Generative Video Compression with One-Dimensional Latent Representation

Supplementary Material

1. Training Details

This section provides the training details of our model. We train the model on the Vimeo [4, 21] and OpenVid-HD [15] datasets. The OpenVid-HD dataset contains 433,523 videos. For the Vimeo dataset, in addition to the 7-frame sequences in Vimeo-90k, we follow DCVC-FM [12] to process the raw Vimeo videos and generate approximately 6,000 additional 32-frame sequences. Specifically, we use the original Vimeo videos [4], apply scene detection, and randomly sample 32-frame clips. Our training strategy is as follows. The model is optimized using the AdamW optimizer [13] throughout the entire training process.

The first stage. We train only the encoder and decoder using two-frame samples from the OpenVid-HD dataset, resized to 256×256, with a batch size of 128. We adopt the reconstruction loss function from TA-Titok [11]:

$$\begin{aligned} L_{\text{stage1}} =& \|x - \hat{x}\|_2 + \\ & \lambda_{\text{LPIPS}} L_{\text{LPIPS}}(x, \hat{x}) + \\ & \lambda_{\text{adv}} L_{\text{adv}}(x, \hat{x}). \end{aligned} \tag{1}$$

Here, L_{LPIPS} denotes the perceptual loss, and L_{adv} denotes the adversarial loss [7]. The corresponding weighting coefficients, λ_{LPIPS} and λ_{adv}, are set to 1.1 and 0.1, respectively, following TA-Titok. Consistent with TA-Titok, we use a maximum learning rate of $1\text{e}{-4}$ and apply a cosine learning rate schedule. Both the encoder and decoder are initialized with TA-Titok pretrained weights, which provide a strong prior and facilitate faster convergence.

The second stage. We introduce the entropy model and train the network on resized 256×256 videos from the Vimeo dataset. Regarding the AR entropy model, we adopt a standard Transformer for token-by-token probability prediction. As for the non-autoregressive version used for ablation experiments, we adopt the two-step entropy model from DCVC-RT [9]. The number of training frames progressively increases from 2 to 32, while the batch size decreases from 128 to 4 according to the training frames. The learning rate is set to $5\text{e}{-5}$. The loss function is defined as:

$$L_{\text{stage2}} = \frac{1}{T} \sum_{t=1}^{t=T} (R + \lambda L_{\text{stage1}}), \tag{2}$$

where R denotes the bitrate and T denotes the number of frames used in cascade training. The trade-off parameter λ is sampled at eight points using logarithmic-space linear interpolation within the range $[0.07, 1.5]$.

The third stage. We sequentially introduce the global Transformer and 1D memory, and finetune the model on OpenVid-HD videos with variable resolutions ranging from 256×256 to 1280×2048. The training sequence length is set to 32 frames with a batch size of 2. The learning rate is set to $1\text{e}{-5}$. To reduce GPU memory consumption, following ECVC [10], we employ the partial cascaded finetuning strategy with a frame group size of 4. The loss function remains the same as in the second stage.

2. Experiments

This section describes the evaluation settings and presents additional quantitative and qualitative results.

2.1. Evaluation Details

We evaluate all models on the 96-frame videos from the HEVC-B [6], UVG [14], and MCL-JCV [18] datasets. All videos are processed so that both their height and width are multiples of 256.

For traditional video codecs, we evaluate VTM-17.0 [3], HM-16.25 [2], and ECM-5.0 [1] in the RGB colorspace, following the protocol used in the DCVC-FM [12], which converts the RGB input to 10-bit YUV444 for internal codec processing and adopts the low-delay encoding setting. We use the official configuration files for VTM, HM, and ECM, respectively: *encoder_lowdelay_vtm.cfg*, *encoder_lowdelay_main_rext.cfg*, and *encoder_lowdelay_ecm.cfg*. The parameters for encoding are as:

- -c {*config file name*}
  --InputFile={*input file name*}
  --InputBitDepth=10
  --OutputBitDepth=10
  --OutputBitDepthC=10
  --InputChromaFormat=444
  --FrameRate={*frame rate*}
  --DecodingRefreshType=2
  --FramesToBeEncoded={*frame number*}
  --SourceWidth={*width*}
  --SourceHeight={*height*}
  --IntraPeriod={*intra period*}
  --QP={*qp*}
  --Level=6.2
  --BitstreamFile={*bitstream file name*}

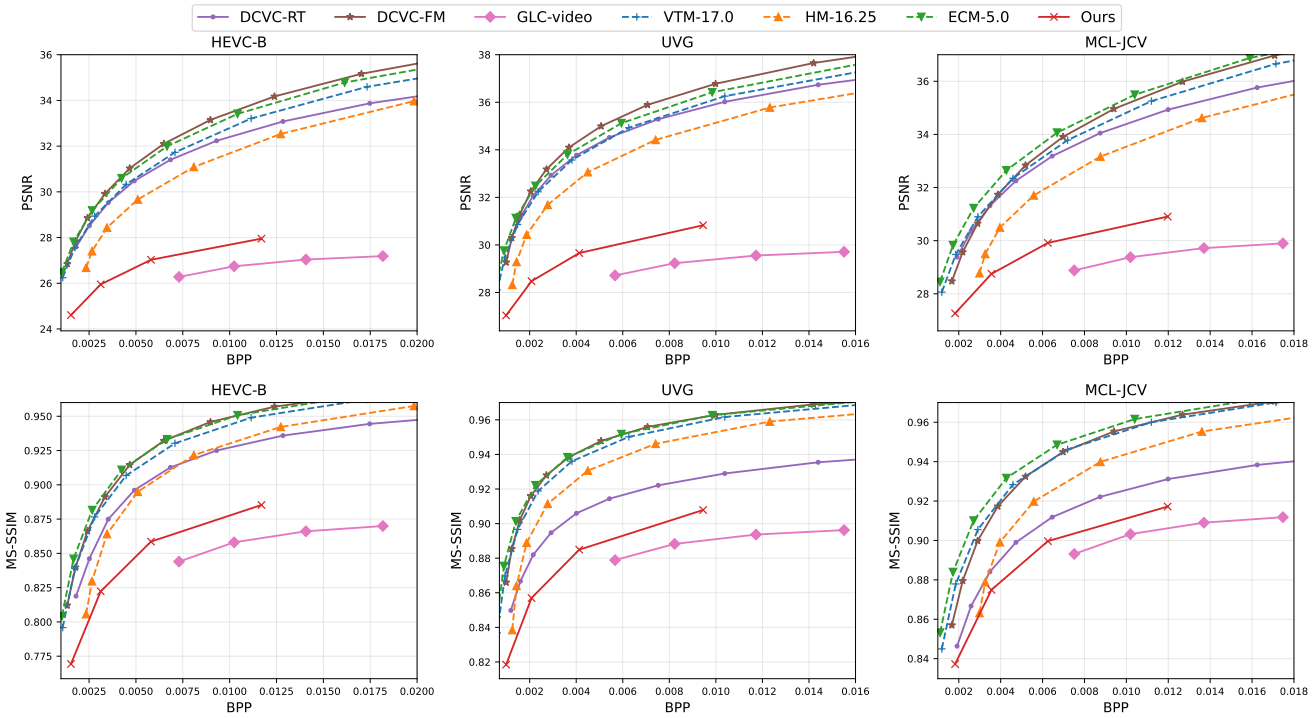

Figure 1. Rate-distortion curves in terms of PSNR and MS-SSIM.

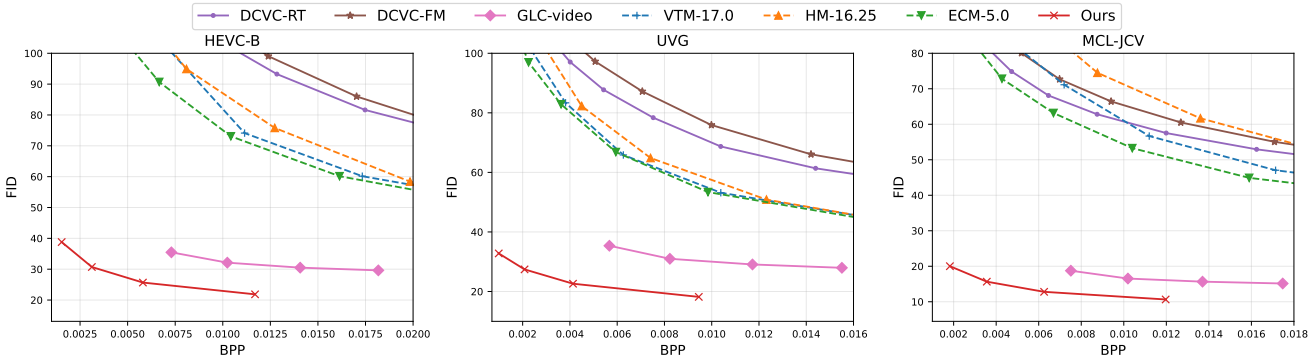

Figure 2. Rate-distortion curves in terms of FID.

For previous neural video codecs [9, 12, 16], and our method, we adopt low-delay encoding settings (intra-period = –1) in the RGB colorspace.

2.2. Quantitative Results

We report traditional pixel-level distortion metrics, PSNR and MS-SSIM [19] results in Fig. 1. Compared with GLC-video [16], which is similarly optimized for perceptual quality, our method not only achieves better perceptual metrics (LPIPS and DISTS) but also attains higher objective metrics (PSNR and MS-SSIM). These results confirm the effectiveness of our approach for perceptual video compression. Although our method produces lower PSNR than objective video codecs [1, 12], both the visual comparisons and the perceptual metrics presented in the main text consistently demonstrate its superior perceptual performance. Moreover, at such low bitrates, objective codecs (typically optimized for PSNR) tend to lose fine details, resulting in blurry reconstructions. Therefore, higher PSNR or MS-SSIM at these bitrates does not necessarily correspond to better perceptual quality.

We further report the FID [8] metric in Fig. 2 and KID [17] metric in Fig. 3. The FID and KID metrics are evaluated by splitting the frames into overlapped 256×256 patches, following the method in [20]. FID and KID mea-

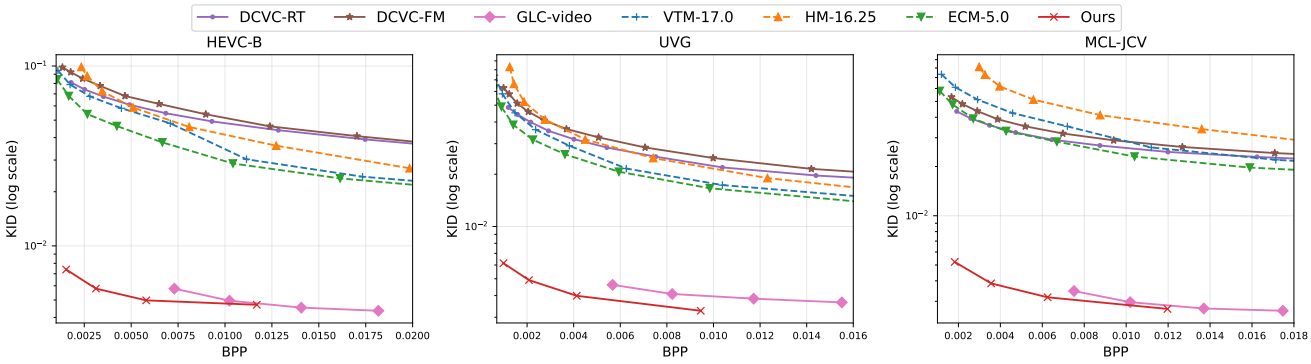

Figure 3. Rate-distortion curves in terms of KID.

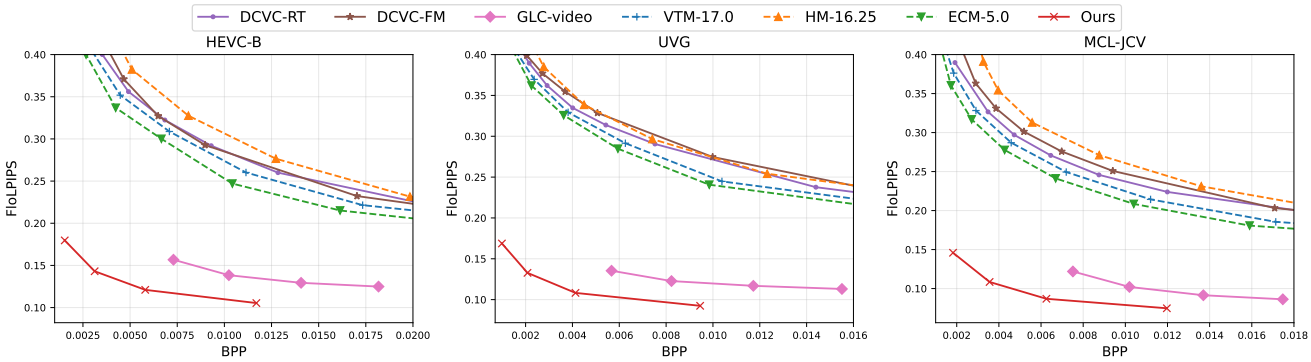

Figure 4. Rate-distortion curves in terms of FloLPIPS.

sure the distribution-level similarity between reconstructed and ground-truth frames, thus providing complementary insights beyond pixel- or feature-level metrics. As shown in the figures, our method consistently outperforms all competing approaches across both metrics, indicating that our reconstructions better align with the statistical characteristics of natural video distributions and exhibit superior overall fidelity.

2.3. Qualitative Results

In this section, we present additional qualitative results from the HEVC-B, UVG, and MCL-JCV datasets (Fig. 5). Our method produces visually sharper and more realistic reconstructions compared with other baselines. These examples further demonstrate that our approach excels not only in quantitative metrics but also in preserving fine structures, color consistency, and perceptual naturalness, thereby delivering superior visual quality.

3. Temporal Consistency Evaluation

In this section, we test the temporal consistency of our method through both quantitative and qualitative evaluation.

For quantitative results, we report the FloLPIPS [5] metric in Fig. 4. FloLPIPS is a flow-guided variant of LPIPS designed to measure temporal perceptual consistency in videos. As shown in the figure, our method consistently achieves lower FloLPIPS scores across all datasets, indicating better temporal coherence and reduced flickering artifacts. This further validates that the proposed method not only enhances perceptual quality but also preserves inter-frame consistency, which is crucial for producing visually stable video results.

For the qualitative evaluation, as illustrated in Fig. 6, our model preserves the structure of the railing while producing clear and temporally consistent results. In contrast, DCVC-FM and ECM-5.0 produce blurrier reconstructions with noticeable temporal instability, while GLC-video exhibits prominent temporal artifacts across frames. These observations further confirm the ability of our method to generate temporally consistent videos.

References

[1] ECM. https://vcgit.hhi.fraunhofer.de/ecm/ECM. 1, 2

[2] HM. https://vcgit.hhi.fraunhofer.de/jvet/HM. 1

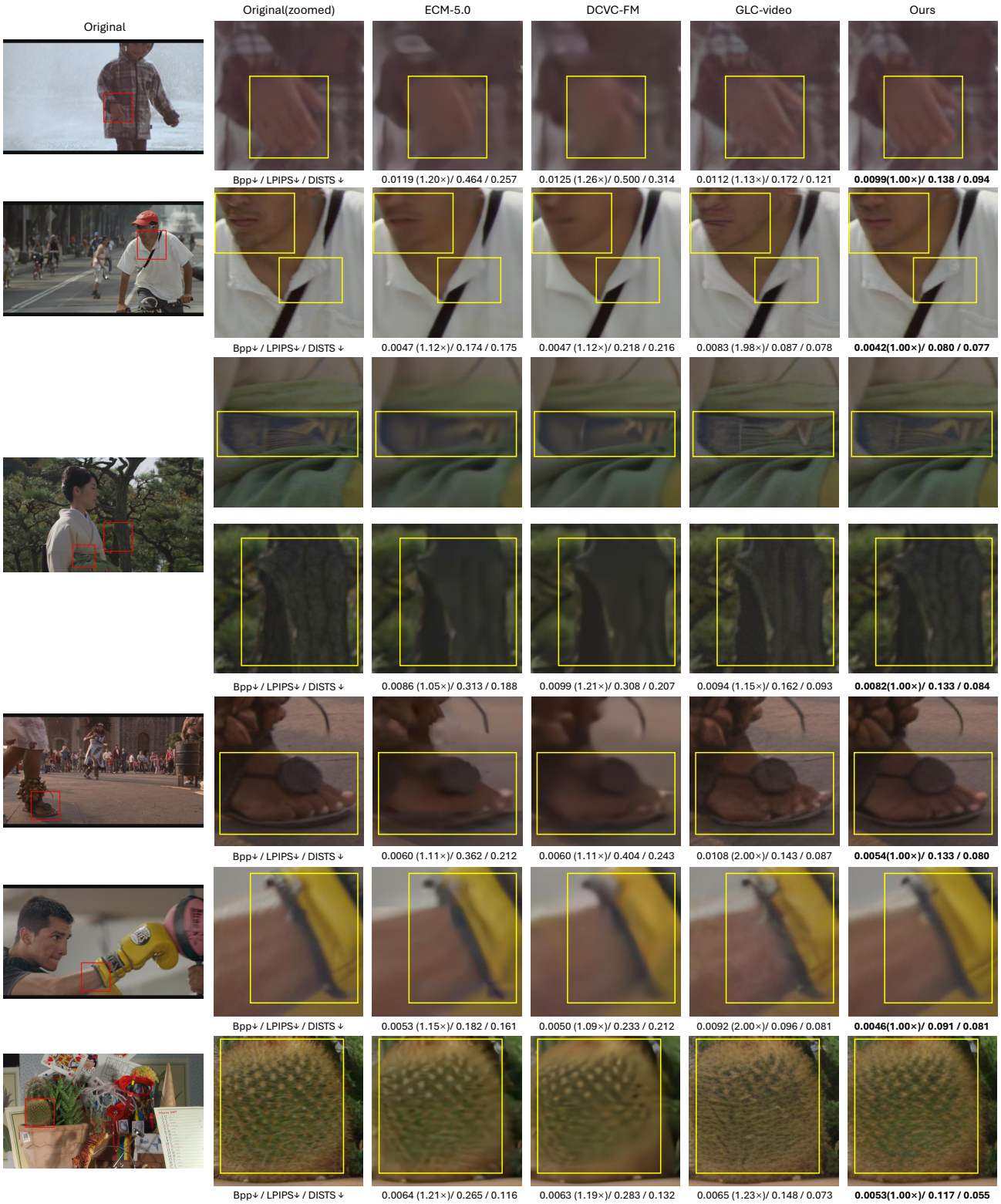

Figure 5. More visual examples.

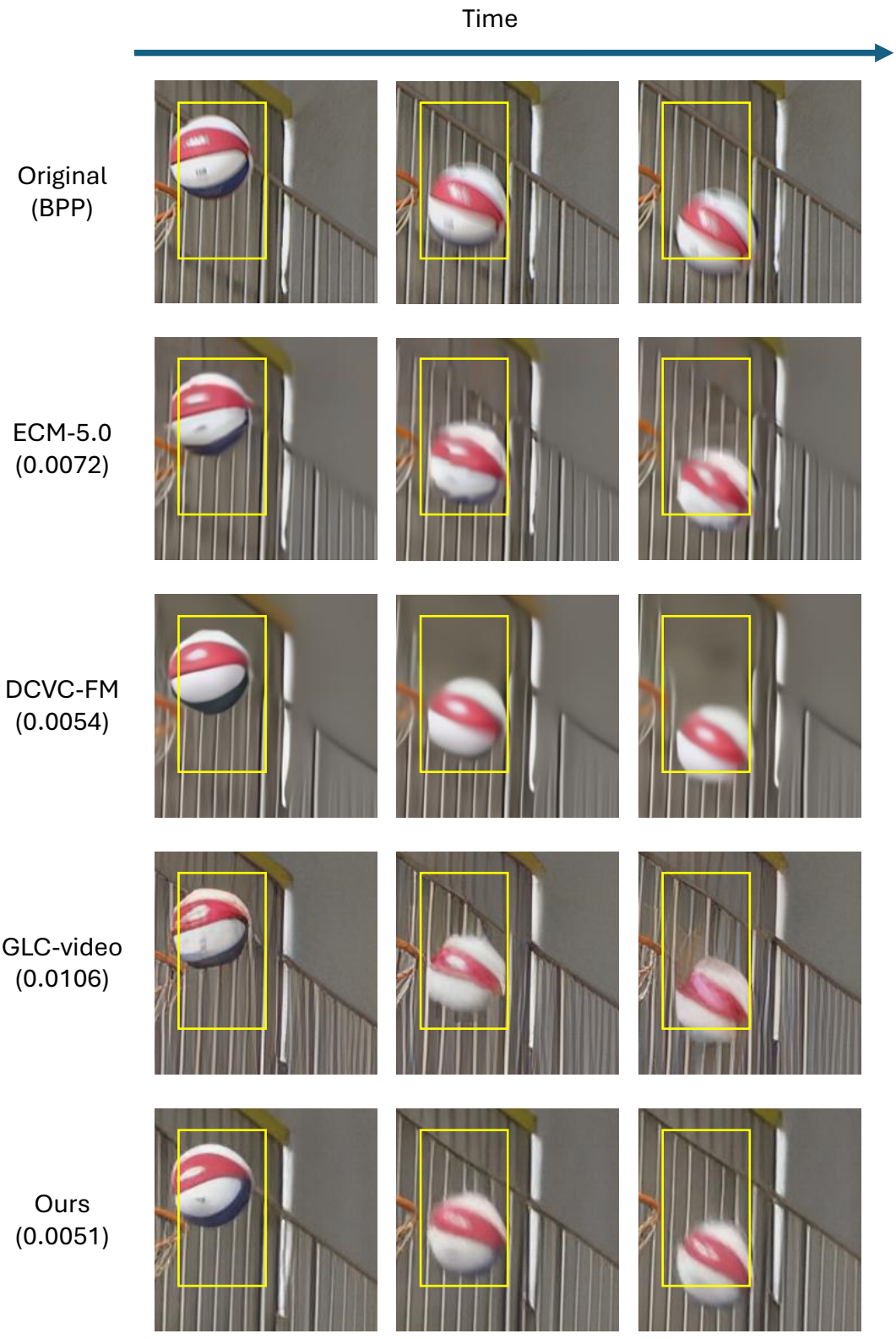

Figure 6. Visualization results across video frames.

[3] VTM. https://vcgit.hhi.fraunhofer.de/jvet/VVCSoftware_VTM. 1

[4] Original Vimeo links. https://github.com/anchen1011/toflow/blob/master/data/original_vimeo_links.txt. 1

[5] Duolikun Danier, Fan Zhang, and David Bull. Flolpips: A bespoke video quality metric for frame interpolation. In *2022 Picture Coding Symposium (PCS)*, pages 283–287. IEEE, 2022. 3

[6] D Flynn, K Sharman, and C Rosewarne. Common Test Conditions and Software Reference Configurations for HEVC Range Extensions, document JCTVC-N1006. *Joint Collaborative Team Video Coding ITU-T SG*, 16. 1

[7] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 2

[9] Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-25, 2024*, 2025. 1, 2

[10] Wei Jiang, Junru Li, Kai Zhang, and Li Zhang. Ecvc: Exploiting non-local correlations in multiple frames for contextual video compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 7331–7341, 2025. 1

[11] Dongwon Kim, Ju He, Qihang Yu, Chenglin Yang, Xiaohui Shen, Suha Kwak, and Liang-Chieh Chen. Democratizing text-to-image masked generative models with compact text-aware one-dimensional tokens. *arXiv preprint arXiv:2501.07730*, 2025. 1

[12] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 17-21, 2024*, 2024. 1, 2

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[14] Alexandre Mercat, Marko Viitanen, and Jarno Vanne. UVG dataset: 50/120fps 4K sequences for video codec analysis and development. In *Proceedings of the 11th ACM Multimedia Systems Conference*, pages 297–302, 2020. 1

[15] Kepan Nan, Rui Xie, Penghao Zhou, Tiehan Fan, Zhenheng Yang, Zhijie Chen, Xiang Li, Jian Yang, and Ying Tai. Openvid-1m: A large-scale high-quality dataset for text-to-video generation. In *The Thirteenth International Conference on Learning Representations*, 2025. 1

[16] Linfeng Qi, Zhaoyang Jia, Jiahao Li, Bin Li, Houqiang Li, and Yan Lu. Generative latent coding for ultra-low bitrate image and video compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025. 2

[17] JD Sutherland, Michael Arbel, and Arthur Gretton. Demystifying mmd gans. In *International conference for learning representations*, 2018. 2

[18] Haiqiang Wang, Weihao Gan, Sudeng Hu, Joe Yuchieh Lin, Lina Jin, Longguang Song, Ping Wang, Ioannis Katsavounidis, Anne Aaron, and C-C Jay Kuo. MCL-JCV: a JND-based H. 264/AVC video quality assessment dataset. In *2016 IEEE international conference on image processing (ICIP)*, pages 1509–1513. IEEE, 2016. 1

[19] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The thrity-seventh asilomar conference on signals, systems & computers, 2003*, pages 1398–1402. Ieee, 2003. 2

[20] Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Yuan Zhang, and Yan Lu. Dlf: Extreme image compression with dual-generative latent fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025. 2

[21] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *International Journal of Computer Vision*, 127(8): 1106–1125, 2019. 1