ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11462603

ENHANCED VIDEO COMPRESSION WITH CONTEXT-AWARE DYNAMIC NEURAL ADAPTER

Shaofan Sun[1], Shuangming Ma[2], Han Chen[2], Ling-Yu Duan[1], Jiaying Liu[1,]*
[1]Peking University, [2]Beijing Connected and Autonomous Vehicles Technology Co., Ltd
carefree_sun@stu.pku.edu.cn, {mashuangming,chenhan}@bcavt.com, {lingyu,liujiaying}@pku.edu.cn

ABSTRACT

The growing demand for video data to simultaneously serve human and machine visual analysis brings significant challenges to compression design. Traditional codecs are primarily tailored for pixel-level reconstruction, offering high compression efficiency but limited adaptability to diverse downstream tasks. With the emergence of end-to-end learned compression, it becomes possible to tailor representations for different purposes. However, such methods usually suffer from high complexity and incompatibility with existing compression standards, which limits their deployment. To address these issues and meet the requirements of diverse application scenarios, we propose for the first time a **C**ontext-**A**ware dynamic neural adapter for enhanced **V**ideo **C**ompression (**CAVC**) that incorporates a context mode injection mechanism, allowing a single model to dynamically adjust to different compression requirements via numerical context modes. This method adds only a small number of learnable parameters to enable diverse adaptation, while maintaining compatibility with standard codecs. Experimental results on UVG and TUMTraf datasets demonstrate that our approach enhances compression performance for visual perception, pixel-level reconstruction, and machine recognition.

***Index Terms*—** Video Compression, Deep Neural Networks, Task-Specific Adaptation

1. INTRODUCTION

With the widespread use of intelligent devices, analysis for machine analytics [1, 2] of high-quality video data has become an increasingly common and essential demand, in addition to the traditional requirement of delivering content optimized for signal fidelity preservation (*e.g.*, high PSNR) or human perception (*e.g.*, low LPIPS [3]). In practical deployments, edge devices collect massive data in diverse environments and transmit them to the cloud for diverse tasks. This raises new challenges for video compression, requiring efficient bitrate reduction and generalization across diverse conditions to preserve downstream performance. While modern video coding standards, such as Advanced Video Coding (AVC) [4], High Efficiency Video Coding (HEVC) [5], and Versatile Video Coding (VVC) [6], achieve remarkable compression efficiency through complex handcrafted tools, their capacity to adapt signal distributions to novel data conditions and new tasks is limited.

Recent advances in end-to-end learned image and video compression [7, 8, 9, 10, 11] have brought higher coding efficiency and greater flexibility, as priors for compression can be directly learned from data. However, such methods suffer from high computational complexity; and despite their adaptability, most existing approaches rely on a single model with a fixed set of parameters and preferences, limiting flexibility for diverse real-world applications. Another line of research explores using Deep Neural Networks (DNNs) for adaptation. Instead of building a full codec, some works [12, 13, 14] employ lightweight DNNs as pre- and post-processors, *i.e.*, mapping inputs with different pixel distributions into spaces that can be more efficiently compressed, or enhancing the compressed outputs for specific tasks. Although these approaches enable low-cost training for adaptation, they are generally limited to particular scenarios and cannot effectively handle diverse signal distributions and tasks.

The key issue lies in the **diversity of signal distributions**, which poses challenges for transform and probability modeling. If **input/output distributions of signals can be decoupled from a core constant compression process**, it becomes possible to not only maintain high compression efficiency but also achieve flexible adaptation across different scenarios and tasks.

To this end, we propose **C**ontext-**A**ware dynamic neural adapter for enhanced **V**ideo **C**ompression (**CAVC**) in more flexible coding manners. This method incorporates requirement information to guide the adaptation process for different purposes. With additional prompts named **context mode**, the input/output signal distributions are decoupled from the core compression procedure, and CAVC is enabled to not only preserve high coding efficiency but also obtain a single model to adapt dynamically to diverse scenarios and downstream tasks. Specifically, we introduce a novel context mode injection mechanism that enables DNNs to dynamically adapt their behavior in response to those modes. These modes indicate different compression requirements and are represented by numerical identifiers for efficient transmission. For the pre-processor network, the parameters are identical for all

*Corresponding author.

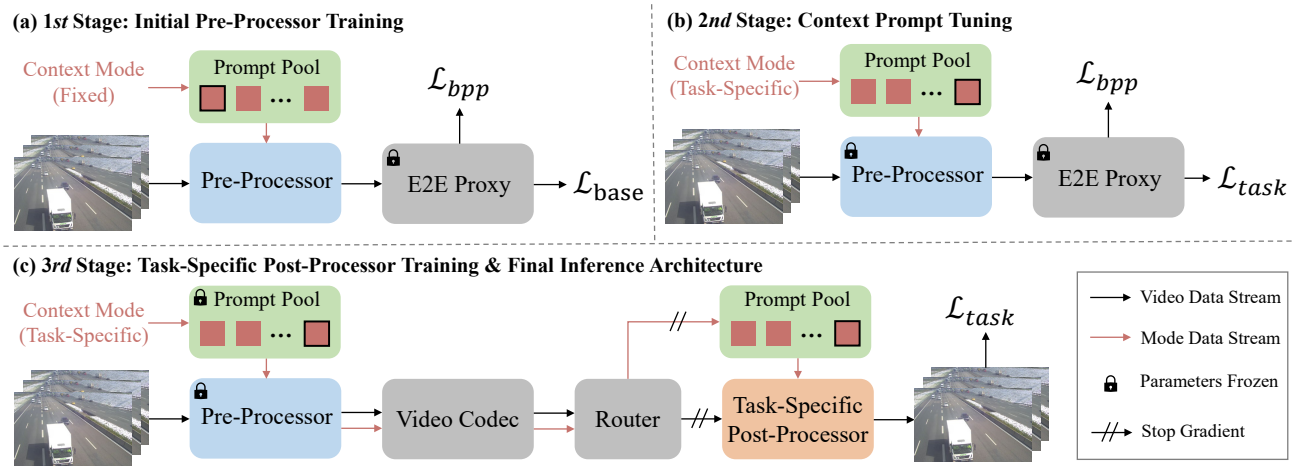

Fig. 1. Overall architecture of **C**ontext-**A**ware dynamic neural adapter for enhanced **V**ideo **C**ompression (**CAVC**) (c) and the proposed progressive training strategy (a-c).

modes, with a prompt feature chosen from the learned prompt pool according to the context mode and fused with the intermediate feature maps. For the post-processor, different parameters are obtained by training with loss constraints associated with the corresponding tasks, with context mode injection similar to the pre-processor. In our design, only a small increase in learnable parameters is required, enabling the model to support edge devices in adapting to a broader range of scenarios and tasks while remaining compatible with standard codecs. Experiments demonstrate that our method effectively enhances the performance of standard codecs for human visual perception, pixel-level reconstruction, and machine recognition, as validated on the UVG [15] and TUM-Traf [16] video datasets.

2. CONTEXT-AWARE DYNAMIC NEURAL ADAPTER FOR ENHANCED VIDEO COMPRESSION

The overall architecture of CAVC and the progressive training strategy are illustrated in Fig. 1 (c) and Figs. 1 (a-c), respectively, which we will introduce in detail next. This framework employs *context-aware pre-processing* to adapt standard video codecs to varying tasks, utilizing context modes for efficient video compression. The architecture includes a *progressive training strategy* to optimize performance across different video compression scenarios.

2.1. Context-Aware Pre-Post-Processing

In this part, we introduce the pipeline of performing context-aware pre-post-processing to adapt standard video codecs to different scenarios, controlled by context mode. Formally, given a video $\mathcal{V} = \{I_t \in \mathbb{R}^{H\times W\times C}\}_{t=1}^{T}$ composed of T frames, each with a resolution of $H \times W$ and C channels, and a context mode $m \in \{1, 2, \cdots, M\}$, where M is the preset number of modes, our goal is to obtain the compressed video adapted for the task corresponding to mode m.

To inject the context mode information into $\mathcal{V}$, we embed both the video frames and the context mode into the feature space and fuse them. Specifically, we adopt a U-Net [17] as the pre-processor to embed and reconstruct video frames into features. We use the first few layers of the U-Net, denoted as $\mathcal{F}_{emb}$, to embed the t-th frame I_t into the feature:

$$f_t = \mathcal{F}_{emb}\left(I_t\right) \in \mathbb{R}^{h\times w\times d}, \tag{1}$$

where h, w, and d represent the height, width, and embedding dimension of the feature map, and then perform reconstruction with the remaining layers, denoted as $\mathcal{F}_{rec}$. For the context mode m, we maintain a prompt pool denoted as $\mathrm{P}_{pre} = \{p_1, p_2, \cdots, p_M\}$ and select specific context prompt feature $p_m \in \mathbb{R}^{d_p}$ based on the mode m, where d_p is the dimension for each prompt feature. Then p_m is fused with f_t by a fusion operation $\mathcal{F}_{fuse}(\cdot,\cdot)$ and an 1×1 convolution layer $\mathcal{F}_{conv}(\cdot)$ to project it to d-dimension feature:

$$\widetilde{f}_t = \mathcal{F}_{conv}\left(\mathcal{F}_{fuse}(f_t, p_m)\right) \in \mathbb{R}^{h\times w\times d}. \tag{2}$$

$\mathcal{F}_{fuse}(\cdot,\cdot)$ can be implemented by strategies such as concatenation or AdaIN [18]. The pre-processed video frame is reconstructed with $\mathcal{F}_{rec}$:

$$B_t = \mathcal{F}_{rec}(\widetilde{f}_t). \tag{3}$$

The reconstructed video is compressed with video codecs, and the t-th compressed frame is denoted as $\widehat{B_t}$. Moreover, in practice, the context mode m can be embedded into the bit stream of video and transmitted to the decoder side using Supplemental Enhancement Information (SEI) defined in AVC and HEVC standards. After the decoder side obtains $\widehat{B_t}$ and m, a router is applied, to determine which task-specific post-processor to use based on the context mode. Then the post-processor and the corresponding prompt pool P_{post} performs similarly to pre-processing, which takes $\widehat{B_t}$ and m as inputs and reconstructs frame $\widehat{I_t}$ targeted at the specific task related to context mode m.

2.2. Progressive Training Strategy

To ensure that our framework is fully optimized across diverse scenarios, we design a three-stage progressive training strategy.

1*st* Stage: Initial Pre-Processor Training. To initially obtain a general-purpose pre-processor, we first choose a fixed mode and employ the rate loss and the distortion loss between the compressed and the original video frames to ensure that the pre-processor preserves the essential information in the videos. To optimize the pre-processor and the prompt pool with a differentiable loss function, we use a pre-trained learned compression model as the codec proxy. Specifically, we employ the TCM [19] in this work, which can be replaced with more advanced models if needed, ensuring good compatibility with the development of learned image/video compression technologies.

We use the combination of rate loss, Mean Squared Error (MSE) loss, and the LPIPS Perceptual loss to optimize the pre-processor, balancing the visual perception and signal fidelity. Formally, the overall loss function $\mathcal{L}_{stage1}$ is formulated as:

$$\mathcal{L}_{mse} = \frac{1}{T \times H \times W \times C} \sum_{t=1}^{T} \|I_t - \widehat{B_t}\|_2^2, \tag{4}$$

$$\mathcal{L}_{lpips} = \frac{1}{T} \sum_{t=1}^{T} \mathrm{LPIPS}(I_t, \widehat{B_t}), \tag{5}$$

$$\mathcal{L}_{base} = \lambda_{mse}\mathcal{L}_{mse} + \lambda_{lpips}\mathcal{L}_{lpips}, \tag{6}$$

$$\mathcal{L}_{stage1} = \lambda_{bpp}\mathcal{L}_{bpp} + \mathcal{L}_{base}, \tag{7}$$

where $\mathcal{L}_{bpp}$ denotes the bit rate cost following the calculation method in [19], and λ_{bpp}, λ_{mse}, λ_{lpips} are the weights for respective losses.

2*nd* Stage: Context Prompt Tuning. To adapt the pre-processor to different requirements, we tune the context prompts in the prompt pool P_{pre} with different context mode inputs and corresponding task-specific losses. During training, the parameters of the pre-processor are frozen, and the fixed prompt trained at the 1*st* stage is broadcast to initialize all other prompts. The loss function $\mathcal{L}_{stage2}$ is formulated as:

$$\mathcal{L}_{stage2} = \lambda_{bpp}\mathcal{L}_{bpp} + \mathcal{L}_{task}, \tag{8}$$

where $\mathcal{L}_{task}$ is determined by the task, which we will introduce the details in Sec. 3.

3*rd* Stage: Task-Specific Post-Processor Training. We train post-processors and context prompts in P_{post} with task-specific loss functions to perform targeted adaptation. At this stage, the pre-processor and prompt pool P_{pre} have been well trained and are frozen. The TCM proxy is replaced by a standard video codec, *e.g.*, HEVC, to generate real compressed videos with various bit rates as the inputs for training the post-processors effectively. Apart from having no bitrate loss, other terms in the loss function for each task remains consistent with those at the 2*nd* stage.

3. EXPERIMENTS

Datasets and Evaluation. We adopt two video datasets UVG [15] and TUMTraf [16], to validate the effectiveness of our CAVC framework. For each video in both datasets, we randomly crop 20% of consecutive frames as the test data, and the remaining 80% is used for training. For the high-quality UVG dataset, we evaluate human visual perception performance using LPIPS [3] with AlexNet and assess signal fidelity using PSNR on the RGB channels. We downsample the videos to a resolution of 1920×1080 for more efficient training and testing. For the TUMTraf dataset with complex traffic scenarios in 1920×1200-resolution videos, we test the detection performance of average precision (AP) with a pre-trained RetinaNet [20] as the detection head. We use the R1_S3 subset, which is labeled with 3D bounding boxes transformed into 2D when testing.

Implementation Details. All our models are trained on an NVIDIA GeForce RTX 4090 GPU with a batch size of 8. Adam optimizer is adopted with a learning rate of 10^{-4}. Frames are randomly cropped to 256×256-resolution. The hyperparameters of U-Nets follow [13], and we fuse the prompt with the feature map output by the second upsampling layer. The prompt dimension d_p is set to 128. By default, we employ concatenation to inject the prompt features. At the 1*st* stage, we train for 50 epochs on the combination of UVG and TUMTraf training data. λ_{lpips} and λ_{mse} are set to 0.2 and 1.0, and we choose the mode related to LPIPS. At the 2*nd* stage, we tune the prompts for PSNR and detection for 20 epochs each. For PSNR, we train the model on UVG with MSE loss with a weight of 20. For detection, we add the loss following RetinaNet [20] with a weight of 30 to $\mathcal{L}_{base}$, and train on TUMTraf. At the 3*rd* stage, we train for 50 epochs. The $\mathcal{L}_{task}$ for LPIPS is the same as $\mathcal{L}_{base}$, and the weights for PSNR and detection are consistent with those in the 2*nd* stage. λ_{bpp} is set to 50 at both the 1*st* and 2*nd* stages.

Quantitative Evaluation. We adopt our CAVC framework to standard codecs of AVC and HEVC to verify the improvements. Moreover, we compare our method with the basic pre-post method, Sandwich [13], and a learned video compression model, DCVC [9]. Fig. 2-Fig. 4 show the rate-distortion performance across LPIPS, PSNR, and detection AP. For the LPIPS metric that we initially optimize at the 1*st* stage, our CAVC significantly improves the performance of standard codecs, demonstrating the adaptation capability of the pre- and post-processors. Meanwhile, for PSNR and detection AP whose prompts for the pre-processor are tuned at the 2*nd* stage, CAVC still achieves notable improvements, which indicates that CAVC can effectively adapt to different scenarios through context injection.

Since the Sandwich model and DCVC cannot adapt to different tasks flexibly, they tend to be optimized for specific

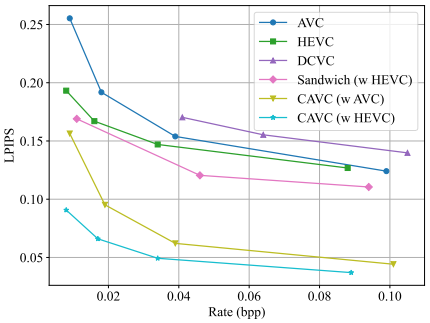

Fig. 2. LPIPS results on *UVG*.

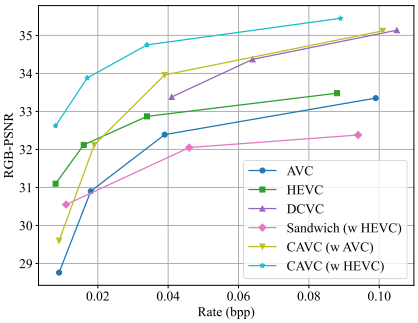

Fig. 3. PSNR results on *UVG*.

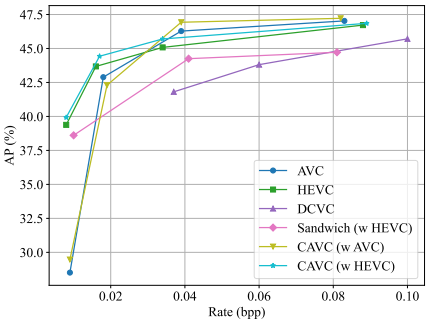

Fig. 4. Detection results on *TUMTraf*.

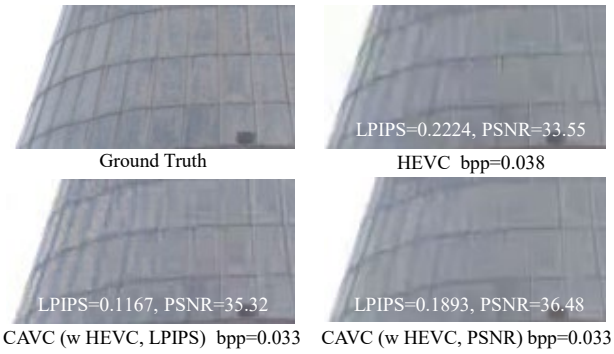

Fig. 5. Visual comparisons on *CityAlley* in *UVG*.

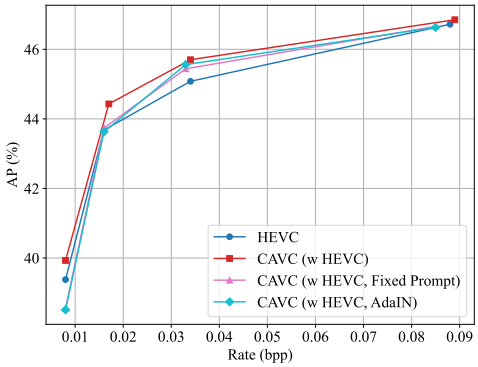

Fig. 6. Ablation study for object detection on *TUMTraf*.

metrics compared to standard codecs, *i.e.*, Sandwich achieves better LPIPS and DCVC achieves high PSNR. However, they perform relatively poorly in other scenarios, especially object detection, which is a machine vision task and is often overlooked in codec design and optimization. Nevertheless, our dynamic adapters can bridge this gap and achieve comprehensive performance improvements. Moreover, experiments show that CAVC (w HEVC) reduces FLOPs by 49.9% compared to DCVC, and it adapts to diverse scenarios with only 1.3% additional parameters for the pre-processor overhead per task, demonstrating its efficiency.

Visual Comparisons. We provide an example for visualization in Fig. 5. CAVC achieves superior LPIPS and PSNR metrics at lower bitrates compared to using HEVC alone. Specifically, setting the context mode to LPIPS results in videos that retain more textures. On the other hand, the PSNR mode yields smoother visuals with superior line continuity.

Ablation Study. We conduct ablation studies for object detection on TUMTraf to verify rationality of module designs.

a) Context Prompt Tuning. To reveal the role of context prompts in the framework, we freeze the prompt pools of the pre- and post-processors after completing the 1*st* stage training oriented to LPIPS optimization. As shown in Fig. 6, the *fixed prompt* version performs relatively poorly compared to the framework with dynamic prompts. When videos are compressed to extremely low bitrates, *e.g.*, bpp < 0.01, the object detection performance using fixed prompts even falls below that of directly using the standard HEVC codec. This demonstrates the importance of employing adaptable prompts to regulate the distribution of video signals.

b) Context Injection Strategy. Besides concatenation, we attempt to use AdaIN for prompt feature injection. However, the detection performance degrades as shown in Fig. 6. Since AdaIN regulates the channel-wise overall distribution (mean and variance) of feature maps but lacks element-level fine-grained control, it fails to effectively adapt the video pixel distribution to different tasks. Although concatenation is simpler, it achieves more precise signal regulation by fusing prompt information with feature maps in an element-wise manner, achieving higher detection precision.

4. CONCLUSION

We propose a context-aware dynamic neural adapter for an enhanced video compression framework, which enhances standard video codecs for both human perception and machine analysis. By integrating context injection with dynamic neural adapters, CAVC allows a single model to adapt to various compression goals, including pixel-level reconstruction and machine recognition. Compatible with existing codecs, *e.g.* AVC and HEVC, the framework is suitable for edge device deployment. Our experiments show that CAVC delivers outstanding performance across all targeted metrics.

5. ACKNOWLEDGEMENT

This work was supported in part by the Beijing Major Science and Technology Project under Contract no. Z251100008425023.

6. REFERENCES

[1] Bo Gu, Junhui Zhan, Shimin Gong, Wanquan Liu, Zhou Su, and Mohsen Guizani, "A spatial-temporal transformer network for city-level cellular traffic analysis and prediction," *IEEE Transactions on Wireless Communications*, vol. 22, no. 12, pp. 9412–9423, 2023.

[2] Xinmei Huang and Sheng Zhang, "Human activity recognition based on transformer in smart home," in *Asia Conference on Algorithms, Computing and Machine Learning*, 2023.

[3] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[4] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra, "Overview of the H. 264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[5] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[6] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm, "Overview of the Versatile Video Coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[7] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston, "Variational image compression with a scale hyperprior," in *International Conference on Learning Representations*, 2018.

[8] Yueyu Hu, Wenhan Yang, Zhan Ma, and Jiaying Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4194–4211, 2021.

[9] Jiahao Li, Bin Li, and Yan Lu, "Deep contextual video compression," in *Advances in Neural Information Processing Systems*, 2021.

[10] Fabian Mentzer, George D Toderici, David Minnen, Sergi Caelles, Sung Jin Hwang, Mario Lucic, and Eirikur Agustsson, "VCT: A video compression transformer," in *Advances in Neural Information Processing Systems*, 2022.

[11] Jinxi Xiang, Kuan Tian, and Jun Zhang, "MIMT: Masked image modeling transformer for video compression," in *International Conference on Learning Representations*, 2022.

[12] Guo Lu, Xingtong Ge, Tianxiong Zhong, Qiang Hu, and Jing Geng, "Preprocessing enhanced image compression for machine vision," *IEEE Transactions on Circuits and Systems for Video Technology*, 2024.

[13] Onur G Guleryuz, Philip A Chou, Berivan Isik, Hugues Hoppe, Danhang Tang, Ruofei Du, Jonathan Taylor, Philip Davidson, and Sean Fanello, "Sandwiched compression: Repurposing standard codecs with neural network wrappers," *arXiv preprint arXiv:2402.05887*, 2024.

[14] Yueyu Hu, Chenhao Zhang, Onur G Guleryuz, Debargha Mukherjee, and Yao Wang, "Standard compliant video coding using low complexity, switchable neural wrappers," in *IEEE International Conference on Image Processing*, 2024.

[15] Alexandre Mercat, Marko Viitanen, and Jarno Vanne, "UVG dataset: 50/120fps 4k sequences for video codec analysis and development," in *ACM Multimedia Systems Conference*, 2020.

[16] Christian Creß, Walter Zimmer, Leah Strand, Maximilian Fortkord, Siyi Dai, Venkatnarayanan Lakshminarasimhan, and Alois Knoll, "A9-dataset: Multi-sensor infrastructure-based dataset for mobility research," in *IEEE Intelligent Vehicles Symposium*, 2022.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[18] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *IEEE International Conference on Computer Vision*, 2017.

[19] Jinming Liu, Heming Sun, and Jiro Katto, "Learned image compression with mixed transformer-CNN architectures," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," in *IEEE International Conference on Computer Vision*, 2017.