DeNC++: Efficient Diffusion-Enhanced Neural Codec for End-to-end Semantic Streaming at the Edge

Qihua Zhou[1], Wangjiang Gong[2], Zili Meng[2], Yaxiong Xie[3], Yaodong Huang[1], Junchen Jiang[4], Laizhong Cui[1]*

[1]College of Computer Science and Software Engineering, Shenzhen University
[2]Hong Kong University of Science and Technology
[3]State University of New York at Buffalo
[4]University of Chicago
{qihuazhou, yd.huang, cuilz}@szu.edu.cn, wgongad@connect.ust.hk, zilim@ust.hk
yaxiongx@buffalo.edu, junchenj@uchicago.edu

Abstract

The neural-enhanced video streaming (NeVS) has been an emerging technique to integrate neural models into video codecs for higher streaming efficiency. The state-of-the-art methods, *e.g.,* DeNC and Gemino, typically compress videos in RGB space and restore video quality via a neural enhancement model hosted on the external media server. However, these methods are not always accessible in resource-constrained edge environments due to their heavy reliance on the media server's computation, which undermines end-to-end performance and restricts NeVS's usage boundary. This limitation raises an interesting question: *is it possible to make NeVS lightweight so that all neural codec operations can be handled directly by clients' edge devices?* In this paper, we present the answer *yes* and develop a new plug-and-play module called DeNC++, which significantly improves the *compression-restoration-overhead* trade-off over existing methods. Our core design philosophy is to wrap all the codec operations within a latent semantic space, in which the original high-dimensional visual signals are efficiently embedded into low-dimensional semantic representations. With this fundamental transformation, DeNC++'s neural encoder introduces the triple *semantic-bitwidth-resolution* compression to effectively lower the streaming traffic. Meanwhile, we make DeNC++'s neural decoder aware of the perceptual loss caused by its encoder and design tiny generative models to guarantee high restoration quality. We also strictly restrict the runtime computational overhead and accelerate the neural enhancement process, making DeNC++ compatible with commodity edge devices. Real-world evaluations reveal that DeNC++ consistently provides higher restoration quality while achieving 24-55× higher compression ratio and 5-7× end-to-end speedup over the latest NeVS solutions.

Introduction

Increasingly, video streaming has become a fundamental information carrier for diverse services, *e.g.,* Zoom meeting (Zoom 2025), YouTube live (YouTube 2025b) and Netflix (Netflix 2025a), which significantly promote the development of multimedia entertainment and virtual collaboration in human daily life. In general, the objective of a high-performance video streaming system is to minimize the streamer's encoding bitrate for efficient transmission while not harming the viewer's perceptual quality after decoding, *i.e.,* achieving a high *compression-restoration* trade-off between traffic saving and quality preservation. Consequently, there have been research efforts for decades, from conventionally proposing new video codecs (*e.g.,* VP9 (Google 2025) and AV1 (AOMedia 2025)) to most recently introducing neural-enhanced video streaming (NeVS) atop of existing video codecs. NeVS (Zhou et al. 2025; Cheng et al. 2024) can further compress the video bitrate while maintaining comparable or even better perceptual quality by running neural enhancement models such as super-resolution.

Unfortunately, existing NeVS solutions are computational intensive and are hard to deploy on clients' edge devices directly. For example, Gemino (Sivaraman et al. 2024) re-

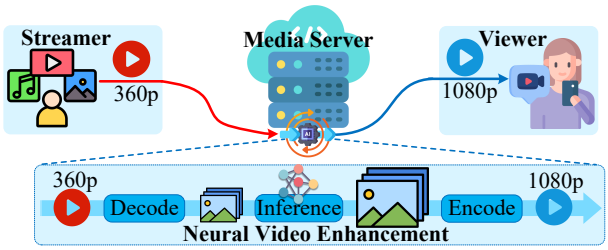

Figure 1: Prior NeVS methods rely on external media server to deploy neural enhancement models inside the codec.

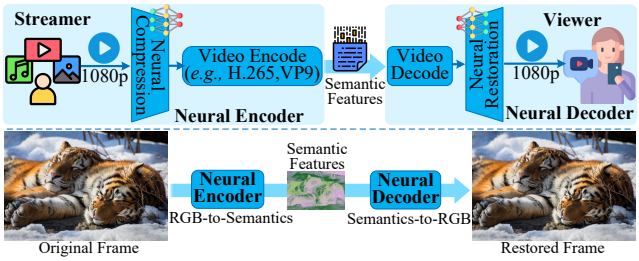

Figure 2: Our DeNC++ avoids involving media servers and handles neural codec operations totally on clients.

*Corresponding author.

quires NVIDIA A100 GPUs to decode the video stream at 30fps and NeuroScaler (Yeo et al. 2022) requires multiple NVIDIA T4 GPUs to restore videos from 360p to 1080p. Even if some of them are designed for mobile devices, such as NEMO (Yeo et al. 2020), they still rely on the collaboration of the media server to prepare super-resolution models. Therefore, most of the existing NeVS solutions have to rely on the computational resources at the media servers to run these heavyweight neural models. As shown in Figure 1, when the streamer generates the video contents at a low bitrate, the media server in the middle will enhance the contents with the neural models and deliver the high-quality video to the viewers. With the increase of the price of GPUs recently (Amazon 2025; Microsoft 2025a; NVIDIA 2025a), such a high computational overhead slows down the adoption of NeVS in practice. Moreover, enhancing the video on the media servers can only alleviate the ingest bandwidth bottleneck between the streamer and the media server, while the downlink between the server and viewer can also be the bottleneck (Meng et al. 2022). There are also emerging scenarios where peer-to-peer services receive increasingly more attention, such as Wi-Fi Direct gaming streaming (Sony 2025; Microsoft 2025b; Kirmizioglu and Tekalp 2020), which also cannot enjoy the benefits of NeVS due to lack of powerful machines. As a result, the current NeVS paradigm is not always the "panacea" for modern streaming services with hardware and usage restrictions. In this case, we are motivated to design a new end-to-end NeVS methodology that is fully deployable on commercial clients' edge devices with no need of external computing resources.

The question in this paper is: *Can we have a NeVS solution that is lightweight enough to be directly deployed on clients' edge devices?* In this paper, we present DeNC++, the first end-to-end lightweight NeVS plugin for practical edge environments to the best of our knowledge. As shown in Figure 2, DeNC++ no longer needs high-end GPUs such as NVIDIA A100 but is fully deployable on commercial edge devices. The insight behind DeNC++ is to let the streamer encode some auxiliary information for the viewer to enhance the video content. It is worth noting that DeNC++ does not simply move the computations from the media server to the viewer – instead, the additional computational and bandwidth overhead at the viewer is lightweight. This comes from a co-design between the streamers and viewers that makes the streamer know how to encode most efficiently, which will also be later demonstrated in our evaluation.

However, we need to address four major challenges: ❶ achieving a high compression-restoration-overhead trade-off on clients' edge devices, ❷ improving compression ratio to fit limited traffic budget, ❸ refining the perceptual quality of the restoration videos, ❹ reducing runtime overhead of inference time cost and memory footprint. We stand from the edge-friendly perspective and present corresponding solutions to address the above four challenges via the algorithm-system co-design. First, different from existing NeVS methods that capture and process the visual signals in the RGB space, we wrap all the video codec operations in a *latent semantic space*, where the original high-dimensional visual signals are effectively embedded into low-dimensional semantic representations. Second, we re-design the video encoder by empowering it with intelligent compression capability in three aspects: visual semantics, feature resolution and pixel bitwidth. This triple *semantic-bitwidth-resolution* compression can additionally reduce the video size by nearly one order of magnitude compared to the pure H.265 (H.265 2025), thus significantly improving the streaming efficiency. Third, we integrate the neural enhancement capability into the video decoder and make it adapt to the encoder's triple compression. The decoder is specifically optimized as an inverse process of the encoder, thus holding a robust restoration capability to compensate for the perceptual loss of compressed videos. Fourth, we optimize the neural inference speed and manage memory footprint of the entire encoder-decoder procedure. Additional refinements are also conducted to make our design adaptive to runtime properties on edge devices.

We implement DeNC++ on the NVIDIA Jetson AGX Orin (NVIDIA 2025b), a typical kind of edge device with ease-of-use APIs. DeNC++ is compatible with conventional video codecs, *e.g.,* H.265/HEVC (H.265 2025), so the benefits of these codecs are still well preserved. Real-world evaluation shows the effectiveness of our DeNC++ in diverse performance metrics, including traffic bitrate reduction, restoration quality improvement, computational cost saving. DeNC++ consistently achieves a higher compression-restoration-overhead trade-off over previous NeVS methods. Specifically, DeNC++ achieves $24\text{-}55\times$ higher compression ratio, better multi-metric restoration quality (*e.g.,* 0.944 VMAF) and $5\text{-}7\times$ end-to-end speedup, compared to NeuroScaler (Yeo et al. 2022), NEMO (Yeo et al. 2020) and Gemino (Sivaraman et al. 2024), respectively.

In summary, our key contributions are as follows.

- **End-to-end optimization.** We conduct a holistic synergy of encoder-decoder rather than single-point NeVS, where our RGB-to-semantic transformation improves overall compression-restoration-overhead trade-off.
- **Lightweight neural codec.** To fit edge resource constraints, we restrict the runtime overhead by reducing inference time and memory footprint, making the neural codec compatible to commodity edge devices.
- **Efficient plug-and-play implementation.** We implement DeNC++ to enable lightweight NeVS services in realistic edge environments. DeNC++ outperforms the latest NeVS solutions with $24\text{-}55\times$ higher compression ratio and $5\text{-}7\times$ end-to-end speedup.

The project of DeNC++ will be open-source and will constantly contribute to the neural video codec community.

Related Work

Rise of Neural-enhanced Video Streaming. As shown in Figure 1, the neural-enhanced video streaming (NeVS) (Zhou et al. 2025; Cheng et al. 2024; Chen et al. 2024) is an emerging technique to save streaming traffic while preserving restoration quality. In general, existing NeVS methods involve collaborations between the streamer and the external media server to improve the video delivery performance (Sivaraman et al. 2024; Zhang et al. 2022). First,

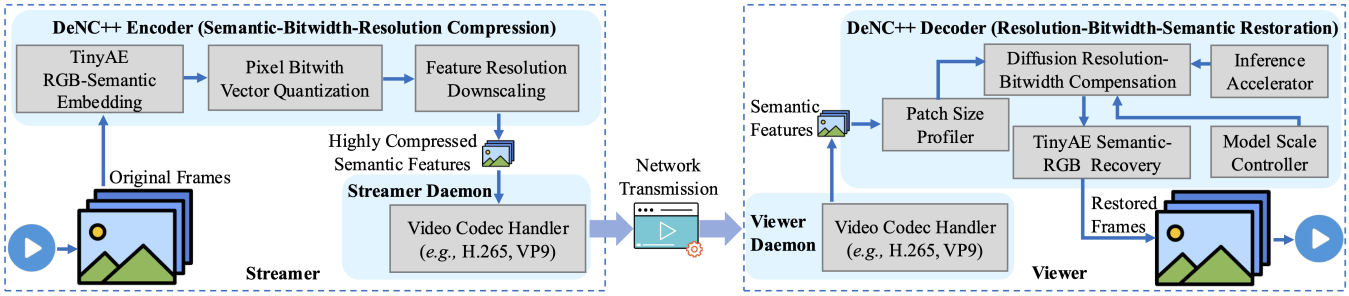

Figure 3: Overview of our DeNC++ based on the RGB-semantic transformation, where the lightweight encoder and decoder handle all the video compression, transmission and restoration in the latent semantic space.

the streamer downscales the original high-resolution video frames into the low-resolution version, then encodes the frames into video bitstreams for network transmission. After receiving, the media server decodes the bitstreams as a series of frames and feeds them into a neural super-resolution (SR) model for quality enhancement (Yeo et al. 2022; Zhang et al. 2022; Nguyen et al. 2022). The final restored video holds a comparable visual experience as the original version. Recently, optimizing the NeVS pipeline has become a hot topic, including improving video encoding efficiency (Du et al. 2022; Dasari et al. 2022), reducing streaming latency (Yeo et al. 2018), and optimizing adaptive bit-rate delivery (Dasari et al. 2022) These studies can be regarded as complementary works to ours for optimizing different stages of the NeVS pipeline.

Neural Models for Codec Enhancement. Restoring compressed videos with high perceptual quality is a fundamental component of the NeVS paradigm. Current mainstream NeVS methods often employ a neural super-resolution (SR) model for quality enhancement (Sivaraman et al. 2024; Ye et al. 2023). Since the restoration capacity of a neural enhancement model is closely related to the parameter scale and pre-training scheme, the current NeVS methods are hard to correlate well with human perception, especially with low input resolutions and large scaling factors. Meanwhile, the recent diffusion-based *generative* models (*e.g.,* DDPM (Ho, Jain, and Abbeel 2020), DDIM (Song, Meng, and Ermon 2021) and LDM (Rombach et al. 2022)) have achieved impressive performance in diverse vision tasks, including inpainting (Lugmayr et al. 2022), colorization (Song et al. 2021), image synthesis (Saharia et al. 2023) and visual restoration (Kawar et al. 2022). Although the diffusion generative models can serve as potential tools to restore video quality, their inference overhead is extremely huge, thus hampering the deployment to NeVS services on clients' edge devices. To fully adapt the diffusion generative models to edge environments, we need to redesign the neural architecture and restrict the computational cost. The major target is to lower model complexity and accelerate inference speed.

Our Insight. Current NeVS methods mostly rely on the media server's computational power to handle the neural enhancement operations. They are computation-intensive and require expensive hardware resources, *e.g.,* NVIDIA A100 GPUs for Gemino (Sivaraman et al. 2024) and NVIDIA T4 GPUs for NeuroScaler (Yeo et al. 2022). The heavy computational dependency of the external media server hampers the end-to-end streaming performance and limits the deployment potential to hardware-constrained scenarios, *e.g.,* Peer-to-Peer streaming (Time 2025; SopCast 2025), Wi-Fi Direct streaming (Sony 2025; Microsoft 2025b) and Ad-hoc streaming (Farahani et al. 2022). As a result, restricting the computational overhead of neural codecs and fitting the environments of clients' edge devices are not well explored by existing works. This motivates us to evolve current NeVS paradigm into a lightweight and edge-friendly manner, where the computations can be totally handled by edge device, so as to significantly extend the usage boundary.

Methodology

Encoder for Video Compression

DeNC++'s encoder aims to compress the video frames in three aspects: visual semantics, pixel bitwidth and feature resolution. As shown in the left half in Figure 3, we call it triple *semantic-bitwidth-resolution* compression.

TinyAE RGB-Semantic Embedding. The first compression perspective is to transfer the video frames from the original RGB space to latent semantic space. To accommodate the limited model parameters requirements of edge devices, we develop a new autoencoder module with a tiny neural structure. As shown in Figure 4, we call it TinyAE. Since the latent space efficiently represents the data in a low-dimensional space, both the forward adding-noise process and the backward sampling process of the diffusion model are conducted in the latent space, resulting in reduced computational costs for inference and training. Our TinyAE only holds $< 0.7M$ parameters to ensure efficient encoding-decoding capabilities, while preserving the decoding quality. TinyAE consists of dilated convolutional and activation layers with highly pruned channels. Given a $H \times W$ input frame with a $K \times K$ convolutional kernel under L stride length, the output size is approximately reduced into $\frac{L^2 HW}{[(H-K)L+1]\cdot[(W-K)L+1]}$ compared to the input, when omitting the impact of zero padding. Together with a series of sequential convolutional layers inside TinyAE's encoder, we can significantly reduce the representation dimensions of the video signals and provide a compression ratio of nearly one order of magnitude compared to the original input size. In

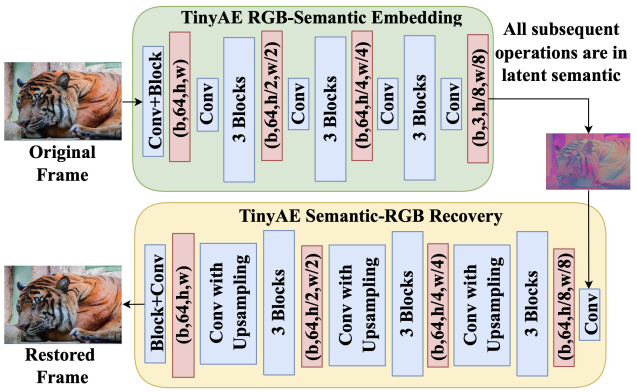

Figure 4: The structure of TinyAE, which contains a series of convolutional layers, activations and fusion.

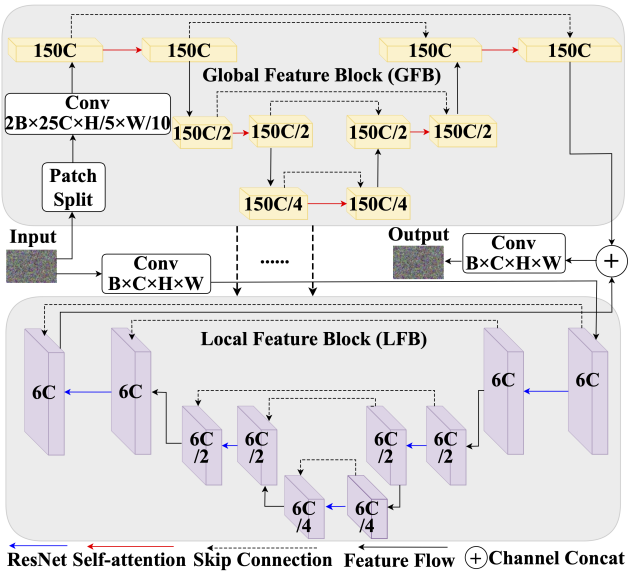

Figure 5: The structure of our DCP-Unet backbone.

practice, we set the channel number of TinyAE's output to 3, so the output can be saved in common image formats for visualization, *e.g.,* PNG and JPEG. Thus, traditional video codecs like H.265, VP9, and AV1 can directly use these output images for video encoding. This compatibility allows DeNC++'s encoder to work seamlessly with traditional video codecs while still retaining their benefits.

Bitwidth Vector Quantization. After transferring the frames from RGB signals into latent semantic features, the second compression perspective is to reduce the "pixel" bitwidth, *i.e.,* the number of bits to identify a unique pixel inside the feature. Since the semantic features are embedded by the TinyAE's encoder, their spatial regions often hold specific vector-wise similarity across channels, especially when the neural kernels extract interrelated semantics with shallow channels. However, these pixels are commonly organized in the 32-bit floating point (FP32) format, which is often redundant to precisely represent the semantic features. This motivates us to revisit the *Vector Quantization* (Zhou et al. 2022) technique and reduce the number of different pixel values. For example, if we quantize the pixels into 4 bits with 2^4 different values, then the feature size will be further compressed into $4/32$ as the original feature with full-bitwidth pixels. The key here is to find a proper vector quantization scheme to transfer the full-bitwidth pixels into low-bitwidth ones. Specifically, we use the K-means clustering to handle the quantization procedure, which corresponds to the principle of vector quantization. Theoretically, given the pixel bitwidth n, we can figure out the entire compression ratio over traditional 32-bit full-bitwidth features as $\frac{32}{n}$.

Patch-wise Feature Resolution Downscaling. The third compression perspective is to downscale the feature resolution, *i.e.,* shrinking the spatial size under the control of a scaling factor s. As the number of pixels within the feature is reduced in both width and height dimensions, resolution downscaling can provide a $s^2\times$ size reduction compared with the original feature. Although fewer pixels are used to represent a feature, its basic semantics should be preserved. Otherwise, the visual quality degradation will exceed the decoder's recovery capacity. This property requires the scaling algorithm to retain the most representative pixels by analyzing the numerical distribution in each feature patch (*i.e.,* the macroblock, a non-overlapping square block with $s \times s$ pixels in spatial). In default, we suggest using 4×4 patch size, which is a fine-grained granularity to retain essential semantics after downscaling. Therefore, as for each pixel, we can figure out a patch where this pixel is located in the center. Specifically, considering the pixels on the feature border, we adopt zero-padding to the border with $\lceil \frac{s}{2} \rceil$ pixels in width and height, so as to guarantee complete patches. Given a scaling factor s, we can divide the feature into a series of $s \times s$ patches. Based on the patch division, we introduce a *Gaussian Blur* to the feature and smooth the edges involving a junction of patches. Inside each patch, we calculate the weighted average of all the pixels inside and shrink the patch by this average. Consequently, through the above semantic-bitwidth-resolution triple compression, DeNC++ can improve the compression ratio by nearly one order of magnitude compared to vanilla H.265 (H.265 2025).

Decoder for Video Enhancement

Upon receiving the highly compressed semantic features, DeNC++'s neural decoder will conduct restoration through two core modules: diffusion resolution-bitwidth compensation and TinyAE semantic-RGB recovery. The former compensates for the perceptual loss of pixel bitwidth and feature resolution. The latter transfers the semantic features from latent space back to RGB space. As shown in the right half in Figure 3, we call it resolution-bitwidth-semantic restoration, which is the inverse process of encoder's compression.

Diffusion Resolution-Bitwidth Compensation. This module serves as the inverse function of bitwidth-resolution compression. Common diffusion generative models (*e.g.,* SR3 (Saharia et al. 2023) and LDM (Rombach et al. 2022)) are computationally expensive and cannot meet the resource constraints on edge devices. As shown in Figure 5, we develop a novel diffusion model called *Dual-Channel*

Dataset	Method	Bitrate (Avg.)	CR	PSNR↑	SSIM↑	VMAF↑	FVD↓	UIQI↑	LPIPS↓	MUSIQ↑	CLIPIQA↑
BDD100K 720p@25fps	H.265	2.5 Mbps	1×	36.39	0.962	99.73	0.36	0.998	0.018	59.86	0.693
	NEMO	452 Kbps	5.4×	28.04	0.806	97.11	3.35	0.955	0.156	49.77	0.628
	Gemino	316 Kbps	7.9×	30.77	0.828	97.41	2.83	0.961	0.137	48.94	0.632
	NeuroScaler	449 Kbps	5.5×	30.82	0.906	99.11	1.27	0.985	0.066	52.15	0.676
	DeNC	263 Kbps	9.5×	30.86	0.889	99.53	1.85	0.984	0.101	52.82	0.653
	DeNC++	73Kbps	35.3×	30.49	0.850	99.63	2.15	0.983	0.104	53.57	0.665
FFHQ 1024^2@30fps	H.265	4 Mbps	1×	37.86	0.975	99.61	0.43	0.998	0.034	53.89	0.773
	NEMO	952 Kbps	4.2×	28.62	0.888	93.35	8.77	0.903	0.247	41.03	0.661
	Gemino	714 Kbps	5.6×	31.04	0.953	97.54	1.83	0.972	0.121	44.23	0.685
	NeuroScaler	950 Kbps	4.2×	29.86	0.925	96.82	2.08	0.963	0.170	49.13	0.724
	DeNC	625 Kbps	6.4×	31.27	0.967	98.86	1.68	0.989	0.087	49.66	0.749
	DeNC++	162 Kbps	24.1×	31.15	0.944	97.78	2.91	0.984	0.054	43.28	0.699
VisDrone2020 1080p@60fps	H.265	5 Mbps	1×	36.36	0.964	98.16	0.55	0.997	0.015	65.84	0.691
	NEMO	724 Kbps	6.9×	27.74	0.711	93.58	3.31	0.882	0.221	58.84	0.584
	Gemino	641 Kbps	7.8×	30.76	0.782	97.41	1.86	0.973	0.097	62.06	0.633
	NeuroScaler	717 Kbps	6.9×	29.24	0.706	96.14	2.08	0.929	0.181	59.07	0.606
	DeNC	508 Kbps	9.84×	29.23	0.726	96.59	2.24	0.962	0.078	63.14	0.632
	DeNC++	126 Kbps	38.4×	29.68	0.731	97.16	2.67	0.953	0.064	63.02	0.627
DIV2K 2K@30fps	H.265	8 Mbps	1×	43.29	0.995	99.87	0.57	0.998	0.025	61.15	0.821
	NEMO	1.7 Mbps	4.7×	30.74	0.918	95.75	3.62	0.926	0.188	42.15	0.734
	Gemino	816 Kbps	9.8×	27.74	0.793	92.21	7.35	0.903	0.114	40.68	0.716
	NeuroScaler	1.6 Mbps	4.8×	33.38	0.945	99.11	2.69	0.977	0.084	48.72	0.791
	DeNC	597 Kbps	13.4×	32.93	0.941	99.46	3.55	0.993	0.038	47.11	0.752
	DeNC++	145 Kbps	54.9×	32.07	0.937	99.37	4.09	0.991	0.040	43.92	0.725

Table 1: The comparison of parameter scales, compression ratios and restoration quality between NeVS baselines and ours. The video size obtained by pure H.265 serves as the lower bound to calculate compression ratios. The scores of eight metrics achieved by H.265 can be regarded as the upper bound of restoration quality. The parameter scales of NEMO, Gemino, Neuroscaler and ours are 0.8M, 155M, 2.4M and 15.31M (TinyAE: 0.61M & DCP-Unet: 14.7M), respectively.

Pruning Unet (DCP-Unet) with a highly efficient structure. Our DCP-Unet comprises two main blocks: Global Feature Block (GFB) and Local Feature Block (LFB). Based on this dual structure, DCP-Unet can extract both global and local features through the GFB and LFB, respectively. In GFB, the semantic features are divided into patches of sizes ranging from 16×16 to 32×32. These patches are stacked along the channel dimension, thus enhancing the restoration quality via self-attention blocks. Unlike the traditional U-Net, which reduces the size of semantic features, our DCP-Unet utilizes residual convolution and channel downsampling for feature embedding, resulting in lower computational complexity. Note that DCP-Unet incorporates multiple long skip connections between the GFB and LFB components. This helps to extract multi-scale perceptual features from the input. The extracted global and local features are concatenated along the channel dimension before being sent into the convolution layer to obtain the final output. Therefore, our DCP-Unet is specifically designed to handle the diffusion generative process and can effectively compensate for the perceptual loss in terms of pixel bitwidth and feature resolution.

TinyAE Semantic-RGB Recovery. Based on the diffusion compensation, we can convert the compensated features from semantic back to RGB space. Here, we upsample the spatial shape of the latent features as the original frames through the fast bilinear interpolation. Since the feature shape is enlarged, the final restored frames may suffer from blur and chromatic aberration. Thus, we also apply a series of dilated convolutional layers to align the texture and color signals, making the final restored frames hold a comparable perceptual quality as the original ones.

Evaluation

Experimental Setup

Hardware. We deploy the streamer and viewer on two NVIDIA Jetson AGX Orin edge devices (NVIDIA 2025b). The two edge devices are connected by a commercial switch (TP-Link TL-SE5420) with 2.5Gbps bandwidth.

Datasets. We use the open YouTube-UGC (YouTube 2025a) dataset and also construct random samples from four vision restoration datasets (each takes 25%), including BDD100K (Yu et al. 2020), VisDrone-cc2020 (Du et al. 2020), FFHQ (Karras, Laine, and Aila 2021) and DIV2K (DIV2K 2025), which are in 720p, 1080p, 1024×1024 and 2K, respectively.

NeVS Baselines. We choose the typical DeNC (Zhou et al. 2025), Gemino (Sivaraman et al. 2024), NeuroScaler (Yeo et al. 2022) and NEMO (Yeo et al. 2020) as baselines. For comparison fairness, all neural models used in baselines and DeNC++ are first trained on YouTube-UGC and then fine-tuned on the samples from four vision restoration datasets.

Performance Metrics. To inspect the neural encoder's compression efficiency, we measure the improvements of video compression ratio (CR), which can be defined as:

$$\text{CR} = \frac{\text{H.265's video size}}{\text{Other method's video size}}.$$

The core command of H.265 encoding is: `ffmpeg -c:v libx265 -crf 28 -preset medium`. Note that we

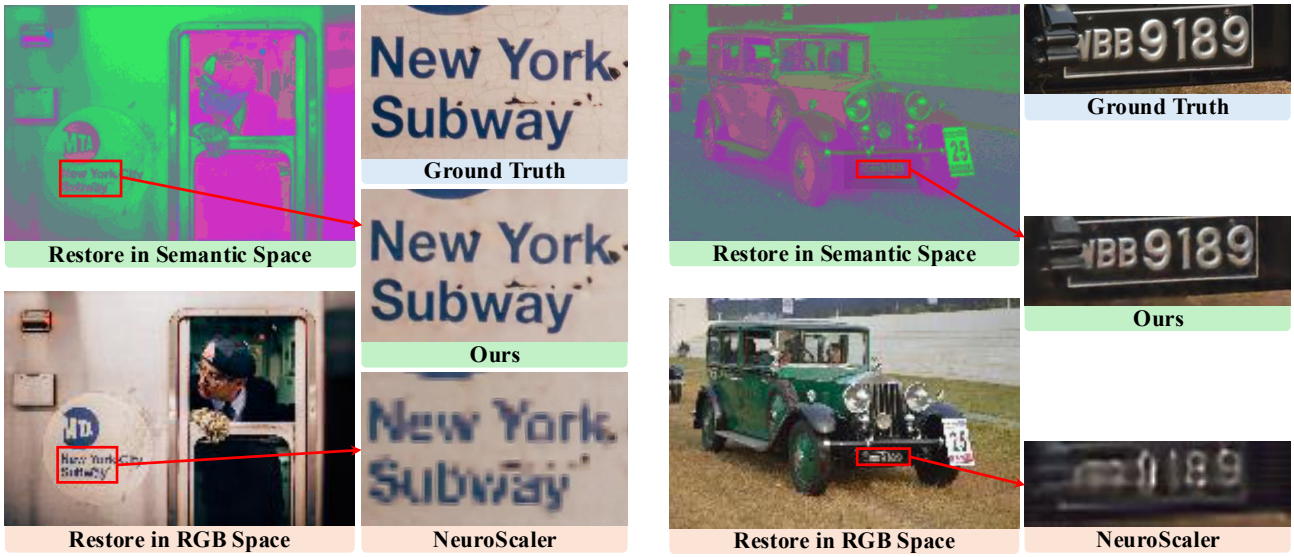

Figure 6: Comparison of restoration quality based on RGB and semantic spaces. **Best viewed in color and zoomed-in.**

omit the impacts of audio encoding. Meanwhile, to comprehensively evaluate neural decoder's restoration performance, we cover seven metrics, *i.e.,* PSNR (Horé and Ziou 2010), SSIM (Wang et al. 2004), VMAF (Netflix 2025b), FVD (Unterthiner et al. 2019), UIQI (Wang and Bovik 2002) and LPIPS (Zhang et al. 2018). We also employ the pertinent MUSIQ (Ke et al. 2021) and CLIPIQA (Wang, Chan, and Loy 2023) for non-reference assessment. The above are modern quality assessment metrics used by latest neural enhancement works (Zhou et al. 2025; Cheng et al. 2024; Chen et al. 2024). Finally, we inspect DeNC++'s compression-restoration-overhead trade-off, in terms of video compression ratios, restoration quality, parameter scale, inference speed and memory footprint.

End-to-end Performance

Traffic Bitrate Reduction. As shown in Table 1, we measure the average bitrate and corresponding compression ratio to encode a good quality video between the baselines and our DeNC++ in different video settings. In latent semantic space, the resolution scaling factors are from $2\times$ to $4\times$, which is suitable to downscale common 1080p and 720p videos. Also, the semantic features are represented from the original 32-bit floating point precision to lower bitwidth in $[4, 8]$. All the semantic features are encoded at 30 frames per second. Generally, DeNC++ can additionally reduce the encoded video size by nearly one order of magnitude compared to the pure H.265, thus achieving up to 24-55$\times$ higher compression ratios over the NeVS baselines. This makes DeNC++ qualified to handle high-definition videos.

Restoration Quality Improvement. Table 1 summarizes the parameter scales, compression ratios and restoration quality between NeVS baselines and ours under different datasets. Our DeNC++ consistently achieves superior compression-restoration-overhead trade-off over the baselines. First, DeNC++ holds a comparable model parameter

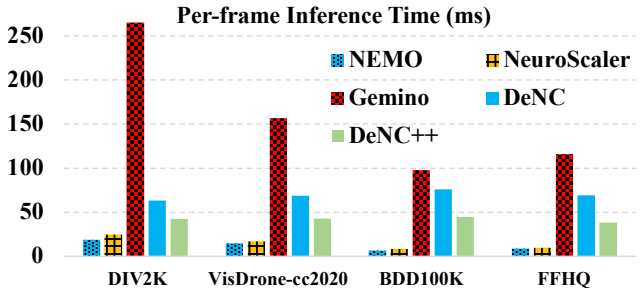

Figure 7: Comparison of inference time (ms).

scale and is much smaller than Gemino. Although DeNC++ is larger than NEMO and Neuroscaler, it still fits the memory constraints on commodity edge devices. Second, taking the video size obtained by default H.265 (H.265 2025) as the lower bound to calculate compression ratio, we can observe that DeNC++ provides a much higher compression ratio with nearly one order of magnitude over H.265, which significantly outperforms all NeVS baselines. Third, DeNC++ provides higher restoration quality (*e.g.,* the VMAF and FVD scores) over the baselines in most cases, even though DeNC++'s compression ratios are much higher.

Visualization. We zoom in on a patch of the restored frame by ours and baselines for a deep inspection. As shown in Figure 6, DeNC++ restores the visual signals close to the ground truth with high PSNR and SSIM scores. Meanwhile, we highlight the significance of conducting compression-restoration in latent semantic space. DeNC++ successfully reconstructs smooth textures and edges as the original frame, while the baseline NeuroScaler based on RGB space fails to restore the details. This visualization clearly verifies the robustness of DeNC++ to restore highly compressed videos.

Inference Process Speedups. We compare the inference time cost in Figure 7. Our DeNC++ holds a close per-frame

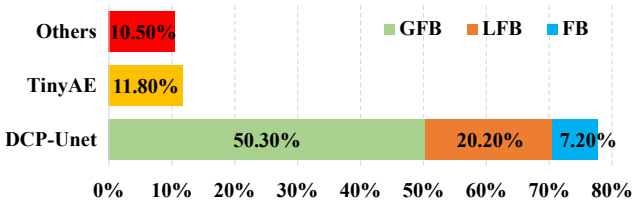

Figure 8: Inference time cost (%) of each component.

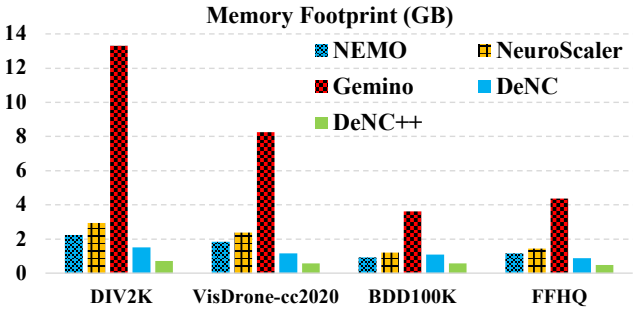

Figure 9: Comparison of memory footprint (GB).

inference time cost as NEMO and Neuroscaler, showing its feasibility of handling real-time streaming. Meanwhile, DeNC++ is over 5× faster than Gemino and restricts per-frame inference time within 50 ms, indicating that it is compatible with common edge devices.

Time Cost Analysis. We also inspect how each component impacts the inference time cost. As shown in Figure 8, we find that the DCP-Unet sampling procedure accounts for the largest proportion, approximately 77.7% (*i.e.,* 50.3% by GFB, 20.2% by LFB, and 7.2% by FB). Meanwhile, the TinyAE and other auxiliary operations account for nearly 11.8% and 10.5%, respectively. This indicates that the GFB inside the DCP-Unet dominates the overall inference time. The use of patches helps unleash the parallel computing power of edge hardware. By dividing the video frames into reasonable patch sizes (16×16 to 32×32) and stacking them along the channel dimension, we can restrict the time cost in an extremely low range, even with high frame resolutions.

Memory Footprint Saving. Commodity edge devices often suffer from strict memory constraints. DeNC++'s TinyAE and DCP-Unet diffusion models fit the memory constraints by restricting the parameter scales in 10.8M, 15.9M and 17.5M for 720p, 1080p and 2K videos, respectively. Smaller parameter scales yield a lower memory footprint. We compare the memory footprint between DeNC++ and the baselines in Figure 9. Due to the model scale controller, we can elastically adjust the memory cost for loading the parameter within 2 GB. Thus, DeNC++ requires a much lower memory footprint than the baselines. This level of memory cost is accessible for most commodity edge devices.

Compression-restoration Trade-off Improvement. The overall compression-restoration trade-off can be best understood by checking Figure 10. We measure how compression ratios impact the restoration quality in four metrics, *i.e.,* PSNR, SSIM, VMAF and FVD, by using the NeVS baselines and our DeNC++. In all cases, the restoration quality decreases with the increase of compression ratios. However, DeNC++ strikes a better balance between compression and restoration. For example, DeNC++ still preserves good quality even applying compression ratios larger than 35×, while other baselines suffer from significant quality degradation in these highly-compressed cases.

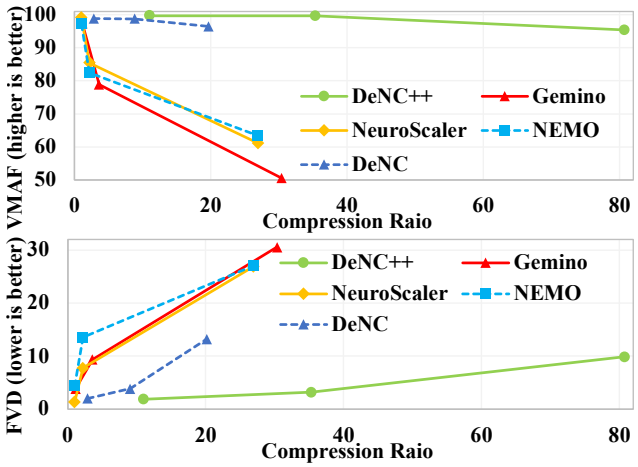

Figure 10: The trade-off between compression ratio and restoration quality based on BDD100K videos, where ours consistently outperforms the NeVS baselines.

Conclusion

This paper investigates the potential to evolve existing NeVS into a lightweight and edge-friendly manner, which natively fits the edge environments and improves end-to-end performance. Following the design philosophy of visual-semantic transformation, we present DeNC++, the first plug-and-play module that handles all the neural codec operations on clients' edge devices and removes the heavy computational dependency of media servers. DeNC++ is optimized by conducting a holistic encoder-decoder synergy, where the inherent intelligence significantly improves the compression-restoration-overhead trade-off, thus being accessible to consumer-level edge devices. Real-world evaluation verifies DeNC++'s performance superiority over the latest NeVS solutions, with 20-40× higher compression ratio and 5-7× end-to-end speedup.

Acknowledgments

This work has been partially supported by the National Natural Science Foundation of China under Grant No. U23B2026, No. 24IAA01392 and No.62372305, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2024B1515040012, Shenzhen Science and Technology Program under Grant No. JCYJ20250604181612017, No. KJZD20230923114809020 and No. RCBS20231211090523043, Scientific Foundation for Youth Scholars of Shenzhen University under Grant No. RC20240254, and Research Team Cultivation Program of Shenzhen University under Grant No. 2023QNT015.

References

Amazon. 2025. AWS Pricing Calculator. https://calculator.aws. Accessed: 2025-11-28.

AOMedia. 2025. AOMedia Video 1 Official Website. https://aomedia.org/av1/. Accessed: 2025-11-28.

Chen, B.; Yan, Z.; Zhang, Y.; Yang, Z.; and Nahrstedt, K. 2024. LiFteR: Unleash Learned Codecs in Video Streaming with Loose Frame Referencing. In Vanbever, L.; and Zhang, I., eds., *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 533–548. USENIX Association.

Cheng, Y.; Zhang, Z.; Li, H.; Arapin, A.; Zhang, Y.; Zhang, Q.; Liu, Y.; Du, K.; Zhang, X.; Yan, F. Y.; Mazumdar, A.; Feamster, N.; and Jiang, J. 2024. GRACE: Loss-Resilient Real-Time Video through Neural Codecs. In Vanbever, L.; and Zhang, I., eds., *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 509–531. USENIX Association.

Dasari, M.; Kahatapitiya, K.; Das, S. R.; Balasubramanian, A.; and Samaras, D. 2022. Swift: Adaptive Video Streaming with Layered Neural Codecs. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 103–118. USENIX Association.

DIV2K. 2025. DIV2K Dataset Official Website. https://data.vision.ee.ethz.ch/cvl/DIV2K/. Accessed: 2025-11-28.

Du, D.; Wen, L.; Zhu, P.; Fan, H.; Hu, Q.; Ling, H.; Shah, M.; Pan, J.; Al-Ali, A.; Mohamed, A.; Imene, B.; Dong, B.; Zhang, B.; Nesma, B. H.; Xu, C.; Duan, C.; Castiello, C.; Mencar, C.; Liang, D.; Krüger, F.; Vessio, G.; Castellano, G.; Wang, J.; Gao, J.; Abualsaud, K.; Ding, L.; Zhao, L.; Cianciotta, M.; Saqib, M.; Almaadeed, N.; Elharrouss, O.; Lyu, P.; Wang, Q.; Liu, S.; Qiu, S.; Pan, S.; Al-Máadeed, S.; Khan, S. D.; Khattab, T.; Han, T.; Golda, T.; Xu, W.; Bai, X.; Xu, X.; Li, X.; Zhao, Y.; Tian, Y.; Lin, Y.; Xu, Y.; Yao, Y.; Xu, Z.; Zhao, Z.; Luo, Z.; Wei, Z.; and Zhao, Z. 2020. VisDrone-CC2020: The Vision Meets Drone Crowd Counting Challenge Results. In Bartoli, A.; and Fusiello, A., eds., *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 12538 of *Lecture Notes in Computer Science*, 675–691. Springer.

Du, K.; Zhang, Q.; Arapin, A.; Wang, H.; Xia, Z.; and Jiang, J. 2022. AccMPEG: Optimizing Video Encoding for Accurate Video Analytics. In *Proceedings of the Machine Learning and Systems (MLSys)*. mlsys.org.

Farahani, R.; Amirpour, H.; Tashtarian, F.; Bentaleb, A.; Timmerer, C.; Hellwagner, H.; and Zimmermann, R. 2022. RICHTER: hybrid P2P-CDN architecture for low latency live video streaming. In *Proceedings of the Mile-High Video Conference (MHV)*, 87–88. ACM.

Google. 2025. Google VP9 Overview. https://developers.google.com/media/vp9. Accessed: 2025-11-28.

H.265. 2025. H.265 Official Website. https://www.itu.int/rec/T-REC-H.265. Accessed: 2025-11-28.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Horé, A.; and Ziou, D. 2010. Image Quality Metrics: PSNR vs. SSIM. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2366–2369. IEEE.

Karras, T.; Laine, S.; and Aila, T. 2021. A Style-Based Generator Architecture for Generative Adversarial Networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(12): 4217–4228.

Kawar, B.; Elad, M.; Ermon, S.; and Song, J. 2022. Denoising Diffusion Restoration Models. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Ke, J.; Wang, Q.; Wang, Y.; Milanfar, P.; and Yang, F. 2021. MUSIQ: Multi-scale Image Quality Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 5128–5137. IEEE.

Kirmizioglu, R. A.; and Tekalp, A. M. 2020. Multi-Party WebRTC Services Using Delay and Bandwidth Aware SDN-Assisted IP Multicasting of Scalable Video Over 5G Networks. *IEEE Trans. Multim.*, 22(4): 1005–1015.

Lugmayr, A.; Danelljan, M.; Romero, A.; Yu, F.; Timofte, R.; and Gool, L. V. 2022. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11451–11461. IEEE.

Meng, Z.; Guo, Y.; Sun, C.; Wang, B.; Sherry, J.; Liu, H. H.; and Xu, M. 2022. Achieving consistent low latency for wireless real-time communications with the shortest control loop. In Kuipers, F.; and Orda, A., eds., *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM)*, 193–206. ACM.

Microsoft. 2025a. Azure Pricing Calculator. https://azure.microsoft.com/en-us/pricing/calculator/. Accessed: 2025-11-28.

Microsoft. 2025b. Microsoft Wi-Fi Direct on Windows 10. https://learn.microsoft.com/en-us/windows-hardware/drivers/partnerapps/wi-fi-direct. Accessed: 2025-11-28.

Netflix. 2025a. Netflix Official Website. https://www.netflix.com/. Accessed: 2025-11-28.

Netflix. 2025b. Video Multi-Method Assessment Fusion. https://github.com/Netflix/vmaf. Accessed: 2025-11-28.

Nguyen, M.; Çetinkaya, E.; Hellwagner, H.; and Timmerer, C. 2022. Super-resolution based bitrate adaptation for HTTP adaptive streaming for mobile devices. In *Proceedings of the Mile-High Video Conference (MHV)*, 70–76. ACM.

NVIDIA. 2025a. NVIDIA GeForce Graphics Cards Prices. https://store.nvidia.com/en-us/geforce/store/. Accessed: 2025-11-28.

NVIDIA. 2025b. NVIDIA Jetson AGX Orin Official Website. https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/. Accessed: 2025-11-28.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10674–10685. IEEE.

Saharia, C.; Ho, J.; Chan, W.; Salimans, T.; Fleet, D. J.; and Norouzi, M. 2023. Image Super-Resolution via Iterative Refinement. *IEEE TPAMI.*, 45(4): 4713–4726.

Sivaraman, V.; Karimi, P.; Venkatapathy, V.; Shirkoohi, M. K.; Fouladi, S.; Alizadeh, M.; Durand, F.; and Sze, V. 2024. Gemino: Practical and Robust Neural Compression for Video Conferencing. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 569–590. USENIX Association.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net.

Sony. 2025. Sony Wi-Fi Direct feature on a BRAVIA TV. https://www.sony.com/electronics/support/articles/00015505. Accessed: 2025-11-28.

SopCast. 2025. SopCast Free Download Website. https://sopcast.en.softonic.com/. Accessed: 2025-11-28.

Time, P. 2025. Popcorn Software Official Website. https://github.com/popcorn-official. Accessed: 2025-11-28.

Unterthiner, T.; van Steenkiste, S.; Kurach, K.; Marinier, R.; Michalski, M.; and Gelly, S. 2019. FVD: A new Metric for Video Generation. In *Proceedings of the International Conference on Learning Representations (ICLR)*. OpenReview.net.

Wang, J.; Chan, K. C. K.; and Loy, C. C. 2023. Exploring CLIP for Assessing the Look and Feel of Images. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2555–2563. AAAI Press.

Wang, Z.; and Bovik, A. 2002. A universal image quality index. *IEEE Signal Processing Letters*, 9(3): 81–84.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612.

Ye, J.; Yeo, H.; Park, J.; and Han, D. 2023. AccelIR: Task-aware Image Compression for Accelerating Neural Restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 18216–18226. IEEE.

Yeo, H.; Chong, C. J.; Jung, Y.; Ye, J.; and Han, D. 2020. NEMO: enabling neural-enhanced video streaming on commodity mobile devices. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 28:1–28:14. ACM.

Yeo, H.; Jung, Y.; Kim, J.; Shin, J.; and Han, D. 2018. Neural Adaptive Content-aware Internet Video Delivery. In Arpaci-Dusseau, A. C.; and Voelker, G., eds., *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 645–661. USENIX Association.

Yeo, H.; Lim, H.; Kim, J.; Jung, Y.; Ye, J.; and Han, D. 2022. NeuroScaler: neural video enhancement at scale. 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 (SIGCOMM)*, 795–811. ACM.

YouTube. 2025a. UGC Dataset. https://media.withyoutube.com/. Accessed: 2025-11-28.

YouTube. 2025b. YouTube Live Streaming Official Website. https://www.youtube.com/live. Accessed: 2025-11-28.

Yu, F.; Chen, H.; Wang, X.; Xian, W.; Chen, Y.; Liu, F.; Madhavan, V.; and Darrell, T. 2020. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2633–2642. Computer Vision Foundation / IEEE.

Zhang, A.; Wang, C.; Han, B.; and Qian, F. 2022. YuZu: Neural-Enhanced Volumetric Video Streaming. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 137–154. USENIX Association.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–595. Computer Vision Foundation / IEEE Computer Society.

Zhou, Q.; Guo, S.; Liu, Y.; Zhang, J.; Zhang, J.; Guo, T.; Xu, Z.; Liu, X.; and Qu, Z. 2022. Hierarchical Channel-spatial Encoding for Communication-efficient Collaborative Learning. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Zhou, Q.; Li, R.; Guo, J.; Huang, Y.; Xu, Z.; Cui, L.; and Guo, S. 2025. DeNC: Unleash Neural Codecs in Video Streaming with Diffusion Enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 1192–1200. AAAI Press.

Zoom. 2025. Zoom Meeting Official Website. https://zoom.us/. Accessed: 2025-11-28.