# Rethinking Video Error Concealment: a Benchmark Dataset and a Partition-based Method

Anonymous

**Abstract**—Error concealment is an important technique to restore a damaged video bistream. Although data-driven inpainting methods can be directly applied to video error concealment, existing mask patterns are remarkably different from the practical damaged video bitstream, which causes a great impact on the repairing effect of video quality. To rethink the gap between existing inpainting schemes and practical video transmission characteristics, we have established a new video error concealment (VEC) benchmark dataset. Specifically, different video sequences compressed by different encoders are collected, and various loss types are generated to satisfy different packet loss scenarios. VEC can be regard as a benchmark for video error concealment research. Based on the proposed dataset, we design a partition-based video error concealment method. Specifically, it divides all patches into repetitive patches and independent patches, and further utilizes the replication module and generation module to repair them respectively. In addition, we introduce a binary quantization compression strategy, which reduces its space storage requirements and computational costs while ensuring video quality.

**Index Terms**—Error Concealment, Binary Quantization, Transformer.

## 1 Introduction

Due to the popularity of mobile smart terminals and online social media, video services have a surge in demand for video transmission [1]. However, video transmission is done over error-prone channels, which will introduce packet errors. Real-time applications, such as live broadcasting and video conferencing, usually use transport protocols without a retransmission mechanism and cannot correct packet errors. Errors in the bitstream hinder the reconstruction of the video, resulting in video quality degradation. With the increasing amount of ultra-high-definition video data, it is more prone to packet errors in the environment of network congestion. To address this problem, video error concealment techniques are developed to conceal errors by exploiting both spatial and temporal redundancy. But the latest video compression standards, such as H.265/High Efficiency Video Coding (HEVC) and H.264/Advanced Video Coding (AVC), only focus on the improvement of coding efficiency and lack research on error concealment algorithms [2], [3]. Therefore, it is necessary to develop a general and efficient video error concealment algorithm to meet the needs of various video applications.

Since the dependence of learning-based algorithms on a large amount of data, the lack of suitable datasets becomes a major obstacle for the research video error concealment algorithms. To this end, we propose a novel video error concealment dataset (VEC), which jointly considers compression and transmission properties in video communication systems. Specifically, we collect a large number of diverse high-quality videos and compress them with different video compression standards to generate compression artifacts, while simulating slice or tile data loss during transmission.

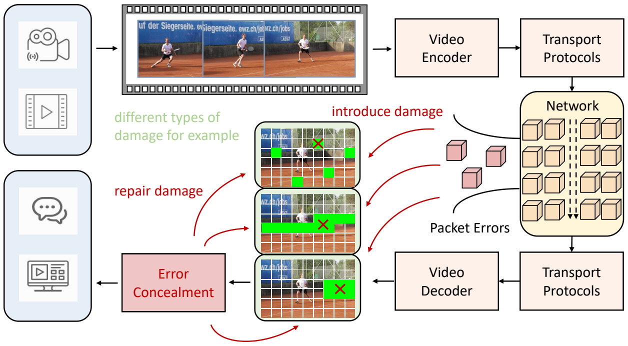


Fig. 1. **An illustration of error concealment.** After the video is encoded into a bit stream, packet loss inevitably occurs during network transmission. It further leads to corruption of the reconstructed video. Error concealment is expected to repair the video corruption caused by these packet losses, thereby improving the video quality at the receiving end.

Based on the proposed dataset, we further explore efficient video error concealment methods to meet practical application requirements. In recent years, transformer has been widely used for the visual task of video inpainting. These methods rely on spatio-temporal information propagation between frames to fill holes with plausible and coherent content. However, there are many spatio-temporal redundant patches in continuous video sequences, *e.g.*, repetitive background patches and content independent image patches. Without considering the above properties, existing state-of-the-art methods treat all patches equally, which leads to unnecessary feature interactions and the introduction of noise information. This further leads to a degradation in the quality of the reconstructed video. To further address this issue, we propose a partition-based video error concealment framework, which divides patches in video frames into repetitive patches and independent

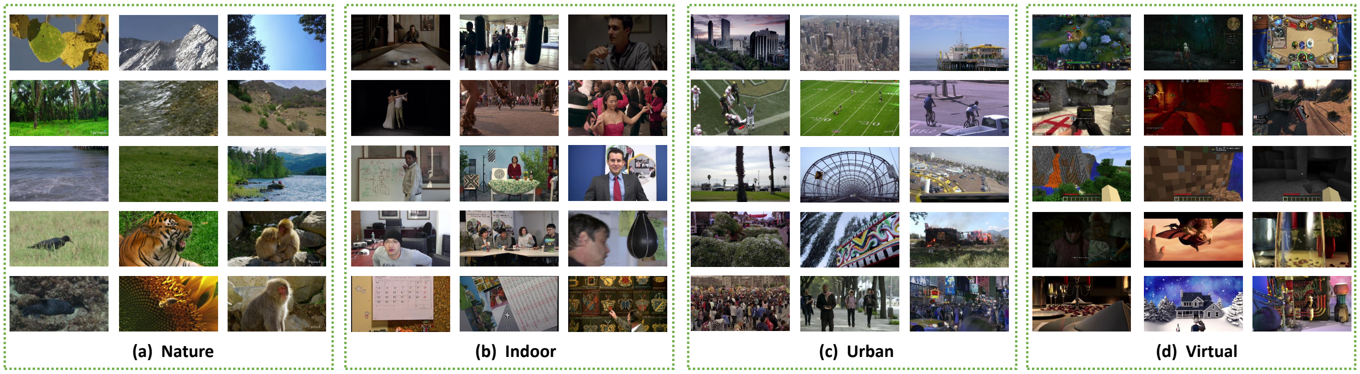

(a) Nature (b) Indoor (c) Urban (d) Virtual

Fig. 2. **Examples of the proposed VEC dataset**. The collected video content includes four categories: (a) *Nature* which is mainly composed of scenery and animals, (b) *Indoor* which mainly contains indoor people and objects, (c) *Urban* which includes urban landscape and outdoor sports, and (d) *Virtual* which mainly consists of games and animations.

patches, and utilizes the replication module and generation module for inpainting, respectively.

In addition, since error concealment techniques are usually deployed on resource-constrained devices, larger models will require more spatial storage requirements and computational costs. Although the existing state-of-the-art methods achieve better video quality, it needs more storage requirements and higher computational cost, which are the key factors hindering deployment in real-world scenarios. To address this issue, we further introduce a binary quantization strategy to reduce the storage requirements and computation cost of the proposed framework, which brings certain advantages in practical deployment.

In this paper, our contributions can be summarized as follow: On the one hand, we propose a novel video error concealment dataset (VEC), which jointly considers compression and transmission properties in video communication systems. Specifically, we collect a large number of diverse high-quality videos and compress them with different video compression standards to generate compression artifacts, while simulating packet data loss during transmission. It can be regarded as a research benchmark for future deep video error concealment techniques. On the other hand, based on the proposed dataset, we design a partition-based video error concealment method that improves video visual quality. Specifically, it divides all patches into repetitive patches and independent patches, and further utilizes the replication module and generation module to repair them respectively. In order to further promote its deployment and application, we introduce a binary quantization compression strategy, which reduces its space storage requirements and computational costs while ensuring video quality.

## 2 Related Works

### 2.1 Existing Datasets

Currently, the most popular video inpainting datasets include Youtube-VOS and DAVIS. They simulate the loss of video content by masking random region or specific object. In addition, a video mask dataset, which provides a variety of temporally dynamic mask patterns, has been also proposed to promote the performance of video inpainting. However, in these existing datasets, the video content and mask type are considered separately. Aiming at the error concealment scenario, the proposed VEC dataset jointly considers the distortion type, loss type, and loss ratio, which is totally different from the previous benchmark datasets.

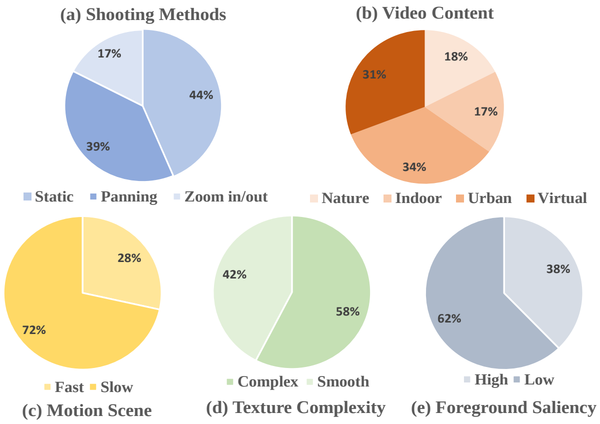

(a) Shooting Methods (b) Video Content (c) Motion Scene (d) Texture Complexity (e) Foreground Saliency

Fig. 3. **Illustration of the statistical distributions of VEC**. There are five non-conflicting attributes, including (a) Shooting methods, (b) Video Content, (c) Motion Scene, (d) Texture Complexity, and (e) Foreground Saliency.

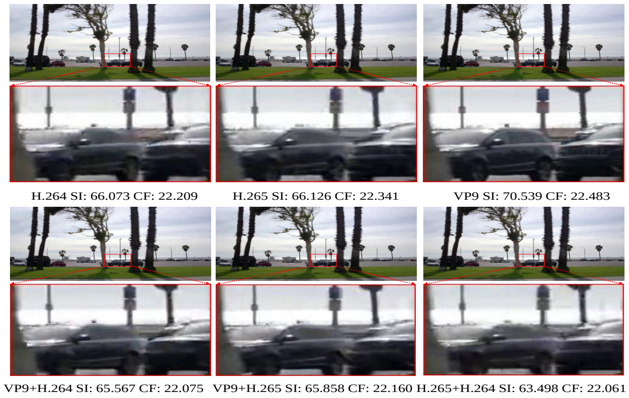
### 2.2 Error Concealment

In computer vision, most of the existing video inpainting approaches are designed for video editing tasks, such as object removal and video editing. In the above application, they restore the loss of video content simulated by some free masks. However, for a real error concealment scenario, video content is usually missed with the maximum coding block as the basic unit. It is determined by the characteristics of the video compression and transmission system. Therefore, these well-designed inpainting approaches are

## 3 Proposed Database

### 3.1 Video Collection

Our goal is to build a video error concealmen (VEC) dataset for error concealment. First of all, we have collected a large number of lossless videos from



H.264 SI: 66.073 CF: 22.209 H.265 SI: 66.126 CF: 22.341 VP9 SI: 70.539 CF: 22.483

VP9+H.264 SI: 65.567 CF: 22.075 VP9+H.265 SI: 65.858 CF: 22.160 H.265+H.264 SI: 63.498 CF: 22.061

Fig. 4. **Illustration of different distortion types** by different coding combinations. The top row shows the results compressed by a single encoder, and the bottom row shows the results compressed by double encoders.

https://media.xiph.org/video/derf/ and the joint video exploration team (JVET). To be consistent with the above common video format, the aspect ratio of all video data is adjusted to 16:9. Different video contents from various applications have been covered in VEC, such as *live TV*, *sports games*, and *video conference* as shown in Fig.2. The raw format of the collected video is 8-bit *YCbCr4:2:0*, which is the most commonly used video format. A total of 87 videos with the resolution of 1280×720, 1920×1080, and 4096×2160 are collected and uniformly resized to 1280×720. According to the change of different scenes, 172 independent video sequences with non-overlapping 100 frames are split from the original videos. In the end, all 172 videos have been compressed by different codecs (*i.e.*, H.265, H.264, and VP9) to obtain a total of 1032 videos, which can be divided into training, validation and test sets (828/102/102) without overlapping. The detail information of video statistics and compression settings will be further discussed in Sec.3.2 and Sec.3.3, respectively.

## 3.2 Dataset Analysis

In this section, we provide some statistical analysis for VEC. We demonstrate all videos by multiple attributes from different perspectives, including shooting methods, video content, motion scene, texture complexity, and foreground saliency. The distribution of VEC in terms of different attributes are show in Fig.3 and the proposed VEC has a diversified video content, which covers abundant video scenarios. The details of these attribute are considered as follow:

**Shooting Method**. A shooting method has a great impact on the video compression quality. It is mainly composed of static, panning, and zoom in/out.

**Video Content**. The video content determines which category a video belongs to. In VEC, each video sequence is divided into one of the four categories: *nature*, *indoor*, *urban*, or *virtual*.

**Motion Scene**. Motion scene also greatly affects the compression performance.In VEC, each video is divided into one of the two categories: fast or slow. For error concealment, whether the loss of information occurs on a moving target has a significant impact on the final repaired performance.

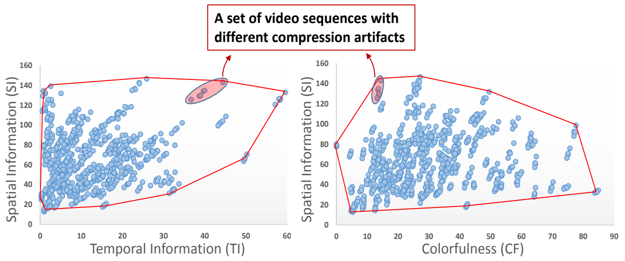


Fig. 5. **Statistical distribution of the proposed VEC dataset** in terms of SI, TI and CF. The red contour represents the range covered by VEC, indicating the richness of VEC. And two small red regions represent a group of videos compressed by multiple codecs.

**Texture Complexity**. The complexity of image texture is also an important factor in a video sequence. In VEC, each video is divided into one of the two categories: complex or smooth. It is obviously more challenging to restore missing blocks in the region with the complex texture.

**Foreground Saliency**. The saliency of foreground [4] plays an essential in the video quality perception. The information loss at the boundary between the foreground and the background will cause more difficulties for video restoration.

## 3.3 Distortion Types

A lossy video bitstream is usually accompanied by different compression artifacts, which is a significant difference compared with other artifacts [5]. Compression artifacts are mainly caused by two aspects: 1) the quantization for each block will cause discontinuity on the block boundary, which leads to the generation of spatial artifacts. 2) the motion compensation used for inter-frame coding will propagate artifacts from the current frame to the subsequent frames, resulting in temporal artifacts [6], [7]. Compression artifacts will result in the loss of some texture details, thereby weakening the temporal and spatial correlation of the adjacent frames. It brings additional difficulties to practical error concealment.

To be consistent with actual error concealment scenarios, some populor video compression standards, including H.265, H.264, and VP9, are used to simulate the compression distortion. Considering that there will be a video that may be shared after being played and compressed by several times, three codecs have been used to compress the related videos in a cross-combination manner, including “VP9+H.264”, “VP9+H.265”, and “H.265+H.264”, to meet a more complex compression artifact produced by the above situation. For instance, artifacts caused by some codecs are shown in Fig.4. All the compression process is completed by *FFMPEG*, in which the coding mode and constant rate factor are set to “inter” and 35, respectively. The statistical distribution of VEC in terms of spatial information (SI), temporal information (TI), and colourfulness (CF) [8] are also provided in Fig.5.

## 3.4 Loss Types

In video transmission, each slice is packaged into an independent transmission unit (*e.g.*, network abstraction layer (NAL) in H.264 and H.265) for efficient communication.

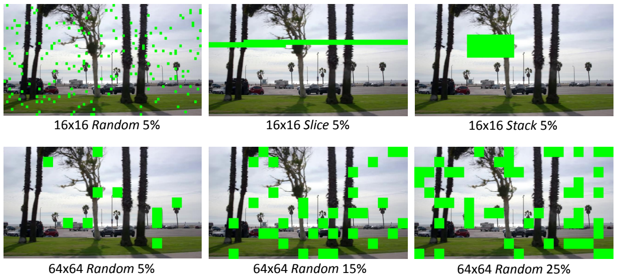


Fig. 6. **Examples of different loss types.** The top row shows the loss types of H.264, and the bottom row shows the loss types of H.265 and VP9.

When the bitstream data is transmitted by different networks, the occurrence of bit errors will cause a packet loss. Assume that even a bit error will cause the complete slice in a data packet to be damaged. Finally, the packet loss will lead to the degradation of a slice region missed in the received video. To satisfy this type of video missing, we generate a variety of loss types, including *random type* which discards blocks randomly, *slice type* which discards successive blocks in a raster scan order, and *stack type* which discards adjacent blocks.

Generally, different codecs use different maximum coding block division strategies for a higher compression ratio [9], [10]. For example, H.264 supports flexible block partitions ranging from 4×4 to 16×16. H.265 uses a quad-tree structure, in which the largest size of s coding block is 64×64. Therefore, it means that videos compressed with different codecs have more complicated block sizes when a packet loss occurs during transmission. Assume that the size of a damaged block is consistent with the largest possible coding block size in each compression standard. Then, the sizes of the related damaged blocks corresponding to H.264, H.265, and VP9 are set as 16×16, 64×64, and 64×64, respectively. They are the basic unit for preparing the test videos on VEC. Some examples are show in Fig.6.

## 4 Proposed Mothod

### 4.1 Problem Formulation

To guarantee the efficiency and quality of video error concealment, we propose a new video error concealment method. The overall framework of our method is shown in Fig. 7. The input corrupted video is generated from given videos $\{X_t \in \mathbb{R}^{H\times W\times 3} \mid t = 1, \cdots, T\}$ and corresponding masks $\{M_t \in 0, 1^{H\times W\times 3} \mid t = 1, \cdots, T\}$. We mine spatio-temporal correlation information from video sequences to repair damage in the target frame regions. Existing methods typically adopt the *Transformer* architecture, which first divides the the video sequence into patches and embeds them into feature space. Then, a self-attention mechanism is employed to guide the fusion of information between image patches by computing the spatio-temporal correlation between patches. Finally, a series of features are reconstructed into video $\{\hat{X}_t \in 0, 1^{H\times W\times 3} \mid t = 1, \cdots, T\}$ by the decoder. The goal of repairing is to make $\{\hat{X}_t \in \mathbb{R}^{H\times W\times 3} \mid t = 1, \cdots, T\}$ as close as possible to the real video $\{X_t \in \mathbb{R}^{H\times W\times 3} \mid t = 1, \cdots, T\}$ to achieve the effect of hiding video transmission errors and repairing damaged regions.

However, there exist a large number of spatio-temporal redundant patches in continuous video sequences, such as repeated background patches and spatio-temporally independent patches in video frames. Existing state-of-the-art methods do not take this video characteristic into account and treat all image patches equally, while redundant patches introduce noise in the fusion between different patch features. This will cause reconstructed video frames to contain unrealistic content. Furthermore, unnecessary feature interactions between video frame patches degrade the quality of inpainting.

In order to solve the above problems, we propose a patch adaptive partition VEC method, which allows the network to perform information interaction for repair only between highly correlated repetitive patches, while independent patches rely on mining their own spatial information for repair. Our method consists of five modules: *coarse repair*, *patch adaptive division*, *content replication*, *content generation*, and *de-artifact reconstruction*. First, all corrupted video frames are separately input into the coarse repair module, which utilizes their own spatial correlation information to perform the repair. Then, the coarsely repaired frames are divided into patches, and the spatio-temporal context information is further utilized for fine repairing. To reduce unnecessary feature interactions and improve repairing quality, we mine spatio-temporal correlation information to repaire lost regions for repetitive patches, while for independent patches, we leverage different channel features as priors to guide lost regions to generate real content. Finally, the repaired patches are aggregated and fed into a reconstruction network to remove spatio-temporal artifacts at patch boundaries to obtain a complete sequence of video frames. In the following sections, we discuss the design details of the network modules in detail and give the adopted stepwise loss function and joint training strategy.

### 4.2 Patch Adaptive Division

The spatio-temporal redundant patches are quite common in daily videos. For example, in videos with obvious foreground objects, the unchanging background can be regarded as a kind of spatio-temporal redundancy. In order to eliminate the obstacle of spatio-temporal redundancy to spatio-temporal feature propagation, we propose a similarity-guided patch adaptive partition module. The module divides all patches into independent patches and repetitive patches by measuring the similarity between image patches. Specifically, an independent patch is defined as a patch that has a low similarity with other patches and hinders the propagation of spatio-temporal features, a repetitive patch is defined as a patch that has a high similarity with other patches and can provide spatio-temporal information. According to the above definition, we compute the similarity score $S_t^i$ between a particular patch $C_t^i$ of a frame $C_t$ and patches $C_{t+\Delta t}$ ($\Delta t = 0$ represents the same frame) in other frames $C_{t+\Delta t}^i$ as follows:

$$S_t^i = \frac{1}{N \times T} \sum_{\Delta t=-\frac{T-1}{2}}^{\frac{T-1}{2}} \sum_{j=1}^{N} (\phi_q(C_t^i)\phi_k(C_{t+\Delta t}^j)^T), \quad (1)$$

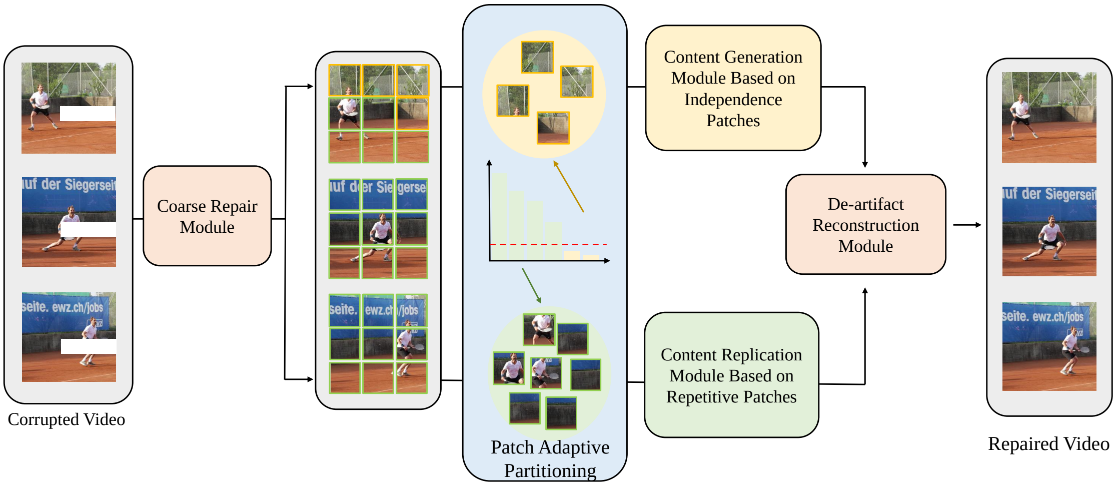

Corrupted Video
Coarse Repair Module
Patch Adaptive Partitioning
Content Generation Module Based on Independence Patches
Content Replication Module Based on Repetitive Patches
De-artifact Reconstruction Module
Repaired Video

Fig. 7. The overall frame of proposed video error concealment.

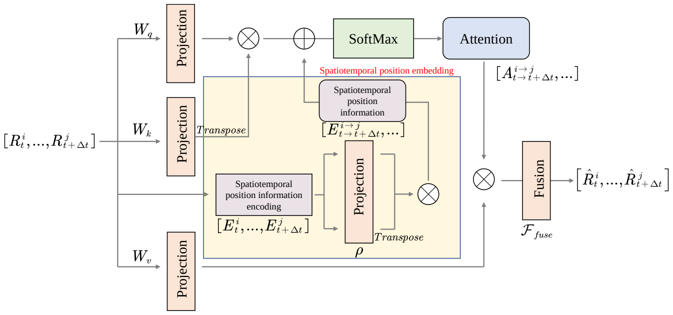

$W_q$
$W_k$
$W_v$
Projection
Projection
Projection
SoftMax
Attention
Spatiotemporal position embedding
Spatiotemporal position information
$[E_{t\to t+\Delta t}^{i\to j},\ldots]$
Transpose
$[R_t^i,\ldots,R_{t+\Delta t}^j]$
Spatiotemporal position information encoding
$[E_t^i,\ldots,E_{t+\Delta t}^j]$
Projection
$\rho$
Transpose
$[A_{t\to t+\Delta t}^{i\to j},\ldots]$
Fusion
$\mathcal{F}_{fuse}$
$[\hat{R}_t^i,\ldots,\hat{R}_{t+\Delta t}^j]$

Fig. 8. Content Replication Module Structure for Duplicate Patches.

where $N$ represents the number of patches in a frame, and $T$ represents the length of the reference video. $C_t^i$ denotes the $i$-th patch in the $t$-th frame. $C_{t+\Delta t}^i$ denotes the $i$-th patch in the $t+\Delta t$-th frame. $\phi_q$, $\phi_k$ denote the transformation used to project the patch into the *Query* and *Key* feature space, respectively.

By setting the threshold hyperparameter $\sigma$, the first $\sigma \cdot N$ patches with the highest similarity scores in Eq 1 is divided into repetitive patches $\{R_t^i = S_t^i \in top - \sigma \cdot N\}$. Since they have high similarity to most patches in the video sequence, we design a content replication network module to copy corresponding content from similar patches for repairing. The post $(1-\sigma)\cdot N$ patches in the ranking are divided into the independent patch $\{U_t^i = S_t^i \in top - \sigma \cdot N\}$. Because the content they contain is relatively independent, we designed a content generation module to fully mine the information inside these patches and generate reasonable content to fill in the missing areas. In our experiments, the threshold hyperparameter was empirically set to $\sigma = 0.9$.

### 4.3 Content Replication

The content contained in the repetitive patch has a high similarity with other patches and can be repaired by copying and pasting. Therefore, we design a content replication module based on repetitive patches, which focuses on searching similar content from patches and copying corresponding content to damaged regions. The specific structure of this module is shown in Fig 8. Previous work has explored aligning and fusing different features through the self-attention mechanism [11]. Inspired by these works, we use the self-attention mechanism to search similar content in repetitive patches. We use the following formula to convert patch similarity to attention value:

$$A_{t\to t+\Delta t}^{i\to j} = SoftMax(\frac{(W_q R_t^u)\cdot(W_k R_{t+\Delta t}^j)^T}{\sqrt{D}}), \quad (2)$$

where the attention value $A_{t\to t+\Delta t}^{i\to j}$ represents the similarity of the current patch $R_t^i$ relative to other patches $R_{t+l}^i$. $SoftMax$ represents the mapping function used to map the similarity to [0,1]. $W_q$ and $W_k$ represent the projection layer weights used to project the patch into the query space and key space, respectively, and $D$ represents the size of the embedded feature dimension of the patch. Since the attention value $A_{t\to t+\Delta t}^{i\to j}$ depends on the similarity between patches, it is used to guide copying content from other patches into the current patch.

Since different patches originate from different spatial locations in different time, it is inefficient to directly perform patch similarity matching on them. Obviously, patches with close spatio-temporal distances are more likely to have similar meaningful reproducible content. Therefore, on the basis of the above formula, we introduce position-aware embedding to enhance the ability of self-attention mechanism to capture spatio-temporal position information. Specifically, the spatio-temporal location features $R_t^i$ corresponding to $E_t^i$ are encoded from different spatio-temporal dimensions as:

$$E_t^i(R_t^i, d) = \begin{cases} \sin(W_k t), & \text{if } d = 2k \ and \ d \le \dfrac{d}{2} \\ \cos(W_k t), & \text{if } d = 2k+1 \ and \ d \le \dfrac{d}{2} \\ \sin(W_i t), & \text{if } d = 2k \ and \ d\dfrac{d}{2} \\ \cos(W_i t), & \text{if } d = 2k+1 \ and \ d \le \dfrac{d}{2} \end{cases} \quad (3)$$

where $w = 1/(10000^{2k/D})$, the feature dimension $D$ is equally divided into two parts to represent the temporal and

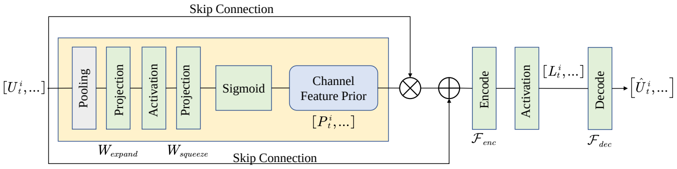


Fig. 9. Content Generation Module Structure for Independence Patches.

spatial location of the patch, $d \in [1, D]$ represents a specific dimension in the embedding feature space.

The relative spatio-temporal position embedding information $E_{t \to t+\Delta t}^{i \to j}$ can be expressed as the feature similarity distance between the current patch spatio-temporal position feature $E_t^i$ and other patch spatio-temporal position features $E_{t+\Delta t}^i$:

$$E_{t \to t+\Delta t}^{i \to j} = \rho(E_t^i)\rho(E_{t+\Delta t}^i)^T, \tag{4}$$

where $\rho$ represents the projection layer used to project spatio-temporal location features into the feature space for computing similarity distances. Noting that the relative position embedding information obtained by the above method $E_{t \to t+\Delta t}^{i \to j}$ can represent the spatio-temporal position correlation between different patches, which is further added to the attention calculation to guide the fusion of patch information.

Adding space-time relative embedding information $E_{t \to t+\Delta t}^{i \to j}$, the formula for calculating the attention value can be rewritten as:

$$A_{t \to t+\Delta t}^{i \to j} = SoftMax(\frac{(W_q R_t^u) \cdot (W_k R_{t+\Delta t}^j)^T + E_{t \to t+\Delta t}^{i \to j}}{\sqrt{D}}). \tag{5}$$

By adding position information, the adopted self-attention mechanism is more sensitive to patches with different spatio-temporal position distances, which can better capture the spatio-temporal relationship in video sequences, and is beneficial to mine spatio-temporal related information from long-distance video sequences. Using attention $A_{t \to t+\Delta t}^{i \to j}$ with position-aware embedding, the process of copying the contents of other patches $R_{t+\Delta t}^j$ to the current patch $R_t^j$ can be expressed as:

$$\hat{R}_t^i = \mathcal{F}_{fuse}(\sum_{\Delta t = -\frac{T-1}{2}}^{\frac{T-1}{2}} \sum_{j=1}^{N} A_{t \to t+\Delta t}^{i \to j}(W_V R_{t+\Delta t}^j)) \tag{6}$$

where $\hat{R}_t^i$ is the patch obtained by copying, $W_V$ represents the weight of the network layer used to learn the patch content, $\mathcal{F}_{fuse}$ represents the fusion layer used to fuse multiple patch information, which consists of two convolutional layers and *ReLU* activation functions. $N$ indicates the number of patches in each frame, $T$ indicates the length of the reference video, $R_t^j$ indicates the $i$-th patch in the $t$-th frame, and $R_{t+\Delta t}^j$ indicates the $j$th patch in the $t + \Delta t$th frame. During the repairing process, we stack multiple modules as described above to continuously benefit from the video spatio-temporal sequences.

### 4.4 Content Generation

The content contained in the independent patch is relatively independent, and it is difficult to obtain reference content from other patches for repair. Therefore, we designed a content generation module based on independent patches. This module focuses on mining the prior feature information inside the patch to synthesize visually coherent and reasonable content on the damaged area. The specific structure of the module is shown in Fig 9. We notice that different channel feature learning expresses different levels of features, such as low-frequency background and high-frequency details. Therefore, we first mine the relevant information between different channel features as prior information $P_t^i$, which can guide the synthesis of coherent and reasonable content details. Specifically, we explore and fuse highly correlated channel features by calculating the correlation coefficients between features at different levels to obtain richer feature representations. The design helps the network to better capture the correlation between channel features, thereby promoting the authenticity and coherence of subsequent synthetic image content. The process of obtaining $P_t^i$ can be expressed by the following formula:

$$P_t^i = \xi(W_{expand}(ReLU(W_{squeeze}(\mathcal{P}(U_i^t))))), \tag{7}$$

where $\xi$ represents the *Sigmoid* function, and $W_{expand}$ and $W_{squeeze}$ represent the fully connected layer weights for dimension expansion and reduction, respectively. $\mathcal{P}$ represents the global pooling and average pooling operations used to extract the features of the channel.

To generate realistic content, we employ an auto-encoder to build the network. For the encoder part, we use the correlation prior information $P_t^i$ associated with channel features to mine the feature correlation between different levels and then guide feature encoding. Conditioned on the channel-dependent prior $P_t^i$, the image patch $U_t^i$ is encoded by the encoding layer into a compact feature representation $L_t^i$:

$$L_t^i = \mathcal{F}_{enc}(U_t^i, P_t^i) = ReLU(W_{enc}((1 + P_t^i)U_t^i)), \tag{8}$$

where $\mathcal{F}_{enc}$ denotes the encoding layer for conditional encoding, and $W_{enc}$ denotes the parameter weights corresponding to $\mathcal{F}_{dec}$.

Through the above mapping, the content information of the image patch is compressed into a simplified compact feature $L_t^i$. By feeding $L_t^i$ into the decoder, the full image patch $\hat{U}_t^i$ is reconstructed.

$$\hat{U}_t^i = \mathcal{F}_{dec}(L_t^i) = W_{dec}L_t^i, \tag{9}$$

where $\mathcal{F}_{dec}$ denotes the decoding layer used for reconstruction, and $W_{dec}$ denotes the network weights corresponding to $\mathcal{F}_{dec}$. Through supervised learning, the reconstructed $\hat{U}_t^i$ is as consistent as possible with the real image patch semantics. On this basis, we utilize a discriminator network to guarantee reasonable realistic details in $\hat{U}_t^i$. In order to ensure the quality of patch content generation, we stack multiple above-mentioned modules to deepen the depth of the network, and ensure that each image patch $U_t^i$ generates semantically reasonable real content in multi-step repair.

### 4.5 Loss Function

Based on the above network design ideas, we use different loss functions for segment optimization for different purposes. First, for all modules, the network must ensure that content copied from other patches or independently generated content is faithful to the real image content. We assume the input video is $X_t$ and the corresponding damaged region mask is $M_t$, and the repaired video is $\hat{X}_t$. In order to ensure the restoration of the content, we minimize the *L1* distance between the repaired image $\hat{X}_t$ and the real image $X_t$ as the reconstruction loss function $\mathcal{L}_{rec}$ for supervised learning:

$$\mathcal{L}_{rec} = \|\hat{X}_t - X_t\|_2 \tag{10}$$

For the content replication module based on repetitive patches and the content generation module based on independent patches, the training objectives are different. For the content replication module, the optimization of the network focuses on ensuring that the content copied from other patches is faithful to the real image content and semantically coherent within a single video frame. To this end, we additionally introduce a perceptual loss to ensure the visual perceptual similarity between the repaired video frame and the original video frame. Specifically, the perceptual loss $\mathcal{L}_{per}$ measures the visual perceptual similarity by computing the similarity between the deep features extracted by the pre-trained network:

$$\mathcal{L}_{per} = \mathbb{E}\sum \|\varphi(X_t) - \varphi(X_t)\|, \tag{11}$$

where $\varphi$ represents the feature map extracted from the feature extraction network, and the feature extraction network is the *VGG-19* network pre-trained on the *ImageNet* dataset.

In summary, the loss function of the content replication module is:

$$\mathcal{L}_R = \mathcal{L}_{rec} + \alpha_R \times \mathcal{L}_{per}, \tag{12}$$

where $\alpha_R$ is a hyperparameter for balancing the loss function, which is empirically set to $\alpha_R = 0.01$. During training, we first assume that all patches are repetitive patches, that is, set the hyperparameter to $\sigma = 0.01$, and use $\mathcal{L}_R$ as the optimization target to train for 100,000 iterations.

For the content generation module, the optimization of the network focuses on ensuring that the content generated by mining the internal information of the patch is semantically reasonable and realistic in a single video frame. To this end, we additionally introduce an adversarial generation loss to ensure that the generated content is realistic and reasonable enough to fool the discriminator network. Specifically, we construct a discriminator network $D$ to assist the training of the whole network following the settings of previous papers. The discriminator $D$ takes the repaired image $\hat{X}_t$ and the real image $X_t$ as input and outputs a binary classification result (0 indicates that $\hat{X}_t$ is a synthetic image, while 1 indicates that $X_t$ is a real image). The training goal of $D$ is to distinguish the synthetic image $\hat{X}_t$ from the input as much as possible. The loss function is:

$$\mathcal{L}_D = \mathbb{E}[logD(X_t)] + \mathbb{E}[log(1 - D(\hat{X}_t))]. \tag{13}$$

For the restore network, it learns to generate indistinguishable realistic details through an adversarial loss, which is calculated as:

$$\mathcal{L}_{adv} = \mathbb{E}[logD(\hat{X}_t)]. \tag{14}$$

To sum up, the loss function of the content generation module is:

$$\mathcal{L}_U = \mathcal{L}_{rec} + \alpha_U \times \mathcal{L}_{adv} \tag{15}$$

where $\alpha_U$ is a hyperparameter for balancing the loss function, which is empirically set to $\alpha_U = 0.01$. Based on the training of the content replication module, all patches are assumed to be independent patches (*i.e.* $\sigma = 0$), and trained 100,000 iterations with $\mathcal{L}_U$ as the optimization target.

Finally, we train all modules jointly by setting $\sigma = 0.9$. The discriminator $D$ is extended to the temporal domain, that is, the discriminator $D$ takes the repaired video $[\hat{X}_{t-l}, \cdots, \hat{X}_{t+l}]$ and the real video $[X_{t-l}, \cdots, X_{t+l}]$ as input to ensure that the spatio-temporal artifacts located at the patch boundaries are eventually eliminated. The final joint training objective can be expressed as:

$$\mathcal{L} = \mathcal{L}_{rec} + \alpha_{per} \times \mathcal{L}_{per} + \alpha_{adv} \times \mathcal{L}_{adv} \tag{16}$$

where $\alpha_{per}$ and $\alpha_{adv}$ are hyperparameters for balancing the loss function, which are empirically set to $\alpha_{per} = 0.01$ and $\alpha_{adv} = 0.01$. During training, 200,000 iterations are trained with $\mathcal{L}$ as the optimization target.

## 4.6 Binary Quantization Strategy

Since the video error concealment methods are usually oriented to the application scenario of real-time video communication transmission, it has high requirements for space memory resources, computational complexity, and reasoning energy cost. In order to meet its real-time application requirements on edge devices, we adopted our binary quantization strategy [12] to compress our VEC model in order to reduce the deployment cost of the model as much as possible while maintaining the visual perception quality. The specific quantization compression method adopted is as follows:

- For all modules in the proposed video error concealment network, binary quantization and compression are performed on the parameters that need to be stored. In this way, the proposed network model can achieve close to the theoretical highest compression rate of 32×, which greatly reduces the memory space cost of the actual deployment of the model.
- The proposed model is aimed at video sequences, where a large amount of reasoning calculations are used for the interaction of spatio-temporal video information. This process is realized by self-attention calculations, which consumes a lot of computational cost. Therefore, we use the binary attention mechanism proposed in [13] to implement this process to effectively improve the efficiency of computational reasoning. To guarantee the visual quality of videos, we employ the proposed attention model based on binary activations.

Note that the above-mentioned quantization compression method can bring about computational simplification, and the inference calculation of the model can be performed almost without multiplication. This quantitative compression strategy can also reduce the computational complexity of the model and control the energy consumption cost of inference. The specific details will be analyzed in detail in Section 5.

# 5 Experimental Results

To verify the effectiveness of the proposed efficient video error concealment method based on binary quantization compression, we conduct experiments on the constructed video error concealment database, the details of which have been introduced in Section 3. Next, we introduce the evaluation indicators, comparison methods, and specific implementation details used in the experiment.

## 5.1 Evaluation Protocol

### 5.1.1 *Evaluation Metrics*

We use three indicators to evaluate the operating efficiency of the model, including model size, calculation amount, and inference energy consumption. For video quality evaluation, four widely used performance evaluation metrics are adopted, namely PSNR, SSIM, LPIPS and VMAF. PSNR is used to measure the pixel difference between the reconstructed video frame and the real video frame, SSIM reflects the structural similarity between the reconstructed video frame and the real video frame, and LPIPS learns the perceptual similarity between the image blocks of the video frame. Among them, PSNR and SSIM can be obtained through their definition calculations, and LPIPS can be obtained by running the corresponding model. Since the above metrics are for individual video frames, for each video we calculate its score frame-by-frame and report the average for the entire video. However, the above indicators can hardly reflect the perceived visual quality of the human visual system. Therefore, we introduced VMAF, which combines visual information fidelity, detail loss indicators, and temporal motion information from the perspective of the entire video to measure perceptual quality. For VMAF, we uniformly convert repaired videos into the *MP4* format. For each video, we perform frame-by-frame calculations and report their average values.

### 5.1.2 *Comparison Method*

To verify the performance of the proposed method, we compare the method with four representative video inpainting methods. STTN [7] is the first to use the *Transformer* architecture to mine the spatio-temporal sequence information in the video for inpainting. DSTT [14] decomposes the execution of the attention mechanism in the spatial and temporal domains on the basis of STTN [7]. FuseFormer [15] performs overlapping segmentation on the video sequence to ensure that the network notices all the details, while E2FGVI [16] introduces the optical flow information in the video sequence to explicitly guide video restoration. All these state-of-the-art video inpainting methods are used to repair the same video, and the results of all methods are obtained based on the code open-sourced by the authors of the paper.

### 5.1.3 *Implement Detail*

All experiments are performed on a Linux server platform equipped with *NVIDIA A100-PCIE-40GB*. During training, we use the *Adam* optimizer to optimize the network weights. In each iteration, 3 random frames from the same video are sampled. As described in Section 4, we train the network model in stages. For the content generation module and the content replication module, the learning rate is set as 0.001 and iteratively trained 100,000 times. On the basis of the former, the learning rate is set to 0.001 and all modules are jointly iteratively trained 2,000,000 times, and the learning rate is attenuated by one-tenth after the 150,000th iteration. For other models, we follow the original training settings of their papers, and obtain the best-performing models.

TABLE 1
COMPARISON OF MODEL SIZE, COMPUTATION AND INFERENCE ENERGY CONSUMPTION OF FIVE DIFFERENT METHODS.

| Method | Model Size | Calculations | Energy |
|---|---|---|---|
| STTN [7] | 135.58MB | 432.28G | 994.24G pJ |
| DSTT [14] | 131.72MB | 432.22G | 994.10G pJ |
| FuseFormer [15] | 135.58MB | 509.43G | 1171.69G pJ |
| E2FGVI [16] | 156.85MB | 509.89G | 1172.75G pJ |
| Proposed | 7.88MB | 284.23G | 252.29G pJ |

TABLE 2
COMPARISON OF RECONSTRUCTED VIDEO QUALITY OF FIVE DIFFERENT METHODS ON THE PROPOSED DATABASE.

| Method | PSNR↑ | SSIM↑ | LPIPS↓ | VMAF↑ |
|---|---|---|---|---|
| STTN [7] | 32.5281 | 0.9558 | 0.0263 | 86.692 |
| DSTT [14] | 32.7921 | 0.9574 | 0.0254 | 86.867 |
| FuseFormer [15] | 33.0360 | 0.9588 | 0.0246 | 87.264 |
| E2FGVI [16] | 33.1355 | 0.9593 | 0.0242 | 87.193 |
| Proposed | 32.6280 | 0.9551 | 0.0268 | 86.708 |

## 5.2 Analysis on Inference Efficiency

In this section, we compare the proposed efficient video error concealment method based on binary quantization with existing methods in terms of model deployment and operation efficiency. As a reference, we provide the experimental results of four state-of-the-art methods and our method on the proposed database in Table 5.2. Due to the use of binary quantization compression technology, the model size of the proposed method is only about 5.63% of the average value of the other methods, which only needs 7.88MB. Through binary quantization and compression, a large number of multiplication operations in the model inference calculation process are converted into bit operations. Due to the low cost of bit operations, the computation of the proposed method is only about 60.35% of the average value of other methods. Moreover, since the energy cost of bit operations is almost negligible, the energy cost of model inference is only about 23.29% of the average value of other methods. The experimental results prove that the proposed method has certain advantages in practical deployment, including small storage space requirement, low computational complexity and low inference energy cost, which are crucial in the practical deployment and application process of video error concealment.

## 5.3 Analysis on Recontruction Quality

We compare the proposed method with existing methods and verify that the proposed method can guarantee the reconstructed video quality while operating more efficiently. In this experiment, the packet loss rate is fixed at 5%, and the mask type is randomly selected as one of Random, Slice, and Tile. The experimental results of the four

TABLE 3
COMPARISON OF RECONSTRUCTED VIDEO QUALITY OF FIVE DIFFERENT METHODS ON THE PROPOSED DATABASE.

| Method | PSNR(dB)↑ | SSIM↑ | LPIPS↓ | VMAF↑ | Model Size(MB) | Calculations(G) | Energy(G pJ) |
|---|---|---|---|---|---|---|---|
| STTN [7] | 32.5281 | 0.9558 | 0.0263 | 86.692 | 135.58 | 432.28 | 994.24 |
| DSTT [14] | 32.7921 | 0.9574 | 0.0254 | 86.867 | 131.72 | 432.22 | 994.10 |
| FuseFormer [15] | 33.0360 | 0.9588 | 0.0246 | 87.264 | 135.58 | 509.43 | 1171.69 |
| E2FGVI [16] | 33.1355 | 0.9593 | 0.0242 | 87.193 | 156.85 | 509.89 | 1172.75 |
| Proposed | 32.6280 | 0.9551 | 0.0268 | 86.708 | 7.88 | 284.23 | 252.29 |

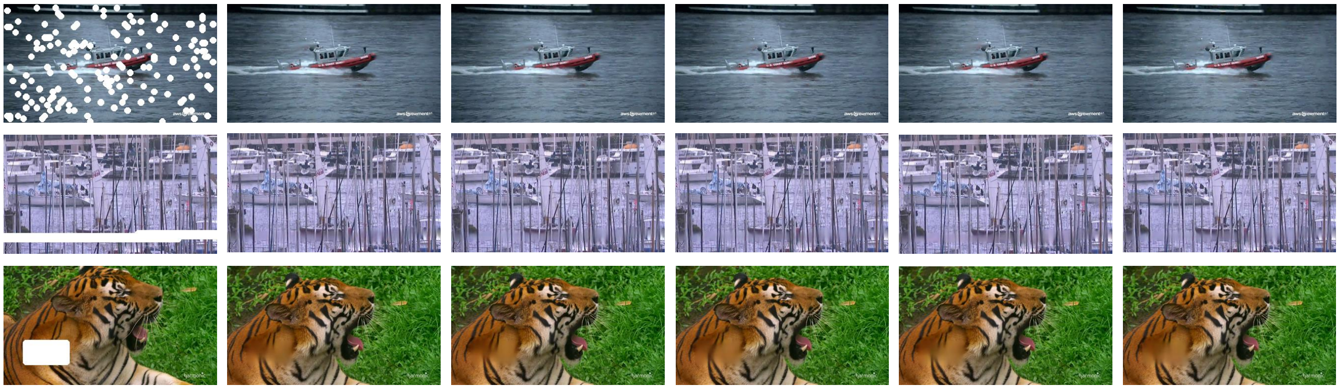

(a) Corrupted Frame (b) STTN (c) DSTT (d) FuseFormer (e) E2FGVI (f) Proposed

Fig. 10. Illustration of video error concealment results for five different methods.

most advanced methods and the proposed methods on the proposed database are provided in Table 2, including several commonly used video reconstruction quality indicators PSNR, SSIM, LPIPS and VMAF. The proposed method has obvious advantages in practical deployment, and the quality of its reconstructed video is close to that of the existing methods without an obvious visual quality gap. Compared with STTN, the PSNR and VMAF achieved by the proposed method are improved by about 0.0999 and 0.99, respectively, while the SSIM and LPIPS are basically the same. Compared with other methods, the proposed method controls the gap of PSNR within about 0.5, and the average gaps of SSIM, LPIPS and VMAF are 0.0034, 0.0021 and 0.296 respectively, which proves that the proposed method can While having obvious deployment advantages.

Since VMAF characterizes the visual perceptual quality, which is more consistent with human eye perception, we further analyzed the VMAF results for selected video sequences belonging to different content categories in the database. Detailed results are provided in Table 2, and the results reported for each sequence are the average of multiple videos with the same content and different compression distortions. The VMAF test model adopted is the HD model, which takes into account the screen size, resolution, and distance between the viewer and the display device. From the experimental results, it can be seen that for video sequences with different contents, the proposed method can complete high-quality inpainting and achieve comparable visual quality with other video inpainting methods. Compared with STTN, the proposed method achieves better restoration quality in multiple video sequences covering natural, indoor, outdoor, and virtual content scenes such as Animals, BarScene, Mobcal, Station, Minecraft and Witcher, And it is 0.016 higher than the average VMAF result for all sequences. Compared to DSTT, FuseFormer and E2FGVI, the proposed method achieves slightly lower VMAF on most video sequences, but still achieves an average of 86.708, proving that the proposed method can still achieve comparable visual quality.

In addition to the above objective reconstruction quality metrics, a visual comparison of the corresponding video error concealment results is provided in Fig 10. In the error concealment results, the first column shows the content error caused by random data loss. It can be seen that the proposed method well recovers the missing part of the lake, and the windows on the yacht, the visual perception is close to the latest method. In the error concealment results in the second row, we show consecutive examples of slice loss, which is caused by the video data packetization transmission. The proposed method fills in missing regions with nearby spatio-temporal content, such as boat poles and running water, ensuring the structural integrity of video frames. Although the details of the repaired boat pole material are slightly inferior to other methods, it is still guaranteed that the repaired content is semantically correct. In the error concealment results in the third row, stacked diced data loss samples are provided, which impose higher requirements on error concealment recovery due to large areas of continuous content loss. Like other state-of-the-art methods, the proposed method recovers semantically correct fur, demonstrating that the proposed method can correctly reconstruct semantically identical content and provide good visual perceptual quality.

### 5.4 Analysis on Robustness

In order to further verify the effectiveness of the proposed method in actual application scenarios, we analyze the performance in different scenarios through different experi-

TABLE 4
COMPARISON OF RECONSTRUCTED VIDEO VMAF RESULTS FOR PARTIAL VIDEO SEQUENCES IN THE PROPOSED DATABASE.

| Sequence | Type | STTN [7] | DSTT [14] | FuseFormer [15] | E2FGVI [16] | Proposed |
|---|---|---|---|---|---|---|
| Animals | Nature | 85.507 | 85.731 | 86.019 | 86.014 | 85.838 |
| Blackfish | Nature | 85.828 | 85.934 | 86.466 | 86.411 | 85.792 |
| Costa | Nature | 85.123 | 85.483 | 85.808 | 85.628 | 85.045 |
| BarScene | Outdoor | 88.140 | 88.451 | 88.794 | 88.682 | 88.150 |
| Johnny | Outdoor | 85.990 | 86.074 | 86.673 | 86.584 | 85.914 |
| Mobcal | Outdoor | 86.519 | 86.633 | 87.066 | 87.028 | 86.533 |
| Coastguard | Indoor | 88.214 | 88.582 | 88.917 | 88.789 | 88.140 |
| Harbour | Indoor | 86.039 | 86.043 | 86.541 | 86.430 | 85.931 |
| Station | Indoor | 87.309 | 87.454 | 87.813 | 87.761 | 87.345 |
| Csgo | Virtual | 85.814 | 86.091 | 86.473 | 86.322 | 85.811 |
| Minecraft | Virtual | 86.740 | 86.851 | 87.285 | 87.241 | 86.747 |
| Witcher | Virtual | 87.653 | 87.864 | 88.156 | 88.118 | 87.751 |
| Total | - | 86.692 | 86.867 | 87.264 | 87.193 | 86.708 |

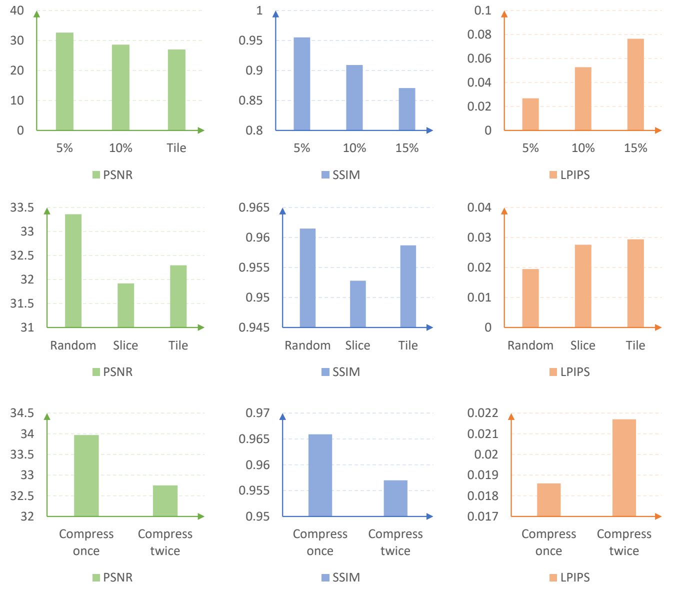


Fig. 11. Robustness experimental results in different scenarios. The top line is the result of different packet loss rates, the middle line is the result of different packet loss scenarios, and the lowest line is the result of different compression distortion scenarios

ments to analyze and verify the robustness of the proposed method, including different packet loss rates, packet loss scenarios, compression distortion, and video frame switching.

**Packet Loss Rate**: The transfer process of video streaming data is often complex and congested. It may include multiple transmissions between different networks, such as content delivery networks and Internet service providers, and different network segments such as wired, wireless, and WiFi, which may cause fluctuations in connection quality. Considering that the packet loss rate is about 1%-5% during video communication transmission, we set the packet loss rate extremes as 5%, 10%, and 15% to generate mask types and use the proposed method for error concealment. Experimental results are provided in Fig, which prove that the proposed method maintains considerable objective visual quality under different packet loss rates and has high robustness.

**Packet Loss Scenario**: The packet loss errors that may occur during the transmission of video stream data are different, and it may include the simulated content loss in the dataset, including Random, Slice and Tile. The comparison results under different packet loss scenarios are provided in Fig 11. It can be seen that for the random type of loss, the proposed method is able to copy the nearby spatio-temporal related content to the loss area due to the small loss area. In the area of repair, it reflects a good video repair effect in many indicators. Compared with Random, Slice and Tile produce larger continuous area content loss, have higher restoration difficulty, and the quality of the reconstructed video is slightly reduced. But the proposed method can still cope well with these complex packet loss error scenarios and stably reconstruct videos with high visual quality.

**Compression Distortion**: Video stream data is compressed using different compression tools on different network platforms or client terminals. They use different video compression standards and have different compression distortion. The proposed video dataset contains distortions generated by various video compression standards. We use the proposed video error concealment method to repair videos with primary compression distortion and secondary compression distortion respectively. The results are shown in Fig 11. Experimental results show that the proposed video error concealment method exhibits good robustness in the face of different compression artifacts. The proposed method still maintains the high visual quality of the video regardless of the artifacts generated by primary compression or secondary compression.

**Frame Switching**: Videos usually contain a large amount of content information, and there is scene switching, which requires video error concealment to ensure the stability of reconstructed video quality when repairing different video frames. Therefore, we randomly sample video sequences in the dataset and provide the inpainting results of each video frame in Fig 12. From the experimental results, it can be seen that the proposed method can still maintain the video inpainting quality near a high-quality level and provide high perceptual quality when different video frames change. Experimental results show that the proposed video error concealment method is robust to video frame switching, and can adapt to video content switching in practical application scenarios. Nevertheless, with the switching of frame scenes, there are still failed repair cases in the repair results shown. For example, in the 80th frame of the video, compared

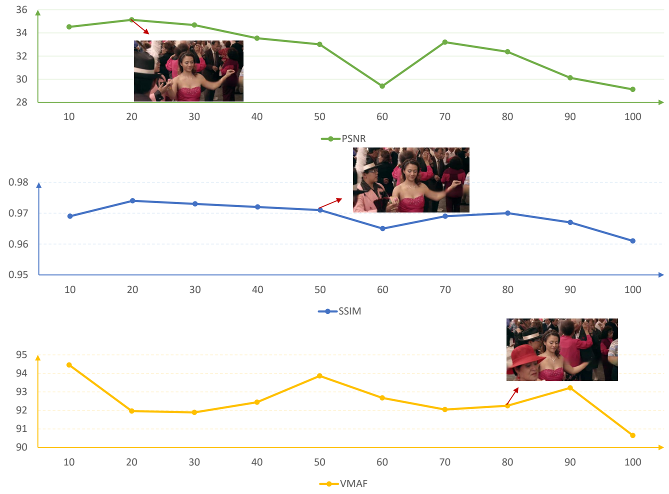


Fig. 12. Robustness experimental results on frame switch.

with the 20th frame and the 50th frame, due to the lack of effective palm reference information, the visual effect of restoring the palm is not satisfactory, which still needs to be explored for improvement in the follow-up research work.

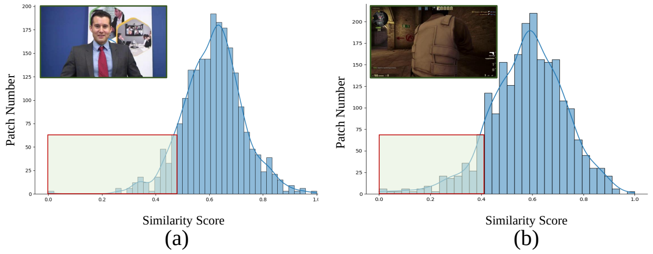
’

### 5.5 Analysis on Patch Adaptive Partition

In order to deeply analyze the operation mechanism of the proposed method, the visualization results of image patch adaptive division are provided in Fig 13. The figure shown is the similarity distribution of all patches, the upper left corner is the corresponding video frame, and the red box represents patches identified by the network as relatively independent in content. The importance distribution curve of patches is close to a normal distribution, and most of the patches near the right side of the curve have higher similarity, which contains the spatio-temporal related information required for video error concealment. Therefore, it is more efficient to utilize replication modules for content restoration between these patches. The red boxed part on the left side of the curve indicates the independent patches that are considered less relevant by the network. These patches are repaired through the generation module without engaging in global information interactions, avoiding unnecessary information references. This process improves the efficiency of repairing using the attention mechanism, and therefore can effectively improve the quality of repair The efficiency of restoration can effectively improve the quality of restoration. The presented visualization results and analysis illustrate the inner working mechanism of the proposed video error concealment method.

### 5.6 Ablation Experiments

To further analyze the effectiveness of the proposed method, we conduct ablation experiments on different modules and training strategies for inpainting repetitive patches and independent patches. The detailed results of ablation experiments are provided in Table 5. It can be seen that when only the content replication module and the content generation module are used, the reconstructed video is similar in many objective visual indicators, and the video restoration quality is low. Therefore, we further jointly build the network with two modules. However, when direct training is adoped, the inpainting quality is worse than using a single module. To solve the above problems, we use the proposed loss function combination and distribution training strategy for further training, and the reconstructed video achieves higher objective visual quality than that of a single module, proving the effectiveness of the proposed method in improving the quality of video inpainting by dividing patches. Effectiveness verifies the rationality of the designed training strategy.



(a)

(b)

Fig. 13. Illustration of patch adaptive division results. (a) is the partition result of *Johnny*, (b) is the partition result of *CSGO*.

TABLE 5
COMPARISON OF RECONSTRUCTED VIDEO QUALITY OF FIVE DIFFERENT METHODS ON THE PROPOSED DATABASE.

| Structure | Strategy | PSNR↑ | SSIM↑ | LPIPS↓ | VMAF↑ |
|---|---|---|---|---|---|
| Replication | Direct | 32.5421 | 0.9546 | 0.0273 | 86.470 |
| Generation | Direct | 32.5581 | 0.9544 | 0.0274 | 86.584 |
| Joint | Direct | 32.4932 | 0.9541 | 0.0277 | 86.584 |
| Joint | Step | 32.6280 | 0.9551 | 0.0268 | 86.708 |

## 6 Conclusion

In this paper, we propose a high-efficiency video error concealment technology based on binary quantization compression, which further extends the binary quantization compression technology to video processing models for pixel-level vision tasks, which can effectively reduce the video error concealment technology. The actual deployment cost, and verified the feasibility of combining binary quantization compression technology with video error concealment. To promote the research of deep video error concealment, we first construct a video error concealment-oriented database, which fully considers the application scenarios of video communication errors. Based on this, we propose a new video error concealment framework, which divides video patches into repetitive patches and independent patches, and utilizes a replication module and a generation module for inpainting, respectively. Finally, we apply the binary quantization compression strategy to the video error concealment model, conduct extensive experiments on the built database, verify its robustness in various application scenarios, and prove that the proposed technique is Advantages in deployment cost and operating efficiency.

## References

[1] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learning image and video compression through spatial-temporal

energy compaction. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 10071–10080, 2019.

[2] Guo Lu, Tianxiong Zhong, Jing Geng, Qiang Hu, and Dong Xu. Learning based Multi-modality Image and Video Compression. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 6083–6092, 2022.

[3] Xihua Sheng, Jiahao Li, Bin Li, Li Li, Dong Liu, and Yan Lu. Temporal context mining for learned video compression. *IEEE Transactions on Multimedia*, 1(1):1–12, 2022.

[4] Mehmood Nawaz and Hong Yan. Saliency Detection using Deep Features and Affinity-based Robust Background Subtraction. *IEEE Transactions on Multimedia*, 1(1):1–8, 2020.

[5] Xinfeng Zhang, Ruiqin Xiong, Weisi Lin, Siwei Ma, Jiaying Liu, and Wen Gao. Video compression artifact reduction via spatio-temporal multi-hypothesis prediction. *IEEE Transactions on Image Processing*, 24(12):6048–6061, 2015.

[6] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-LVC: multiple frames prediction for learned video compression. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3546–3554, 2020.

[7] Yanhong Zeng, Jianlong Fu, and Hongyang Chao. Learning joint spatial-temporal transformations for video inpainting. In *Springer European Conference on Computer Vision (ECCV)*, pages 528–543, 2020.

[8] Stefan Winkler. Analysis of public image and video databases for quality assessment. *IEEE Journal of Selected Topics in Signal Processing*, 6(6):616–625, 2012.

[9] Somdyuti Paul, Andrey Norkin, and Alan C Bovik. Speeding up VP9 intra encoder with hierarchical deep learning-based partition prediction. *IEEE Transactions on Image Processing*, 29(1):8134–8148, 2020.

[10] Miaohui Wang, Wuyuan Xie, Xiandong Meng, Huanqaing Zeng, and King Ngi Ngan. UHD video coding: A light-weight learning-based fast super-block approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 1(1):3083–3094, 2019.

[11] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in Transformer. In *Advances in Neural Information Processing Systems (NIPS)*, pages 15908–15919, 2021.

[12] Miaohui Wang, Zhuowei Xu, Bin Zheng, and Wuyuan Xie. Binaryformer: A hierarchical-adaptive binary vision transformer (vit) for efficient computing. *IEEE Transactions on Industrial Informatics*, 2024.

[13] Miaohui Wang and Zheng Bin. A Hierarchical-Adaptive Binary Vision Transformer (ViT) for Efficient Computing. *IEEE Transactions on Industrial Informatics*, 1(1):1–11, 2023.

[14] Liu, Rui and Deng, Hanming and Huang, Yangyi and Shi, Xiaoyu and Lu, Lewei and Sun, Wenxiu and Wang, Xiaogang and Dai, Jifeng and Li, Hongsheng. Decoupled spatial-temporal transformer for video inpainting. *arXiv preprint arXiv:2104.06637*, pages 1–10, 2021.

[15] Rui Liu, Hanming Deng, Yangyi Huang, Xiaoyu Shi, Lewei Lu, Wenxiu Sun, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fuseformer: Fusing fine-grained information in transformers for video inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14040–14049, 2021.

[16] Zhen Li, Cheng-Ze Lu, Jianhua Qin, Chun-Le Guo, and Ming-Ming Cheng. Towards an end-to-end framework for flow-guided video inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17562–17571, 2022.