Network of ME module. The example is the video *"swan"* in DAVIS 2017. The optical flow estimation module follows the implementation in [26].

multiplication could make the motion feature easier to compress. which means the bit cost of the proposed approach decreases. Compared to the optical flow or motion vector in traditional codecs, the proposed ME module could be end-to-end optimized with more flexibility.

FC. Since the motion feature extracted by ME module could not be transmitted directly. We deploy a modified hyperprior guided autoencoder proposed by [27], which is utilized to compress images with three channels. Therefore we modify the shape of the inputs and outputs to meet that of $v_{t,k}$. And the output of FC is the compressed motion feature $\hat{v}_{t,k}$, the bitstream $z_{t,k}$ and the bits of the bitstream $b_{t,k}$. The architecture of FC module is shown in Fig. 4. Every $v_{t,k} \in \mathbb{R}^{W\times H\times 2}$ is compressed into $z^1 \in \mathbb{R}^{\frac{W}{16}\times\frac{H}{16}\times 96}$ and $z^2 \in \mathbb{R}^{\frac{W}{64}\times\frac{H}{64}\times 64}$. z^1 denotes the representation of $v_{t,k}$. z^2 represents the parameters of the distribution to recover $\hat{v}_{t,k} \in \mathbb{R}^{W\times H\times 2}$.[4] Compared to other traditional compression methods, the proposed FC module is more efficient. Because the trainable FC module can be optimized to fit the signal distribution of $v_{t,k}$, which is different from that of images.

MC. The MC module is deployed to generate the predicted binary mask $\hat{m}_{t,k}$ by the compressed motion feature $\hat{v}_{t,k}$ and the reference mask $\hat{m}_{t-1,k}$. The flowchart of MC is shown in Fig. 5. Firstly, we warp the compressed motion feature $\hat{v}_{t,k}$ on the reference binary mask $\hat{m}_{t-1,k}$. However, the warped mask is far from an accurate prediction. Thus we deploy a CNN based refinement network after the warp operation. The network takes the reference mask and the warped mask as the inputs to generate the final predicted binary mask. Limited by the length of the paper, more comparisons between the warped masks before refinement and after refinement are shown in the supplementary materials.

[4] $z^1_{t,k}$ and $z^2_{t,k}$ for completeness but we drop the subscript (t,k) for simplicity.

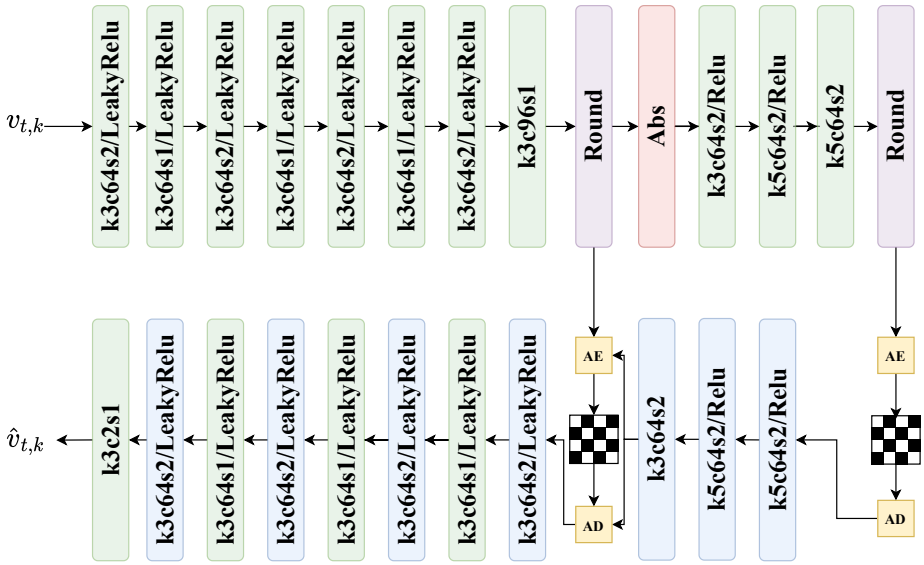


Fig. 4: Network architecture of FC module. The notations on the blocks denote the hyper-parameters. For example, "k3c64s2" indicates a convolution or deconvolution layer with 64 channels, the stride step is 2, and the kernel size is 3. The blocks in green represent convolution layers, and the blocks in blue denote deconvolution layers. The parameter of all of the LeakyRelu is set to be 0.1.

4 Experimentation

4.1 Training Details

Loss Function. The goal of the proposed framework is to leverage the accuracy of the predicted masks and minimize bits required for transmission simultaneously. Therefore, the optimization problem at time t could be formulated as:

$$\begin{aligned}\mathcal{L}_t &= \alpha R_t + \lambda D_t \\ &= \alpha \sum_{k=1}^{N_t} H(\hat{v}_{t,k}) + \lambda \sum_{k=1}^{N_t} d(m_{t,k}, \hat{m}_{t,k}),\end{aligned} \tag{4}$$

in which $d(m_{t,k}, \hat{m}_{t,k})$ denotes the distortions between $m_{t,k}$ and $\hat{m}_{t,k}$. In practice, we use the mean square error (MSE) in our experiments. $H(\cdot)$ represents the number of bits utilized to compress the motion feature. Actually, the bitrate estimation is a very complicated problem in end-to-end image compression. However, it is beyond the scope of this work. Therefore, we directly utilize the implementation of hyper-prior based entropy model [27] denoted by function $H(\cdot)$. λ indicates the Lagrange multiplier to adjust the trade-off between R_t and D_t. α is a binary parameter to remove the loss from R_t. **Training Details.** The training details of each stage is shown in Table 2. And the optical flow estimation net

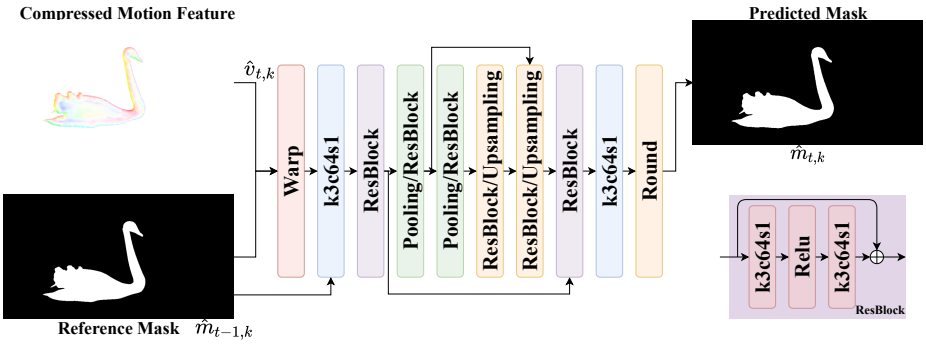


Fig. 5: Network architecture of the MC module. The notations on the blocks follows the rules mentioned in Fig. 4. All of the ResBlocks are deployed with the structure at the bottom right corner.

Table 2: Detailed Training Configuration

	Components Trained	α	λ	Learning Rate	Batch Size	Epochs
Stage I	ME,MC	0	1	1e-3	6	40
Stage II	FC	1	1e-5	1e-3	4	50
Stage III	FC,ME,MC	1	1e-5	1e-3	2	200

in ME module is initialized with a pre-train model[5]. The optimizer is AdamW with decay, ϵ equalling to 5e-5 and 1e-8, respectively. Although the approach is set to optimize for 200 epochs.

The entire scheme is implemented by using PyTorch 1.11.0 with CUDA 11.3. The simulation environment is based on Ubuntu 18.04 with one NVIDIA 3080ti graphic card. We train distinct models for different bitrate points to realize optimal performance.

4.2 Experimental Settings

Datasets & Preparation The experiments are deployed on three VOS benchmarks: DAVIS 2016 [28], DAVIS 2017 [29], and YouTube-VOS [30]. DAVIS 2016 and DAVIS 2017 are small datasets with 50 and 120 video sequences, respectively. YouTube-VOS is a large-scale dataset with 3945 video sequences. All the video sequences are converted from RGB to YUV420 format, which is widely employed color space for traditional codecs such as x265 and VVEnc[6] to compress.

For the test on three datasets, we use the same model trained by DAVIS 2017 dataset. Note that our framework does not use enormous videos for training. Only 60 video sequences are utilized. This configuration proves that our work is simple and robust for unseen VOS datasets.

[5] https://github.com/zacjiang/GMA
[6] VVEnc [31] is a light-weighted implementation of the reference software of VVC [32].

Table 3: Experimental Results on DAVIS2016 and DAVIS2017

Dataset	VOS Model	Method	Bitrate↓	$\mathcal{J}_m$↑	$\mathcal{F}_m$↑	$(\mathcal{J}\&\mathcal{F})_m$↑
DAVIS 2016 [28]	AOT [33]	Original	-	0.9014	0.9217	0.9117
		x265(baseline)	0.0198	0.7944	0.8118	0.8031
		x265+Ours	0.0168	0.8728	0.9044	0.8886
	STCN [34]	Original	-	0.9042	0.9305	0.9172
		x265(baseline)	0.0198	0.7517	0.7848	0.7683
		x265+Ours	0.0158	0.8814	0.9148	0.8981
DAVIS 2017 [29]	AOT [33]	Original	-	0.8251	0.8791	0.8521
		x265(baseline)	0.0209	0.7165	0.7608	0.7386
		x265+Ours	0.0178	0.8056	0.8652	0.8354
	STCN [34]	Original	-	0.8200	0.8862	0.8531
		x265(baseline)	0.0209	0.6734	0.7161	0.6947
		x265+Ours	0.0158	0.7890	0.8641	0.8265

Metrics. For the accuracy on VOS, we follows the standard criteria from [28]: Jaccard Index $\mathcal{J}$ and $\mathcal{F}$-scores, which represent the region similarity and contour accuracy, respectively. Additionally, we also report some detailed performance for each dataset: {Mean↑, Recall↑, Decay↓}x{$\mathcal{J}$,$\mathcal{F}$} for DAVIS 2017 and {Seen, Unseen}x{$\mathcal{J}$,$\mathcal{F}$} for YouTube-VOS. The experimental results of these metrics are provided in supplementary materials

In addition to the traditional VOS evaluation metrics, we have added the evaluation metric: bitrate. Bitrate denotes the compression ratio of codecs. Generally speaking, the method with less bitrate is better given the same performance. Specifically, we utilize bits-per-pixel (bpp) as the evaluation of bitrate in the experiments. As the proposed approach is an additional module to existing codecs. Thus the bitrate of our work is the sum of it and the baseline codec for a fair comparison.

Base Video Codecs. We choose two conventional codecs (x265 and VVEnc) as the baselines. Considering the efficiency and effectiveness, we choose the x265 library in FFmpeg with *veryfast* preset for main experiments. And VVEnc is utilized to compress videos for the supplementary experiments. The compressed sequences will be shared to encourage others to research compressed video segmentation.

Base VOS Models. We choose the AOT and STCN as the base model in our framework.[7] AOT employs a Long Short-Term Transformer to construct hierarchical matching and propagation. STCN combines the computational advantages of temporal convolutional networks with the representational power and robustness of stochastic latent spaces.

4.3 Experimental Results

In this subsection, the experiments are deployed with different VOS models and codecs for comparison. Please refer to supplementary material for more details

[7] AOT and STCN are representative VOS models with codes and models available.

Table 4: Experimental Results on YouTube-VOS dataset.

VOS Model	Method	Bitrate↓	$\mathcal{J}_s$↑	$\mathcal{F}_s$↑	$\mathcal{J}_u$ ↑	$\mathcal{F}_u$ ↑	$\mathcal{G}$↑
AOT [33]	Original	-	0.8387	0.7990	0.8880	0.8848	0.8526
	x265(baseline)	0.0159	0.7966	0.8429	0.7530	0.8450	0.8094
	x265+Ours	0.0121	0.8265	0.8811	0.7782	0.8742	0.8400
STCN [34]	Original	-	0.8259	0.8695	0.7946	0.8772	0.8418
	x265(baseline)	0.0159	0.7867	0.8264	0.7372	0.8200	0.7867
	x265+Ours	0.0122	0.8149	0.8634	0.7726	0.8656	0.8291

and commands about the x265/VVEnc settings. Furthermore, the concrete VOS models are also provided in it.

The performance of the proposed framework compared to x265 codec is reported in Table 3 and Table 4. On DAVIS 2016, our work achieves 8% and 13% accuracy($\mathcal{J}\&\mathcal{F}$) improvement for AOT and STCN with 35% and 25% bitrate saving, respectively. On DAVIS 2017, our work achieves 10% and 12% accuracy($\mathcal{J}\&\mathcal{F}$) improvement for AOT and STCN with 14% and 24% bitrate saving, respectively. On YouTube-VOS, we achieve 3.1% and 4.3% $\mathcal{G}$($\mathcal{G}$ denotes the score calculated by the evaluation server[8]) improvement for AOT and STCN with 20% and 23% bitrate saving. The performance improvement over all of the VOS models and datasets reveals the generality of our framework.

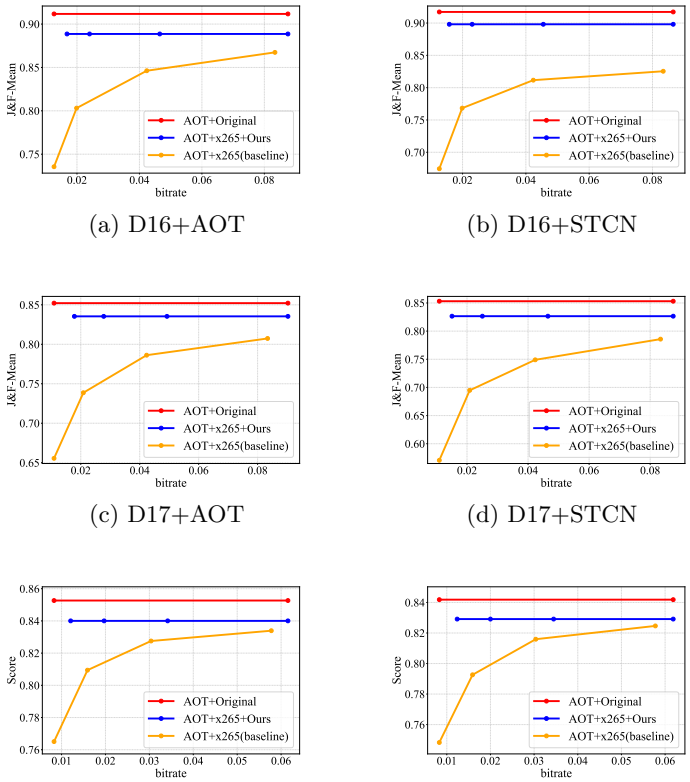
Rate-Performance Curves. We report the Rate-Performance (RP) curves in Fig. 6. The QP set of both x265 and x265+Ours are {32,37,42,47}. Taking **DAVIS 2016+AOT** in Fig. 6(a) as an example, the line in orange denotes the RP curve of video sequences compressed by x265 codec without the proposed framework. The line in blue indicates the RP curve of video sequences compressed by x265 with the proposed framework. From these curves, several conclusions could be drawn. First of all, the proposed framework outperforms x265 at every bitrate. Secondly, the accuracy loss of our method is minimal compared to the results on the original videos. Last but not the least, the proposed framework could achieve remarkable performance for different datasets and VOS methods.

4.4 Ablation Studies and Modular Analysis

FC Module. We utilize a hyperprior guided auto-encoder model to compress the underlying motion feature. Actually, it is also reasonable to deploy a traditional codec to encode the feature. Thus we investigate the efficiency of FC module by using x265 codec for comparison. As shown in Table 5, experimental results prove that the proposed FC module could efficiently compress the motion vector. It is because that the learned parameters in FC module were optimized to fit the distribution of motion features during training.

MC Module. In our framework, we propose a CNN-based MC model to refine the predicted mask after the warp operation. Another alternative is to utilize the warped mask directly as the prediction results. Thus we conduct a set of

[8] https://competitions.codalab.org/competitions/20127#participate-submit_results



(a) D16+AOT (b) D16+STCN

(c) D17+AOT (d) D17+STCN

(e) YTB+AOT (f) YTB+STCN

Fig. 6: Rate-Performance curves of our approach. D16,D17, and YTB represent DAVIS 2016, DAVIS 2017 and YouTube-VOS, respectively. Original denotes that the videos are uncompressed.

ablation study to verify the effectiveness of the MC module. From the visualization in Table 6, it is obvious that the propose MC module significantly improve the quality of the predicted mask. Then the experimental results shows that the $(\mathcal{J}\&\mathcal{F})_m$ of the approach without MC module will drop about 30% on DAVIS-2017 dataset.

More Codecs. We conduct experiments on more codecs to verify the generality of our framework. The experimental results on are shown in Table 7. Although VVEnc is the most efficient conventional codecs, the proposed framework still achieve 9.1% improvement on DAVIS 2017 with 32% bits saving.

More Bitrates. By adjusting the λ in the loss function ($\lambda = 10^{-a}$, a=2,3,4,5), the proposed framework are trained for different bitrates. To valid the performance at different bitrates, we conduct experiments on DAVIS 2017. Note that the bitrate is calculated without that of codecs. As shown in Table 8, with the increase of the bitrate, the difference compared to the analysis performance of the

Table 5: Ablation Study for FC Module on DAVIS 2017 Dataset.

Method	**Bitrate**↓	$\mathcal{J}_m$↑	$\mathcal{F}_m$↑	$(\mathcal{J}\&\mathcal{F})_m$↑
x265+Ours(x265)	0.0209	0.8052	0.8655	0.8353
x265+Ours(FC)	0.0178	0.8056	0.8652	0.8354

Table 6: Ablation Study for MC Module on DAVIS 2017 Dataset.

Method	Bitrate↓	$\mathcal{J}_m$↑	$\mathcal{F}_m$↑	$(\mathcal{J}\&\mathcal{F})_m$↑
x265+Ours(Warp)	0.0178	0.4725	0.5358	0.5041
x265+Ours(MC)	0.0178	0.8056	0.8652	0.8354

original videos become smaller and smaller. This experimental result demonstrates that our model could be adapted to different application scenarios by adjusting the bitrate.

Direct Comparison to Codecs. Rethinking the paradigm illustrated in 1b, it is feasible to utilize existing compression codecs to compress the VOS masks to replace our approach. Therefore we conduct a comparative experiment. We utilize x265 as the mask codec to compress the masks generated by AOT on DAVIS 2017. The experimental results are shown in Table 9. It is obvious that the masks encoded by x265 are not accurate for VOS anymore. Then main reason to the degradation is the noise caused by codec at extremely low bitrate.

5 Discussions

Extensibility. The proposed framework can be easily extended to other mask-based vision tasks. Mask based vision tasks denotes tasks where the output is a mask related to the semantics of the video such as video instance segmentation and multi object tracking. To run our work on other tasks, simply replace the VOS model with the model for the corresponding task.

Robustness. As reported in Table 8, the proposed framework could utilize an extremely low bitrate to achieve 95% performance compared to that on original video sequences. It proves that our work could handle practical scenarios with low network.

Generality. The generality of the proposed framework is manifested in two aspects. Firstly, the proposed framework does not rely on any VOS methods. Therefore, our framework can also be adapted to new VOS methods with higher

Table 7: Experimental Results for VVEnc on DAVIS 2017 Dataset.

Method	**Bitrate**↓	$\mathcal{J}_m$↑	$\mathcal{F}_m$↑	$(\mathcal{J}\&\mathcal{F})_m$↑
Original	-	0.8251	0.8791	0.8521
VVEnc(baseline)	0.0179	0.7235	0.7646	0.7440
VVEnc+Ours	0.0121	0.8056	0.8652	0.8354

Table 8: Experimental Results for Different $\lambda(\lambda = 10^{-a})$

a	**Bitrate**↓	$\mathcal{J}_m$↑	$\mathcal{F}_m$↑	$(\mathcal{J}\&\mathcal{F})_m$↑
Original	-	0.8251	0.8791	0.8521
2	0.0262	0.8174	0.8706	0.8440
3	0.0162	0.8135	0.8690	0.8413
4	0.0069	0.8056	0.8652	0.8354
5	0.0023	0.7752	0.8502	0.8127

Table 9: Experimental Results for Direct Comparison to Codecs

Method	**Bitrate**↓	$\mathcal{J}_m$↑	$\mathcal{F}_m$↑	$(\mathcal{J}\&\mathcal{F})_m$↑
Original	-	0.8251	0.8791	0.8521
x265	0.0072	0.5856	0.6488	0.6172
Ours	0.0069	0.8056	0.8652	0.8354

performance. Secondly, the input of the proposed frameworks are masks rather than objects, which means the category information of the objects are not utilized in our work. Thus it could be easily transferred to other datasets without fine-tuning.

6 Conclusion

In this paper, we propose a simple and effective framework to improve the performance of semi-supervised VOS on compressed video sequences by preserving the semantic information during compression procedure. Such a framework could effectively diminish the degradation of VOS performance during lossy compression. Moreover, the proposed is a plug-and-play framework which means that it does not rely on a specific VOS method or a specific codec. Concretely, it could improve the analysis performance by up to 13% at similar bitrates. As VCM has been an emerging research topic recently, we provide an promising solution for efficient and high performance compressed VOS.

References

1. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the h. 264/avc video coding standard. IEEE Transactions on Circuits and Systems for Video Technology **13** (2003) 560–576
2. Luiten, J., Voigtlaender, P., Leibe, B.: Premvos: Proposal-generation, refinement and merging for video object segmentation. In: Asian Conference on Computer Vision, Springer (2018) 565–580
3. Maninis, K.K., Caelles, S., Chen, Y., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: Video object segmentation without temporal information. IEEE Transactions on Pattern Analysis and Machine Intelligence **41** (2018) 1515–1530
4. Oh, S.W., Lee, J.Y., Xu, N., Kim, S.J.: Video object segmentation using space-time memory networks. In: IEEE International Conference on Computer Vision. (2019) 9226–9235

5. Caelles, S., Maninis, K.K., Pont-Tuset, J., Leal-Taixé, L., Cremers, D., Van Gool, L.: One-shot video object segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017) 221–230
6. Robinson, A., Lawin, F.J., Danelljan, M., Khan, F.S., Felsberg, M.: Learning fast and robust target models for video object segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2020) 7406–7415
7. Voigtlaender, P., Leibe, B.: Online adaptation of convolutional neural networks for the 2017 davis challenge on video object segmentation. In: The 2017 DAVIS Challenge on Video Object Segmentation-CVPR Workshops. Volume 5. (2017)
8. Li, X., Loy, C.C.: Video object segmentation with joint re-identification and attention-aware mask propagation. In: European Conference on Computer Vision. (2018) 90–105
9. Li, X., Qi, Y., Wang, Z., Chen, K., Liu, Z., Shi, J., Luo, P., Tang, X., Loy, C.C.: Video object segmentation with re-identification. arXiv preprint arXiv:1708.00197 (2017)
10. Perazzi, F., Khoreva, A., Benenson, R., Schiele, B., Sorkine-Hornung, A.: Learning video object segmentation from static images. In: IEEE Conference on Computer Vision and Pattern Recognition. (2017) 2663–2672
11. Jamrozik, M.L., Hayes, M.H.: A compressed domain video object segmentation system. In: International Conference on Image Processing. Volume 1., IEEE (2002) I–I
12. Porikli, F., Bashir, F., Sun, H.: Compressed domain video object segmentation. IEEE Transactions on Circuits and Systems for Video Technology **20** (2009) 2–14
13. Babu, R.V., Ramakrishnan, K., Srinivasan, S.: Video object segmentation: a compressed domain approach. IEEE Transactions on Circuits and Systems for Video Technology **14** (2004) 462–474
14. Tan, Z., Liu, B., Chu, Q., Zhong, H., Wu, Y., Li, W., Yu, N.: Real time video object segmentation in compressed domain. IEEE Transactions on Circuits and Systems for Video Technology **31** (2020) 175–188
15. Xu, K., Yao, A.: Accelerating video object segmentation with compressed video. In: IEEE Conference on Computer Vision and Pattern Recognition. (2022) 1342–1351
16. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. IEEE Transactions on Circuits and Systems for Video Technology **22** (2012) 1649–1668
17. Lu, G., Zhang, X., Ouyang, W., Chen, L., Gao, Z., Xu, D.: An end-to-end learning framework for video compression. IEEE Transactions on Pattern Analysis and Machine Intelligence **43** (2020) 3292–3308
18. Hu, Z., Lu, G., Xu, D.: Fvc: A new framework towards deep video compression in feature space. In: IEEE Conference on Computer Vision and Pattern Recognition. (2021) 1502–1511
19. Li, J., Li, B., Lu, Y.: Deep contextual video compression. Advances in Neural Information Processing Systems **34** (2021) 18114–18125
20. Chen, Z., He, T.: Learning based facial image compression with semantic fidelity metric. Neurocomputing **338** (2019) 16–25
21. Wang, S., Wang, S., Yang, W., Zhang, X., Wang, S., Ma, S., Gao, W.: Towards analysis-friendly face representation with scalable feature and texture compression. IEEE Transactions on Multimedia (2021)
22. Le, N., Zhang, H., Cricri, F., Ghaznavi-Youvalari, R., Rahtu, E.: Image coding for machines: an end-to-end learned approach. In: IEEE International Conference on Acoustics, Speech and Signal Processing. (2021) 1590–1594

23. Huang, Z., Jia, C., Wang, S., Ma, S.: Hmfvc: A human-machine friendly video compression scheme. IEEE Transactions on Circuits and Systems for Video Technology (2022)
24. Chamain, L.D., Racapé, F., Bégaint, J., Pushparaja, A., Feltman, S.: End-to-end optimized image compression for machines, a study. In: IEEE Data Compression Conference. (2021) 163–172
25. Le, N., Zhang, H., Cricri, F., Ghaznavi-Youvalari, R., Tavakoli, H.R., Rahtu, E.: Learned image coding for machines: A content-adaptive approach. In: IEEE International Conference on Multimedia and Expo. (2021) 1–6
26. Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R.: Learning to estimate hidden motions with global motion aggregation. In: International Conference on Computer Vision. (2021) 9772–9781
27. Minnen, D., Ballé, J., Toderici, G.D.: Joint autoregressive and hierarchical priors for learned image compression. Advances in Neural Information Processing Systems **31** (2018)
28. Perazzi, F., Pont-Tuset, J., McWilliams, B., Van Gool, L., Gross, M., Sorkine-Hornung, A.: A benchmark dataset and evaluation methodology for video object segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition. (2016) 724–732
29. Pont-Tuset, J., Perazzi, F., Caelles, S., Arbeláez, P., Sorkine-Hornung, A., Van Gool, L.: The 2017 davis challenge on video object segmentation. arXiv preprint arXiv:1704.00675 (2017)
30. Xu, N., Yang, L., Fan, Y., Yue, D., Liang, Y., Yang, J., Huang, T.: Youtube-vos: A large-scale video object segmentation benchmark. arXiv preprint arXiv:1809.03327 (2018)
31. Wieckowski, A., Brandenburg, J., Hinz, T., Bartnik, C., George, V., Hege, G., Helmrich, C., Henkel, A., Lehmann, C., Stoffers, C., et al.: Vvenc: An open and optimized vvc encoder implementation. In: IEEE International Conference on Multimedia & Expo Workshops, IEEE (2021) 1–2
32. Bross, B., Chen, J., Ohm, J.R., Sullivan, G.J., Wang, Y.K.: Developments in international video coding standardization after avc, with an overview of versatile video coding (vvc). Proceedings of the IEEE **109** (2021) 1463–1493
33. Yang, Z., Wei, Y., Yang, Y.: Associating objects with transformers for video object segmentation. Advances in Neural Information Processing Systems **34** (2021) 2491–2502
34. Aksan, E., Hilliges, O.: Stcn: Stochastic temporal convolutional networks. International Conference on Learning Representations (2019)