Enhanced Asymmetric Invertible Network for Neural Video Delivery

Wenbin Tian[0009−0004−2295−7986], Qingmiao Jiang[0000−0001−6078−6736], Lu Chen[0009−0003−7929−9454], Haolin Li[0009−0009−1855−6553], and Jinyao Yan[0000−0003−4153−313X]

Communication University of China, China
tianwenbin@cuc.edu.cn

Abstract. Internet video streaming has experienced explosive growth over the past few years. Recently, super-resolution (SR) networks have been utilized to reduce the bandwidth and improve the quality of Internet video streaming. These methods first employ a predefined and immutable downscaling kernel, such as bicubic interpolation, to transform high-resolution (HR) video into low-resolution (LR) video. Subsequently, the LR video is partitioned into segments, which are streamed alongside corresponding models to the clients. The client subsequently executes inference models to perform SR on the LR segments. However, this normal downscaling is not an injective mapping because high-frequency information is lost. This creates the ill-posed problem of the inverse upscaling procedure and makes it highly difficult to get details back from downscaled LR videos. In this paper, we propose a novel method for video delivery. Specifically, we deliberately designed an Enhanced Asymmetric Invertible Network (EAIN) to produce high-quality LR videos while capturing the distribution of missing information using a latent variable that follows a specified distribution in the downscaling process. HR videos are available by passing a randomly extracted latent variable through the network in reverse with LR videos. Extensive experiments show that our methods significantly improve video streaming quality compared to state-of-the-art neural video delivery methods, paving the way for the application of neural video delivery techniques in practice. The code is available at https://github.com/Anonymous-ACCV-2024/EAIN.

Keywords: Neural video delivery · Invertible neural network · Video stream quality

1 Introduction

Internet video traffic has experienced tremendous growth in recent years, which brings intense pressure to the current video delivery infrastructure. Fig. 1 is a schematic diagram of the current video streaming infrastructure, which usually consists of two sides: (1) The ingest side involves the delivery of video from the original streamer to a media server. (2) At the distribution side, the server sends the prepared videos to clients through the server's downlink [5]. The video

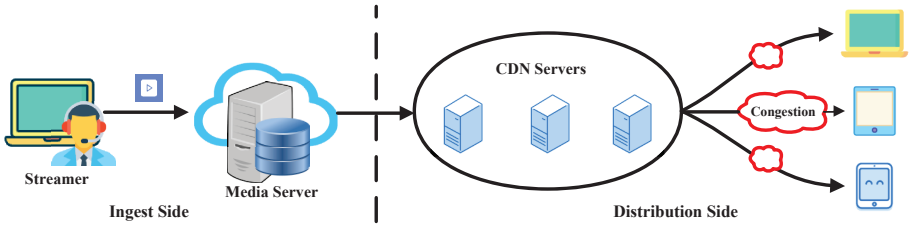

Fig. 1. Workflow for the current video streaming infrastructure.

streaming quality on the user side is primarily determined by the bandwidth between the server and client. When downstream bandwidth resources become constrained, user quality of experience (QoE) suffers from serious deterioration. Encouraged by the rapid increase in client computing power and recent developments in deep learning, lots of studies [25,7,23,26,13,10,28,9] propose to employ SR networks to reduce the downstream bandwidth of video delivery. The specific practices of these works are to first use a predetermined and immutable downscaling kernel, such as bicubic interpolation, then divide LR videos into segments and stream both LR video segments (e.g., 270p) and corresponding content-aware SR models (e.g., $\times 4$) from servers to clients. The client subsequently executes inference models to perform SR on the LR segments and get the HR videos (e.g., 1080p). In this way, HR videos can be available with limited downstream network bandwidth. This SR network-based video delivery method outperforms commercial techniques like H.264, H.265 [13], and WebRTC [7].

Although the application of SR networks to video delivery is promising, there are still some problems. One major limitation is that the traditional downscaling method (e.g., bicubic) will lose the detail of the video frame, which is extremely challenging for the SR network to recover the original HR video. The specific reason is that downscaling images is called non-injective mapping, which indicates that there could be multiple possible HR images that could result in the same downscaled LR images. Consequently, this upscaling process is commonly perceived as being ill-posed [4,21].

To alleviate this ill-posed problem, most previous works chose SR models [27] with better performance to upscale the downscaled LR video frames, which further improves the quality of the video stream. However, high-performance SR models are typically accompanied by higher computational requirements and longer training times, posing a burden on practical deployment. Moreover, these methods ignore the compatibility between the upscaling and downscaling operations. From an opposing perspective, it is extremely invaluable to take the video frame downscaling method into consideration for recovering the original HR video. In addition, rather than simply regarding the image downscaling and upscaling as two divided and unrelated processes, some works [6,11,17] attempt to represent image downscaling and upscaling as a combined task using an

encoder-decoder framework. Specifically, they introduce an auto-encoder-based framework allowing simultaneous learning of the downscaling and upscaling networks to optimize recovery performance. Although these methodologies exhibit substantial potential for improving the quality of HR images derived from their respective downscaled LR counterparts, it is not an optimal recovery. Since these endeavors did not sufficiently address the ill-posed problem, they merely connected the two processes via training objectives without capturing any feature of the missing information. Most recently, several works [20,22] adopt invertible neural network (INN) to model the processes of downscaling and upscaling. For example, IRN [20] models the rescaling of images as a bijective transformation with INN to obtain as much information about the HR images. The high-frequency residual components are integrated into a case-agnostic latent distribution for efficient restoration. [22] proposes the SAIN model, which combines rescaling and compression into a single invertible process, employing decoupled modeling. These methods have a certain effect on solving the ill-posed problem.

To further improve the quality of video streaming on the client side, in this paper, we propose a new method to mostly alleviate this ill-posed problem posed by SR networks in neural video delivery. We deliberately design an enhanced asymmetric invertible network (EAIN) to produce high-quality LR videos and while capture the distribution of missing information using a latent variable that follows an isotropic Gaussian distribution in the downscaling process. And HR videos can be available by inversely passing a randomly extracted latent variable through the network in reverse with LR videos. Specifically, to effectively alleviate the ill-posed problem, we adopt the idea of INN to model the distribution of information lost during downscaling. The first module in our INN module is the wavelet transformation, which is to break down the HR frame x into low and high-frequency components. Since the case-specific high-frequency information is lost after downscaling, we propose an enhanced invertible block (E-InvBlock) to generate the visually appealing LR frame while modeling the distribution of the missing information by introducing an additional latent variable z, which can effectively recover the HR frame in the upscaling process. E-InvBlocks are effective when modeling invertible transformations, but they usually have a limited capacity for nonlinear representation [2]. To better extract the features into the video frames, we adopt an effective feature enhancement module (EFEM) in a cascade manner before the E-InvBlock. Moreover, we utilize a more fine-grained segmentation scheme (FGSS) before feeding video frames into the network. The detailed procedure is that dividing video frames into smaller patches, which effectively improve the quality of the video stream and reduce the network training burden. The specific workflow of applying our method to neural video delivery is that the model is trained on the server side and generates LR videos using the forward process, followed by transmitting the model and LR videos to the client side, where the inverse of the model is used to restore the HR videos. Overall, our contributions are as follows:

- We propose an Enhanced Asymmetric Invertible Network (EAIN) for neural video delivery, significantly mitigating ill-posed problems caused by SR networks and achieving remarkable performance.
- We present a new Enhanced Invertible Block (E-InvBlock) in the INN module, which can effectively improve the quality of reconstructed HR video frames in the upscaling process. To improve the nonlinear representation of E-InvBlock, we propose an Effective Feature Enhancement Module (EFEM). Additionally, we adopt a more Fine-Grained Segmentation Scheme (FGSS), which further improves the quality of the video stream on the client side and reduces the network training burden.
- We conduct extensive experiments across various neural video delivery methods, video durations, and scaling factors to show the benefits of our method.

2 Related work

2.1 Neural Video Delivery

NAS [25] is a novel and pioneering Internet video delivery system that integrates SR network-based quality enhancement. It can effectively tackle the video quality deterioration issue when bandwidth resources become limited. The key idea is to exploit the overfitting property of the SR network and leverage training accuracy to achieve superior performance. Lots of the following works [7,23,26,13,10,28,9] also exploit overfitting properties to ensure high performance on the client side. [7] proposes a live video ingest framework called LiveNAS, enhancing the NAS framework by incorporating an online learning module, which further improves video quality. NEMO [23] proposes an ingenious algorithm to select keyframes in the video to SR, which can dramatically reduce the amount of client-side computation. NeuroScaler [26] introduces a unique and innovative platform that provides both efficient and scalable neural enhancement capabilities specifically designed for live streaming applications. CaFM [13] efficiently diminishes the quantity of SR models and enhances video quality by implementing a handcrafted layer with a tactful combined training framework. EMT [10] suggests employing meta-learning and selective patch sampling techniques to further decrease the number of models and computational expenses. RepCaM [28] adopts reparameterization content-aware modulation and an online video patch sampling method to speed up the training convergence. STDO [9] presents an innovative spatial-temporal data overfitting strategy aimed at enhancing both the quality and efficiency of video resolution upscaling tasks at the user end. We have done comprehensive experiments to show that our method outperforms these neural video delivery methods using SR networks.

2.2 Image Rescaling

SR endeavors to reconstruct the HR image from its pre-downscaled counterpart. Despite being a prevalent technique for image upscaling and yielding promising

outcomes in LR image upscaling tasks, SR is inherently ill-posed. For conventional methods of image downscaling, frequency-based kernels like bilinear or bicubic filters are commonly employed as low-pass filters to downscale input HR images to the desired target resolution. However, these methods produce overly smoothed LR images due to the loss of high-frequency information. [6] suggests a new task-aware image downscaling model that uses a relatively deep convolutional auto-encoder. The encoder and decoder are the models that do the downscaling and upscaling, respectively. CNN-CR [11] presents a learning method for compact resolution images using a convolutional neural network that can be trained separately or together with a CNN for SR images. CAR [17] is an comprehensive system trained to maximize SR performance that simultaneously learns a mapping to reduce resolution and improve SR performance. In recent studies [20,22] there have been proposals to incorporate INN for modeling both the downscaling and upscaling procedures. For example, IRN [20] employs a bijective INN to characterize both the downscaling and upscaling operations. High-frequency components are encapsulated within a case-agnostic latent distribution to facilitate effective reconstruction. SAIN [22] develops an end-to-end unsymmetrical framework for compression-aware image rescaling. In this paper, we propose an enhanced asymmetric invertible network (EAIN) that performs well in neural video delivery.

3 Method

3.1 Overall Framework

In this subsection, we describe the overall framework of our method (EAIN) in detail. Fig. 2 presents the outline of our modeling framework. To effectively mitigate the challenges posed by the ill-posed nature of the upscaling task, we employ the concept of INN to model the distribution of the loss information during the downscaling process. Following the previous work [20,22], we also use a wavelet transformation to separate the HR frame x into its low-frequency components (x_l) and high-frequency components (x_h) in the INN module. Due to the fact that case-specific high-frequency information is lost after downscaling, to recover the HR frame x as much as possible in the upscaling process, similar to IRN [20], we propose an enhanced invertible block (E-InvBlock) to produce the visually appealing LR frame y while modeling the distribution of missing information by incorporating an additional latent variable z. In summary, EAIN forces z to be case-agnostic and follow a straightforward, specified distribution, e.g., an isotropic Gaussian distribution. In this way, it is not necessary to save the high-frequency component and latent variable z after downscaling. During the upscaling process, z can be arbitrary sampled and utilized to reconstruct the HR frame x alongside the LR frame y using EAIN. To enhance the nonlinear representation of E-InvBlock, we add an effective feature enhancement module (EFEM) before the E-InvBlock. Moreover, we adopt a more fine-grained segmentation scheme (FGSS) before feeding video frames into the network. FGSS

means dividing video frames into smaller patches, which effectively improves the quality of the video stream and reduces the network training burden.

3.2 Invertible Neural Network

The Wavelet Transformation. To be able to efficiently acquire the ability to break down HR frame x into the downscaled frame y and case-agnostic high-frequency information embedded in z, following the previous work [20,22], we also utilize the wavelet transformation to explicitly break down the input frame into a low-frequency approximation and three sets of high-frequency coefficients in different directions [19,1]. Specifically, the wavelet transformation transforms a set of feature maps with dimensions C channels, H height, and W width into a tensor with a shape of (4C, 1/2H, 1/2W). The initial group of C slices for the output tensor is generated through average pooling, serving as an approximate low-pass representation. The remaining three sets of C slices represent residual components in the vertical, horizontal, and diagonal directions, respectively, encapsulating the high-frequency details in the original HR frame. Through the wavelet transformation, the low and high-frequency components are effectively divided and will be sent into the next module.

Efficient feature enhancement module. In this part, we suggest that our proposed efficient feature enhancement module (EFEM). E-InvBlock commonly has a limited capacity for nonlinear representation [2]. Therefore, we add an EFEM in a cascade manner before the E-InvBlock to increase the nonlinear representativeness of our network. Note that EFEM only works on the downscaling process of the EAIN. As shown in Fig. 2, the EFEM is mainly composed of two enhanced dense blocks (EDBs), three general convolutions, and an enhanced spatial attention (ESA) block [14]. ESA has been shown to enhance the model's performance effectively from the spatial perspective [14]. Let F_{in} denote the input feature; the overall process can be expressed as

$$F_0 = H_{\text{EDB1}}\left(F_{\text{in}}\right) \tag{1}$$

$$F_1 = H_{\text{conv1}}\left(H_{\text{conv3}}\left(H_{\text{conv1}}\left(F_0\right)\right)\right) \tag{2}$$

$$F_{\text{out}} = H_{\text{EDB2}}\left(F_1\right) \tag{3}$$

where H_{EDB1} and H_{EDB2} denote the first EDB and the second EDB, respectively. H_{conv1} and H_{conv3} mean general 1×1 convolution and 3×3 convolution, respectively. F_0 is the extracted feature map by the first EDB. F_1 denotes the feature after three conventional convolutions, and F_{out} is the final output feature of EFEM.

Enhanced Invertible Block. In this subsection, we will introduce in detail our proposed enhanced invertible block (E-InvBlock). In order to enhance both the low and high-frequency inputs for achieving a satisfactory LR frame appearance and ensuring an independent and appropriately distributed latent representation, we utilize a series of E-InvBlocks to refine the LR frame and latent

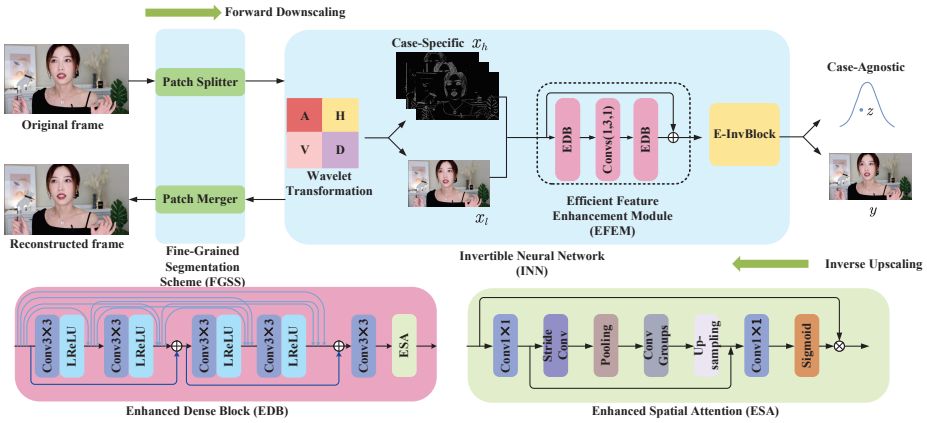

Fig. 2. The overall architecture of EAIN and the structure of EDB and ESA.

representations. First, review the plain invertible block (P-InvBlock) [20]. It is a reorganization of current coupling layers [2,3] to adapt the image rescaling task. The workflow is shown in Fig. 3. Let x_l and x_h denote the low and high-frequency components of the input, respectively. The j-th layer output is

$$x_l^{j+1} = x_l^j + \phi\left(x_h^j\right) \tag{4}$$

$$x_h^{j+1} = x_h^j \odot \exp\left(\rho\left(x_l^{j+1}\right)\right) + \eta\left(x_l^{j+1}\right) \tag{5}$$

where $\odot$ denotes the Hadamard product, $\exp(\cdot)$ denotes the exponential function, ϕ, η and ρ employ Dense Block [18]. It is common to employ a centered sigmoid function to ensure computational stability after exponentiation.

In P-InvBlock, the high-frequency component is polished by two shortcut connections on the low-frequency branch. In E-InvBlock, to retain more high-frequency information, we augment the high-frequency branch. As shown in Fig. 3, the detailed process is:

$$g_j = x_h^j \odot \exp\left(\rho\left(x_l^{j+1}\right)\right) + \eta\left(x_l^{j+1}\right) \tag{6}$$

$$x_h^{j+1} = g_j - \xi\left(x_l^{j+1}\right) \tag{7}$$

this process imposes a slight computation cost but effectively enhances the model's performance. Please note that, in the E-InvBlock, ϕ, η, ρ and ξ employ the residual density block [15], which generates the visual quality of the frame is significantly superior to that of the simple Dense Block [18].

3.3 Fine-Grained Segmentation Scheme

To improve the efficiency of our network, we adopt a more fine-grained segmentation scheme (FGSS). The specific workflow is presented in Fig. 2. Before

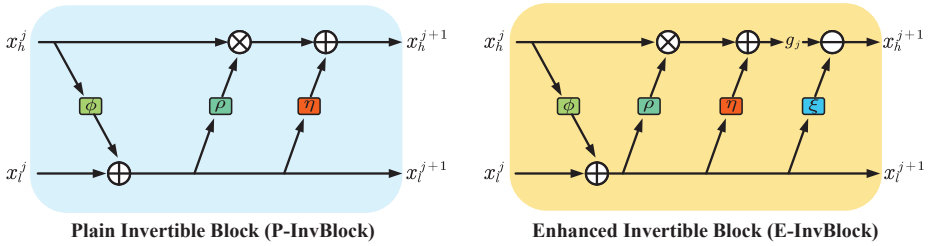

Fig. 3. Workflow diagram of P-InvBlock and E-InvBlock.

downscaling, we first slice each frame of the video into many small patches using the Patch Splitter. After inverse upscaling, we merge the model output patches into complete frames via Patch Merger. These operations are usually executed in seconds and can be regarded as insignificant compared to the entire training process.

3.4 Training Objectives

We train our EAIN network by minimizing the concise loss L_{total}, which is a linear combination of the final HR reconstruction loss L_{rec}, LR guidance loss L_{gui}, and distribution matching loss L_{dis}.

$$L_{\text{total}} = \alpha L_{\text{rec}} + \beta L_{\text{gui}} + \gamma L_{\text{dis}} \tag{8}$$

where α, β and γ are weight factors for balancing different loss terms.

LR Guidance. The LR frames downscaled by the EAIN should exhibit visual appeal. To achieve this, we adopt the approach from prior research [20] where the LR frames are constrained to minic bicubic interpolated frames, providing guidance for the downscaling process f_θ^y:

$$L_{\text{gui}}(\theta) = \sum_{\text{k}=1}^{\text{N}} l_\chi \left(Bicubic\left(x_{\text{k}}\right), f_\theta^y\left(x_{\text{k}}\right) \right) \tag{9}$$

where l_χ represents a difference metric on χ, e.g., the $L1$ or $L2$ loss. Detailed information will be discussed later in the experimental section. x denotes the input HR frame, and N represents the total number of video frames, f_θ^y denotes the EAIN forward process that generates the LR frame y.

HR Reconstruction. Despite the loss of information in the downscaling process, we expect that the reconstructed HR frames should be infinitely close to

the original HR frames. Specifically, the model-downscaled LR frames are upscaled by the inverse upscaling process f_θ^{-1} with a arbitrarily extracted z from the learned distribution $p(z)$.

$$L_{rec}(\theta) = \sum_{\mathrm{k}=1}^{\mathrm{N}} l_v\left(x_{\mathrm{k}}, f_\theta^{-1}\left(f_\theta^y\left(x_{\mathrm{k}}\right), z\right)\right) \tag{10}$$

where l_v calculates the difference between the input HR frame and its reconstructed counterpart. Specific details will be shown in the experimental section.

Distribution Matching. The final objective of the training is to ensure that the model accurately captures the data distribution $q(x)$ of the original HR frames. To be able to make the training process more stable, we adopt the same strategy as IRN [20] and take cross entropy (CE) as the distribution matching loss.

$$L_{\mathrm{dis}}(\theta) = CE\left(\Gamma\left(f_\theta^z\left[q\left(x\right)\right]\right), p\left(z\right)\right) \tag{11}$$

where f_θ^z denotes the EAIN forward process that generates the latent variable z, and original HR frame $x \sim q\left(x\right)$, $p\left(z\right)$ represents the distribution that the latent variable z sampled from the inverse process of EAIN follows. Γ denotes the distribution formed by the EAIN forward process.

4 Experiment

4.1 Implementation Details

Datasets: We followed the previous neural video delivery works [13,10,9], adopting the VSD4K datasets collected in [13]. This dataset comprises 6 video types: dance, sports, game, city, vlog, and interview. Each category consists of multiple video lengths, including 15s, 30s, and 45s. We configure the resolution for HR videos to 1080p, and guided LR videos (270p, 360p, 540p) are generated by bicubic interpolation to adapt corresponding scaling factors ($\times 4$, $\times 3$, $\times 2$). Due to space constraints, specific information about the resolution of the video and the corresponding bitrate can be found in the anonymous Github.

Training protocol: During training, the input HR frames are enhanced by applying random horizontal and vertical rotation. We use the Adam optimizer with parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ to train our model. The mini-batch size is set at 8. The learning rate is initialized as 2×10^{-4} and decays at different iterations depending on the video length. Each INN module comprises 8 E-InvBlocks and downscales the original image by a factor of 2. We set the weight coefficients in the loss function $\alpha = 1$, $\beta = 16$, and $\gamma = 1$.

4.2 Compare with the State-of-the-Art Methods

In this subsection, we compare our method with the state-of-the-art (SOTA) neural video delivery methods that use general model overfitting, time-divided model overfitting, and spatial-temporal data overfitting on different SR networks, including ESPCN [16], VDSR [8], EDSR [12], and WDSR [27]. Owing to space constraints, we randomly select three video categories (game, inter, vlog) from VSD4K and conduct experimental testing on two different video durations, 15s and 45s.

Quantitative Results. The experimental results are illustrated in Table **??**. We compare with the SOTA SR network-based video delivery methods such as awDNN [24], where an entire video is overfitted by an independent SR model; NAS [25], which divides a long video into multiple video segments in advance and overfits each of the time-divided segments with a separate SR model; CaFM [13] takes advantage of time-split video segments and a single SR model with a handmade module to overfit the entire video, and STDO [9] utilizes the spatial-temporal information to accurately divide video into segments, each of which has a similar PSNR value after SR, thus keeping the number of segments as well as the model size to a minimum. It can be seen that our method consistently outperforms the other neural video delivery methods on different SR networks. To be more specific, our EAIN outperforms STDO using the high-performance SR model (WDSR) by 1.98 dB and 2.61 dB at 15s and 45s for $\times 4$, respectively. In addition, our method work better with long videos; on average, it is 2.56 dB better than STDO with the WDSR network for video lengths of 45s. Compared to awDNN and NAS using WDSR, the EAIN is about 6 dB and 4 dB higher for different videos, respectively. More noticeably, EAIN is far superior to these neural video delivery methods using lightweight SR networks, like ESPCN.

Qualitative Results. Fig. 4 shows the qualitative comparison results of various neural video delivery methods using EDSR. It can be clearly observed that our method achieves better visual clarity and authenticity than those of previous approaches. EAIN generates finer and more realistic visuals frames, which contribute to the enjoyable visual experience.

4.3 The benefit of the downscaling process

To demonstrate the advantages of the EAIN downscaling process, we upscale the interpolation-based video frames as well as the video frames generated by the EAIN downscaling process with the inverse process of EAIN, respectively. As shown in Fig. 5, the video quality of the EAIN downscaling video is much higher than that of the interpolated video after upscaling. This powerfully illustrates the rationality and effectiveness of our approach for neural video delivery.

Table 1. Quantitative results of EAIN with different neural video delivery methods.

Model	Dataset	game-15s			inter-15s			vlog-15s		
	Scale	×2	×3	×4	×2	×3	×4	×2	×3	×4
ESPCN	awDNN [24]	37.94	32.85	29.97	40.43	35.36	29.91	46.41	42.90	39.65
	NAS [25]	37.58	32.71	30.59	40.62	35.42	30.43	46.53	43.01	39.98
	CaFM [13]	38.07	33.14	30.96	40.71	35.54	30.47	47.02	43.20	40.16
	STDO [9]	38.61	33.57	31.30	42.65	35.63	30.63	47.11	43.25	40.73
VDSR	awDNN [24]	41.27	35.03	32.16	44.16	35.99	30.65	48.18	43.03	41.07
	NAS [25]	42.53	35.97	33.86	44.71	36.57	31.05	48.49	43.41	41.33
	CaFM [13]	43.02	36.17	33.98	44.85	36.46	31.08	48.61	43.62	41.49
	STDO [9]	43.56	36.71	35.02	45.16	36.81	33.43	48.75	43.82	41.71
EDSR	awDNN [24]	42.24	35.88	33.44	43.06	37.89	34.94	48.87	44.51	42.58
	NAS [25]	42.82	36.42	34.00	45.06	38.38	35.47	49.10	44.80	42.83
	CaFM [13]	43.13	37.04	34.47	45.35	38.66	35.70	49.30	45.03	43.12
	STDO [9]	44.93	37.80	35.47	45.91	39.26	36.76	50.24	45.68	43.46
WDSR	awDNN [24]	43.36	37.12	34.62	44.83	39.05	35.23	49.24	45.30	43.33
	NAS [25]	44.17	38.23	36.02	45.43	39.71	36.54	49.98	45.63	43.51
	CaFM [13]	44.23	38.55	36.30	45.71	39.92	36.87	50.12	45.87	43.79
	STDO [9]	45.75	40.17	38.62	46.34	41.13	38.76	50.58	46.43	44.62
INN	**EAIN(ours)**	**46.54**	**41.59**	**40.27**	**47.62**	**44.31**	**42.12**	**51.78**	**47.84**	**45.54**
		game-45s			**inter-45s**			**vlog-45s**		
		×2	×3	×4	×2	×3	×4	×2	×3	×4
ESPCN	awDNN [24]	35.42	30.63	28.65	38.64	31.97	28.32	45.71	41.40	39.20
	NAS [25]	35.55	30.67	28.74	38.81	32.14	28.61	45.81	41.52	39.29
	CaFM [13]	36.09	31.06	29.05	38.88	32.22	28.75	46.19	41.72	39.52
	STDO [9]	37.75	32.29	29.96	41.20	32.48	29.09	46.33	42.26	40.26
VDSR	awDNN [24]	40.29	34.53	31.28	41.99	33.80	30.34	47.61	42.92	40.94
	NAS [25]	41.37	34.92	32.42	42.40	34.53	31.10	47.88	43.33	41.23
	CaFM [13]	41.92	35.56	33.16	42.86	34.49	30.95	48.00	43.50	41.38
	STDO [9]	42.65	36.23	33.76	43.36	35.64	31.77	48.17	43.67	41.49
EDSR	awDNN [24]	42.11	35.75	33.33	42.73	34.49	31.34	47.98	43.58	41.53
	NAS [25]	43.22	36.72	34.32	43.31	35.80	32.67	48.48	44.12	42.12
	CaFM [13]	43.32	37.19	34.61	43.37	35.62	32.35	48.45	44.11	42.16
	STDO [9]	45.65	39.93	37.24	44.52	38.28	35.51	49.84	45.47	43.07
WDSR	awDNN [24]	42.61	36.17	33.85	42.94	34.71	31.81	48.02	44.16	42.19
	NAS [25]	43.72	37.25	34.93	43.41	36.05	33.11	48.52	44.75	42.80
	CaFM [13]	43.97	37.64	35.12	43.52	36.03	32.97	48.51	44.72	42.87
	STDO [9]	45.71	40.33	37.76	44.54	38.72	36.03	49.76	45.95	43.99
INN	**EAIN(ours)**	**46.80**	**42.30**	**40.30**	**46.95**	**43.53**	**39.87**	**52.24**	**48.36**	**45.45**

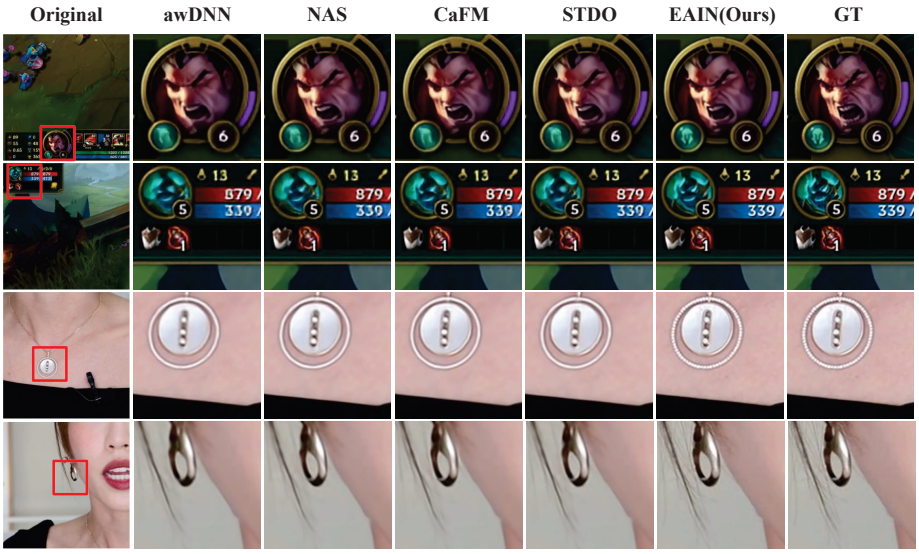

Fig. 4. Qualitative results of EAIN with different neural video delivery methods.

4.4 Effects of the loss function

In this subsection, we conduct experiments to demonstrate the impacts of various loss functions based on EAIN. The experiment results are summarized in Table 2. The optimal performance of EAIN is achieved when the LR guidance loss and HR reconstruction loss are formulated as the $L2$ loss with distribution matching loss. The specific reason is that, in contrast to the $L1$ loss function, the $L2$ loss function possesses some favorable mathematical properties, such as continuous derivability as well as convexity, which means that in the optimization process, the global optimal solution can be found more easily, avoiding the problem of locally optimal solutions. Moreover, the $L2$ loss function is computationally efficient as it solely requires squaring the difference between the predicted and actual values. This enables faster computation of the loss values and subsequent parameter updates during the training process. In addition, the distribution matching loss also contributes to the performance of the model.

4.5 Ablation study

Effectiveness of the proposed Enhanced InvBlock We conduct experiments to show the validity of our proposed E-InvBlock through comparative analysis with the P-InvBlock. The results of the correlational analysis can be compared in Table 3. Note that we train the EAIN with E-InvBlock and P-InvBlock on game-15s and vlog-15s videos for $\times 4$ upscaling. We can observe

Table 2. Quantitative results (PSNR/SSIM) of training EAIN with $L1$ or $L2$ LR guide and HR reconstruction loss, with/without distribution matching loss on game-15s and vlog-15s datasets with scale $\times 4$.

L_{gui}	L_{rec}	L_{dis}	**game-15s**	**vlog-15s**
L_1	L_1	✓	34.21/0.956	41.85/0.969
L_1	L_2	✓	32.87/0.938	40.56/0.965
L_2	L_1	✓	40.00/0.979	45.38/0.985
L_2	L_2	✓	**40.27/0.982**	**45.54/0.989**
L_2	L_2	✗	40.12/0.979	45.47/0.986

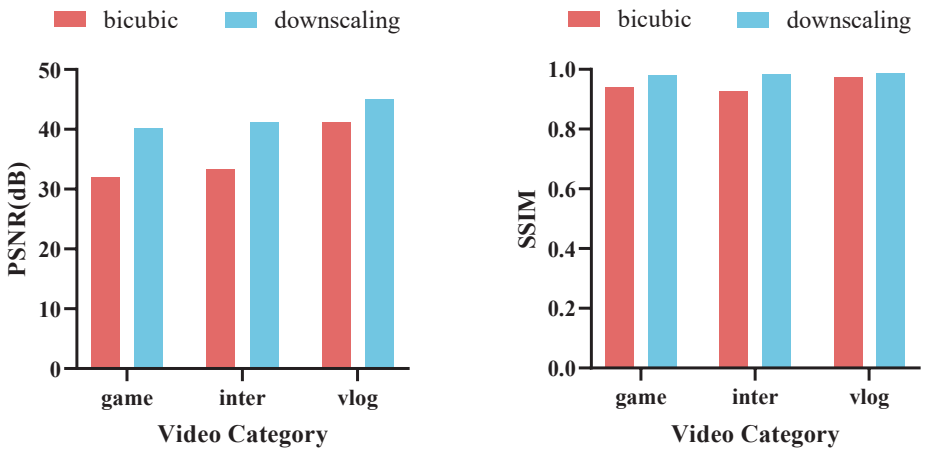

Fig. 5. Quantitative comparison result for upscaling the bicubic interpolation-based video frames and the video frames generated by the EAIN downscaling process with the inverse process of EAIN.

that our proposed E-InvBlock obtains better performance gains compared to P-InvBlock. The experimental results demonstrate that our proposed E-InvBlock substantially increases the model capacity.

Effectiveness of EFEM We additionally performed an ablation experiment to confirm the efficacy of the EFEM. We train EAIN with EFEM and without EFEM on game-15s and vlog-15s video for $\times 4$ upscaling. It is apparent from Table 4 that a performance increase appears for EAIN with EFEM. Compared to EAIN without EFEM, the complete EAIN obtains performance gains of 0.38 dB, 0.62 dB for game-15s and vlog-15s, respectively. The experimental results show that EFEM can effectively enhance the nonlinear representativeness of our network.

Benefit of FGSS As depicted above, the FGSS plays a crucial role in enabling our method to attain superior performance. To evaluate the advantage of FGSS, we train the EAIN with and without FGSS separately; not using FGSS

Table 3. Quantitative comparison of P-InvBlock and E-InvBlock.

Method	game-15s	vlog-15s
P-InvBlock	39.56/0.961	44.98/0.969
E-InvBlock	**40.27/0.979**	**45.54/0.989**

Table 4. Ablation study on EFEM and FGSS.

EFEM	FGSS	game-15s	vlog-15s
✗	✗	38.08/0.947	42.95/0.949
✗	✓	39.89/0.950	44.92/0.953
✓	✗	38.68/0.952	43.43/0.961
✓	✓	**40.27/0.979**	**45.54/0.989**

means sending the entire frame for training. Table 4 shows that using FGSS results in approximately 1.6 dB and 2.1 dB higher gains for game-15s and vlog-15s respectively compared to not using FGSS, which fully proves the benefit of the FGSS. In addition, FGSS can help improve the efficiency of model training, using FGSS to train a model in the NVIDIA GeForce RTX 3090 takes 5 hours less than without it.

5 Conclusion

In this paper, we propose an Enhanced Asymmetric Invertible Network (EAIN) for neural video delivery, in which the inherent ill-posedness of the SR network is significantly alleviated. Our method uses the forward process of the network to generate high-quality LR videos while capturing the distribution of missing information using a latent variable that follows an isotropic Gaussian distribution in the downscaling process. HR videos are available by passing a arbitrarily extracted latent variable through the network in reverse with LR videos. Moreover, we also propose a new Enhanced Invertible Block (E-InvBlock) in the INN module, which can effectively recover the HR frame in the upscaling process. To improve the nonlinear representativeness of E-InvBlock, we present an Effective Feature Enhancement Module (EFEM). In addition, we adopt a more Fine-Grained Segmentation Scheme (FGSS) for video frames, which further improves the quality of the video stream on the client side and reduces the network training burden. Comprehensive experiments demonstrate that our method significantly improves the quality of video streaming on the client side.

Acknowledgments. This work was supported in part by the Fundamental Research Funds for the Central Universities.

References

1. Ardizzone, L., Lüth, C., Kruse, J., Rother, C., Köthe, U.: Guided image generation with conditional invertible neural networks. arXiv preprint arXiv:1907.02392 (2019)
2. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
3. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
4. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence **38**(2), 295–307 (2015)
5. Jiang, X., Peng, X., Zheng, C., Xue, H., Zhang, Y., Lu, Y.: End-to-end neural speech coding for real-time communications. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 866–870. IEEE (2022)
6. Kim, H., Choi, M., Lim, B., Lee, K.M.: Task-aware image downscaling. In: Proceedings of the European conference on computer vision (ECCV). pp. 399–414 (2018)
7. Kim, J., Jung, Y., Yeo, H., Ye, J., Han, D.: Neural-enhanced live streaming: Improving live video ingest via online learning. In: Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication. pp. 107–125 (2020)
8. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1646–1654 (2016)
9. Li, G., Ji, J., Qin, M., Niu, W., Ren, B., Afghah, F., Guo, L., Ma, X.: Towards high-quality and efficient video super-resolution via spatial-temporal data overfitting. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10259–10269. IEEE (2023)
10. Li, X., Liu, J., Wang, S., Lyu, C., Lu, M., Chen, Y., Yao, A., Guo, Y., Zhang, S.: Efficient meta-tuning for content-aware neural video delivery. In: European Conference on Computer Vision. pp. 308–324. Springer (2022)
11. Li, Y., Liu, D., Li, H., Li, L., Li, Z., Wu, F.: Learning a convolutional neural network for image compact-resolution. IEEE Transactions on Image Processing **28**(3), 1092–1107 (2018)
12. Lim, B., Son, S., Kim, H., Nah, S., Mu Lee, K.: Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 136–144 (2017)
13. Liu, J., Lu, M., Chen, K., Li, X., Wang, S., Wang, Z., Wu, E., Chen, Y., Zhang, C., Wu, M.: Overfitting the data: Compact neural video delivery via content-aware feature modulation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 4631–4640 (2021)
14. Liu, J., Tang, J., Wu, G.: Residual feature distillation network for lightweight image super-resolution. In: Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. pp. 41–55. Springer (2020)
15. Rakotonirina, N.C., Rasoanaivo, A.: Esrgan+: Further improving enhanced super-resolution generative adversarial network. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3637–3641. IEEE (2020)

16. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1874–1883 (2016)
17. Sun, W., Chen, Z.: Learned image downscaling for upscaling using content adaptive resampler. IEEE Transactions on Image Processing **29**, 4027–4040 (2020)
18. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European conference on computer vision (ECCV) workshops. pp. 0–0 (2018)
19. Wilson, P.I., Fernandez, J.: Facial feature detection using haar classifiers. Journal of computing sciences in colleges **21**(4), 127–133 (2006)
20. Xiao, M., Zheng, S., Liu, C., Wang, Y., He, D., Ke, G., Bian, J., Lin, Z., Liu, T.Y.: Invertible image rescaling. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16. pp. 126–144. Springer (2020)
21. Yang, J., Wright, J., Huang, T.S., Ma, Y.: Image super-resolution via sparse representation. IEEE transactions on image processing **19**(11), 2861–2873 (2010)
22. Yang, J., Guo, M., Zhao, S., Li, J., Zhang, L.: Self-asymmetric invertible network for compression-aware image rescaling. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 3155–3163 (2023)
23. Yeo, H., Chong, C.J., Jung, Y., Ye, J., Han, D.: Nemo: enabling neural-enhanced video streaming on commodity mobile devices. In: Proceedings of the 26th Annual International Conference on Mobile Computing and Networking. pp. 1–14 (2020)
24. Yeo, H., Do, S., Han, D.: How will deep learning change internet video delivery? In: Proceedings of the 16th ACM Workshop on Hot Topics in Networks. pp. 57–64 (2017)
25. Yeo, H., Jung, Y., Kim, J., Shin, J., Han, D.: Neural adaptive content-aware internet video delivery. In: 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18). pp. 645–661 (2018)
26. Yeo, H., Lim, H., Kim, J., Jung, Y., Ye, J., Han, D.: Neuroscaler: Neural video enhancement at scale. In: Proceedings of the ACM SIGCOMM 2022 Conference. pp. 795–811 (2022)
27. Yu, J., Fan, Y., Yang, J., Xu, N., Wang, Z., Wang, X., Huang, T.: Wide activation for efficient and accurate image super-resolution. arXiv preprint arXiv:1808.08718 (2018)
28. Zhang, R., Du, L., Liu, J., Song, C., Wang, F., Li, X., Lu, M., Guo, Y., Zhang, S.: Repcam: Re-parameterization content-aware modulation for neural video delivery. In: Proceedings of the 33rd Workshop on Network and Operating System Support for Digital Audio and Video. pp. 1–7 (2023)