# Single-step Diffusion-based Video Coding with Semantic-Temporal Guidance

Supplementary Material

## 1. Experiment

This section first outlines the detailed experimental settings and then presents additional quantitative / qualitative results for a more comprehensive evaluation.

### 1.1. Compared Methods

Detailed configurations for each method are listed below.

**Traditional Video Codecs.** We evaluate VTM-17.0 [1] and ECM-5.0 [5] in RGB color space, following the setting used in DCVC series [6], where RGB input is converted to 10-bit YUV444 for internal codec processing. As noted in [6], this configuration improves compression performance. The following configuration files are used for each codec:

- VTM: *encoder_lowdelay_vtm.cfg*
- ECM: *encoder_lowdelay_ecm.cfg*

The parameters used for each video are as follows:

- −−c {config file name}
  −−InputFile={input video 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}

Notably, all coding tools and reference structure of traditional codecs use their best settings to achieve optimal compression performance.

**MSE-optimized Neural Codecs.** We use the officially released models of DCVC-FM [7] and DCVC-RT [3].

**Perceptual-optimized Neural Codecs.** For PLVC [21], we use the officially released model. For DiffVC [9], whose implementation is not publicly available, we adopt the results reported in their paper, as their evaluation protocol aligns with ours.

### 1.2. Additional Quantitative Evaluation

For completeness, we further evaluate S$^2$VC using PSNR and MS-SSIM [19], as shown in Fig. 1. We also report the warping error $E_{\text{warp}} = \frac{1}{N-1}\sum_{i=1}^{N-1}||x_{i+1} - F_{\text{wp}}(\hat{x}_i)||_1$ in Fig. 2, where the optical flow used in $F_{\text{wp}}$ is estimated from GT frames. At low bitrates, optimizing for PSNR tends to suppress high-frequency details, resulting in overly smooth reconstructions [11], as confirmed by our visual examples. As a result, higher objective metrics at these low bitrates do not necessarily reflect accurate perceptual quality.

Although S$^2$VC produces lower PSNR than objective-only codecs, both perceptual metrics and qualitative results consistently demonstrate its superior visual fidelity. Furthermore, compared to perceptually optimized codecs like PLVC and DiffVC, S$^2$VC achieves significantly higher objective scores while maintaining better perceptual quality. These results highlight that S$^2$VC delivers superior fidelity over prior perceptual codecs, without sacrificing visual realism, emphasizing the effectiveness of our approach.

We also report results on the lower-resolution datasets HEVC-C (832×480) and HEVC-E (1280×720), as shown in Fig. 3. On these datasets, S$^2$VC continues to outperform other methods in terms of perceptual quality, as measured by LPIPS, DISTS, FloLPIPS, and FID, demonstrating its generalization capability to 480p and 720p videos.

### 1.3. Additional Qualitative Examples

We present additional qualitative comparisons on Fig. 10. S$^2$VC consistently outperforms prior video codecs, delivering superior visual quality across diverse content, yet with the lowest bitrate cost.

Temporal consistency comparisons across neural codecs are provided in Fig. 11, Fig. 12 and Fig. 13. S$^2$VC yields stable reconstructions for both moving objects and background regions. In contrast, DCVC-FM produces blurry results, while PLVC introduces jitter and flicker artifacts in motion areas, leading to degraded temporal coherence.

These results demonstrate that S$^2$VC excels not only in generating detailed frame-level content but also in maintaining strong temporal consistency.

### 1.4. Coding Latency

We compare the coding latency of S$^2$VC with the neural codec DCVC-FM, evaluating both encoding and decoding speed in frames-per-second (fps). All tests are conducted on an A100 GPU using 1920×1080 video inputs. As reported in Table 1, the motion-vector-free design of S$^2$VC enables faster encoding than DCVC-FM, demonstrating its efficiency on the encoding side. In contrast, the large one-step diffusion generator results in slower decoding, which is expected given the high generative capacity required for perceptual reconstruction. As demonstrated by the RD results in the main paper, this computational cost translates into substantially improved perceptual quality.

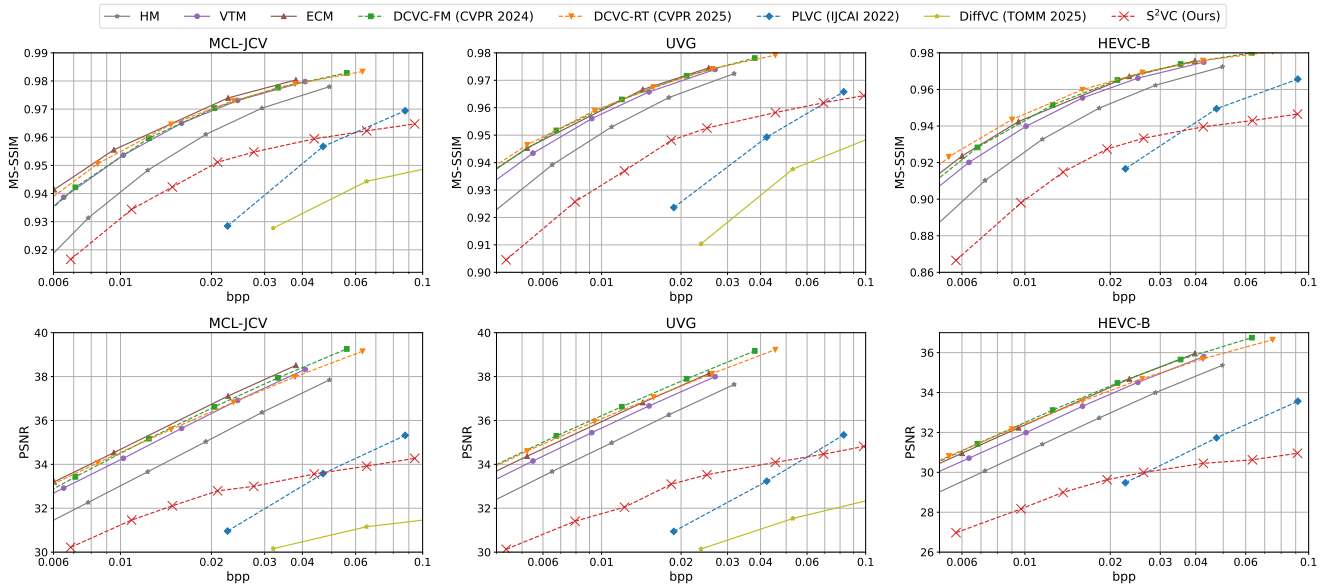


Figure 1. Rate-distortion curves in terms of PSNR and MS-SSIM [19]. Datasets: MCL-JCV [18], UVG [12] and HEVC-B [15].

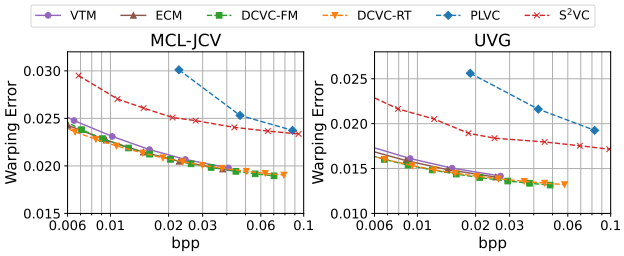


Figure 2. Rate-distortion curves in terms of warping error $E_{\text{warp}}$.

Since the primary objective of this work is to investigate perceptual optimization with single-step diffusion in video coding, runtime optimization is not the main focus. Future improvements, such as integerization of the compression module or distilling a smaller diffusion model, could further reduce coding latency.

Table 1. Coding speed on 1920×1080 video with an A100 GPU

| Model | Enc. fps | Dec. fps |
|---|---|---|
| DCVC-FM [7] | 5.0 | 5.9 |
| S$^2$VC (Ours) | 6.6 | 1.27 |

## 2. Model Training

This section outlines the training protocol of the proposed S$^2$VC model, including the pretrained components, dataset, and overall training procedure. We also describe the random-resolution strategy and the random group-wise cascade scheme used during training.

**Pretrained Models.** To initialize the one-step diffusion model, we use the SD1.5 variant of DMD2 [22], leveraging its pretrained generative prior to enhance training. For semantic guidance, we adopt the *dinov3-convnext-base* model [14], as its convolutional architecture efficiently supports variable spatial resolutions. For the I-frame codec, we use the pretrained OneDC [20] models at five bitrate levels.

**Training Dataset.** We train our model on the OpenVid-HD dataset [13], a high-quality HD collection containing approximately 0.4M video clips.

**Training Procedure.** The training loss is defined as the sum of rate, distortion, semantic, and motion losses:

$$L = \frac{1}{T}\sum_{t=1}^{t=T}(\lambda R + L_{\text{D}} + \alpha L_{\text{sem}} + \beta L_{\text{motion}}) \quad (1)$$

where $T$ is the number of frames per iteration, and $R$ is the bitrate from the spatial-temporal entropy model [3]. The parameter $\lambda$ controls the rate-distortion trade-off, and we set $\lambda$ to $\{0.2, 0.4, 0.8, 1.2, 1.8, 2.6, 3.7, 5.2\}$ for different bitrate. Other terms are defined as follows:

$$L_{\text{sem}} = ||E_{\text{DINO}}(x_t) - P_{\text{aux}}(s_t)||_1 \quad (2)$$

$$L_{\text{D}} = ||x_t - \hat{x}_t||_1 + L_{\text{LPIPS}}(x_t, \hat{x}_t) \quad (3)$$

$$L_{\text{motion}} = ||O(x_{t-1}, x_t) - O(\hat{x}_{t-1}, \hat{x}_t)||_1 \quad (4)$$

where $O$ refers to the pretrained RAFT model [16], used to improve temporal consistency through optical flow alignment. We set $\alpha = 0.001$ and $\beta = 0.25$ during training, and optimize using AdamW [8] with a total batch size of 16 across all GPUs. We gradually increase the frame number T and decrease the learning rate to stabilize training:

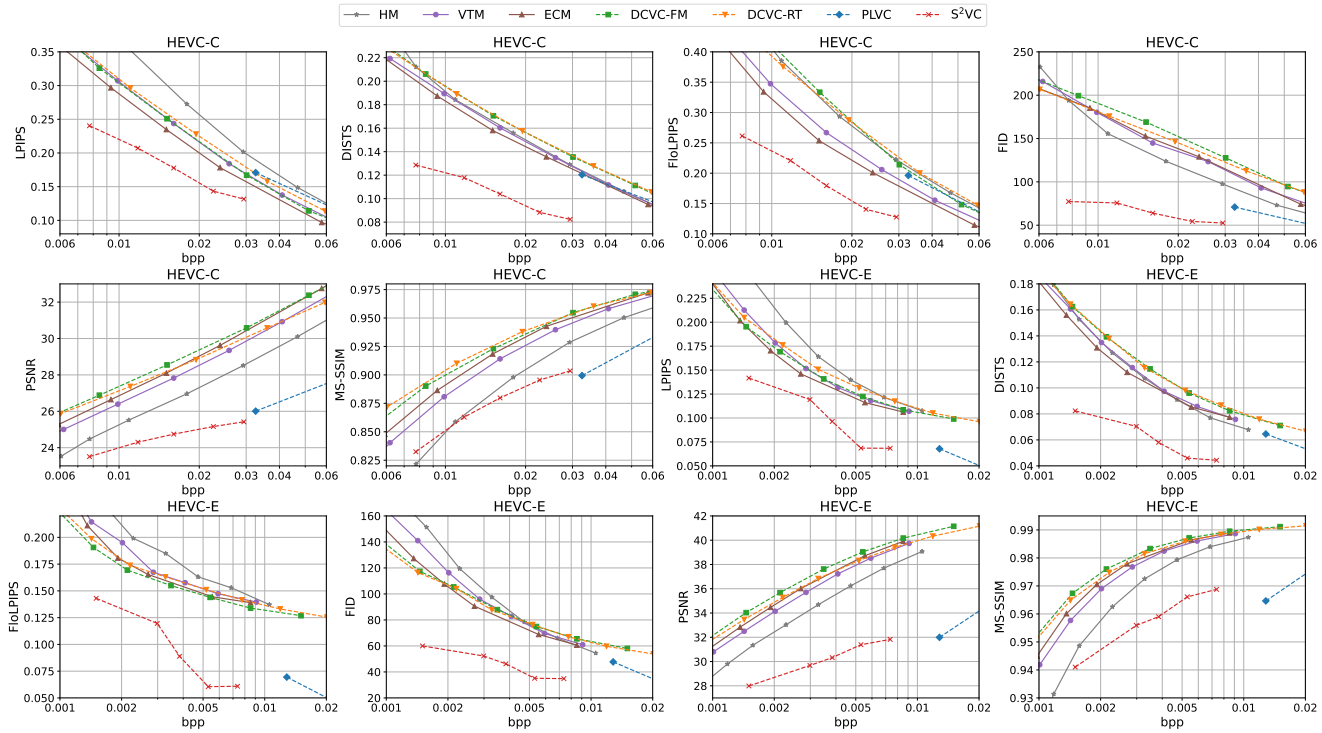


Figure 3. Rate-distortion curves on low-resolution datasets: HEVC-C and HEVC-E [15].

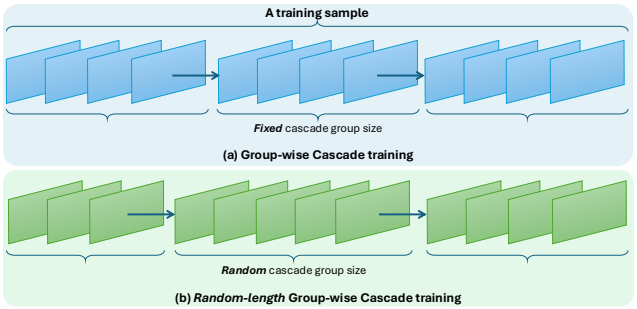


Figure 4. Comparison of two types of group-wise training. The proposed variant in **(b)** increases training diversity by randomly choosing the length of each group during cascade training.

- For the first 20,000 iterations, we linearly increase T from 2 to 32, while linearly decreasing the learning rate from 1e-4 to 5e-6.
- For the next 10,000 iterations, we fix T=32 and set the learning rate to 1e-6.

All training tasks are conducted on 4×A100 80GB GPUs, taking approximately 14 days.

**Random Resolution Training.** To enable the one-step diffusion model to generalize across videos with varying resolutions, we train it using randomly selected resolutions from

$$\{512 \times 512, 512 \times 768, 768 \times 1024, 1024 \times 1024\}$$

For each training iteration, a resolution is randomly sampled from this set, and a corresponding patch is cropped from a random spatial location within the training video.

Table 2. Parameter Count

| Module | Parameters |
|---|---|
| Codec Module | 144M |
| Diffusion Module | 1152M |
| Base U-Net * | 860M |
| P-frame LoRA | 68M |
| TCG blocks | 224M |
| VAE decoder | 50M |
| **Total** | **1346M** |

*Auxiliary predictor (Training only): 36M.*
*Module with * is frozen.*

**Random Group-wise Cascade Training.** We adopt the group-wise cascade training scheme following ECVC [4] to reduce memory consumption when training the codec. To increase training diversity and enhance generalization, we randomize the cascade propagation length with a maximum of 9 frames per iteration, as shown in Fig. 4.

## 3. Model Architecture

This section first describes the LoRA [2] configuration applied to the diffusion U-Net and provides the associated parameter count. It then outlines the architecture of all modules used in the proposed model.

**LoRA Configuration and Parameter Count.** We incorporate LoRA layers into all blocks of the diffusion U-Net

(DMD2 distilled SD1.5 version [22]) using the PEFT library [10]. Following the configuration of OneDC [20], we set the LoRA rank to 64, the scaling factor (LoRA $\alpha$) to 8.0, and the dropout rate to 0.0. The detailed parameter statistics are provided in Table 2.

**Detailed Model Architecture.** The complete architectures of all modules are illustrated in Fig. 5–9. The two-step entropy estimation module follows the design in [3], with channel dimensions adjusted to match our framework. The ResBlock and AttnBlock structures are adopted from the Diffusers library [17].

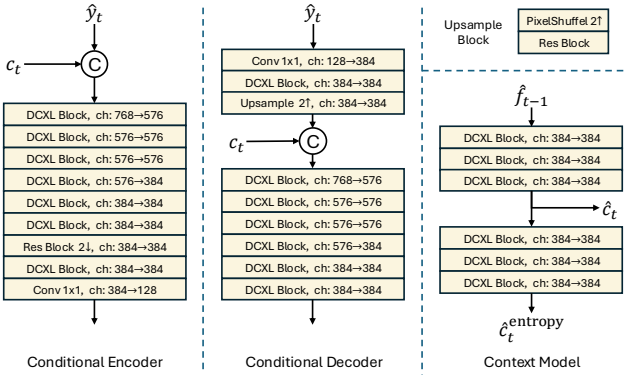


Figure 5. Architecture of the Conditional Encoder/Decoder and Context model used in S$^2$VC.

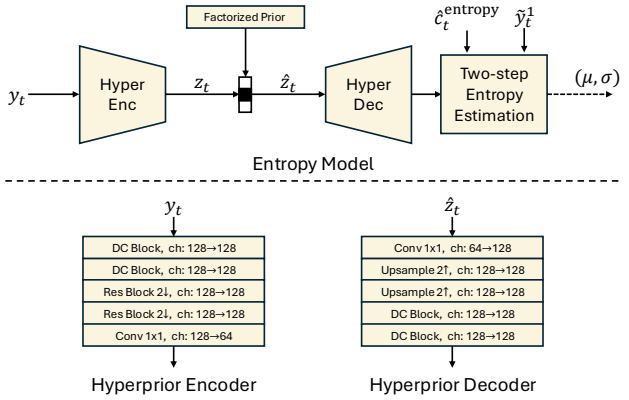


Figure 6. Architecture of the Entropy Model used in S$^2$VC. For Two-step Entropy Estimation, we use the implementation in [3].

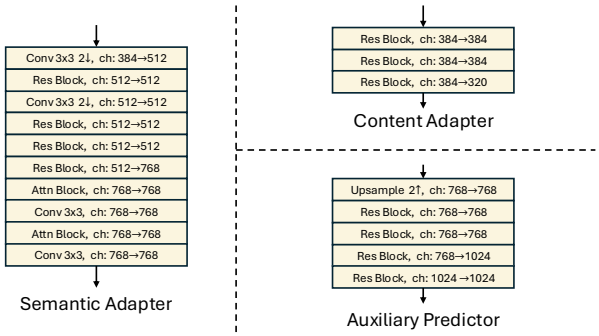


Figure 7. Architecture of the Content/Semantic Adapter and Auxiliary predictor.

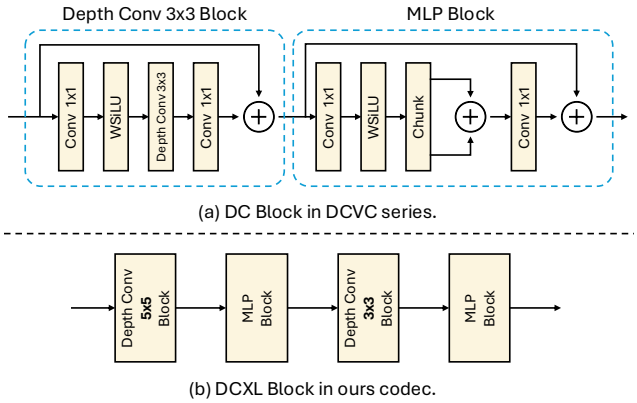


Figure 8. DC / DCXL blocks used in S$^2$VC. DC Block is from [3], and we enlarge it for higher model capacity.

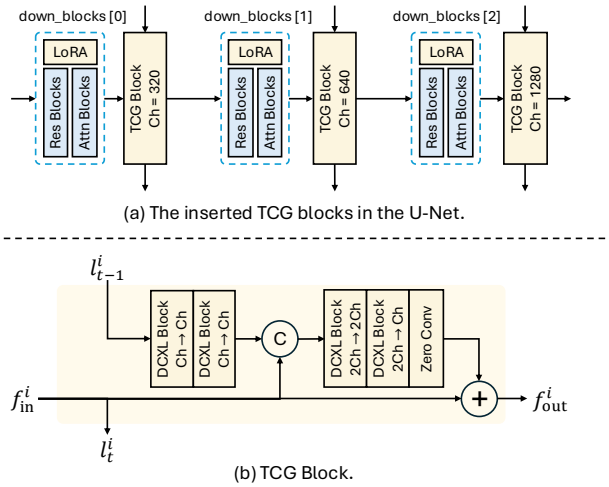


Figure 9. TCG block insertion in the diffusion U-Net. We build upon the *UNet2DConditionModel* architecture from Diffusers [17] and insert TCG blocks into the first three scales, placing one after each *down_blocks* stage. $l_t^i$ denotes the intermediate feature at the $i$-th scale of frame $t$. $f_{\text{in}}^i$ / $f_{\text{out}}^i$ are input and output of a TCG block.

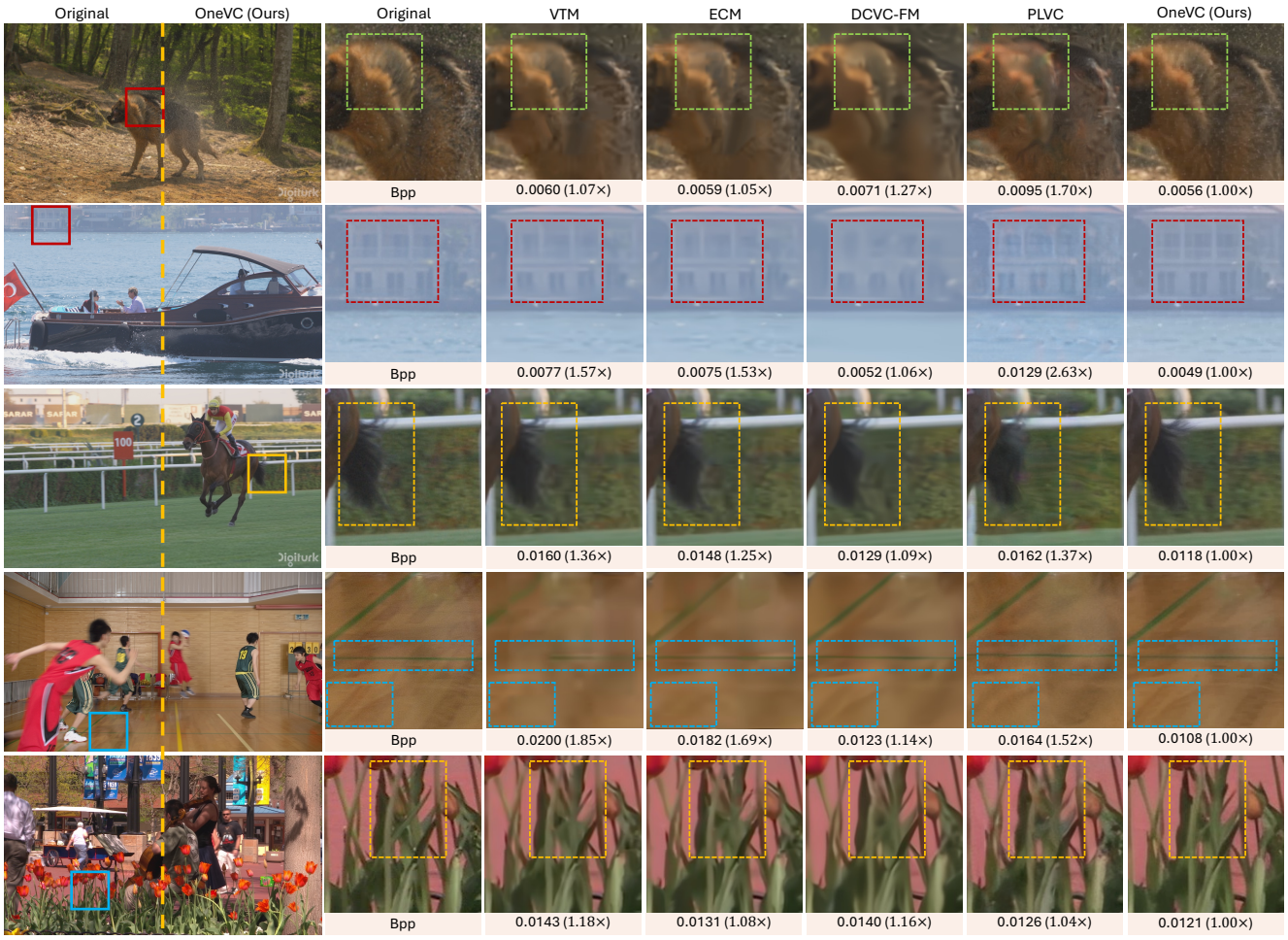


Figure 10. Additional visual examples. The proposed $S^2$VC delivers the most realistic and faithful reconstruction at the lowest bitrate. In contrast, traditional codecs (VTM/ECM) and the neural codec DCVC-FM produce blurry reconstructions at low bitrates, while the perceptual codec PLVC generates distorted detail.

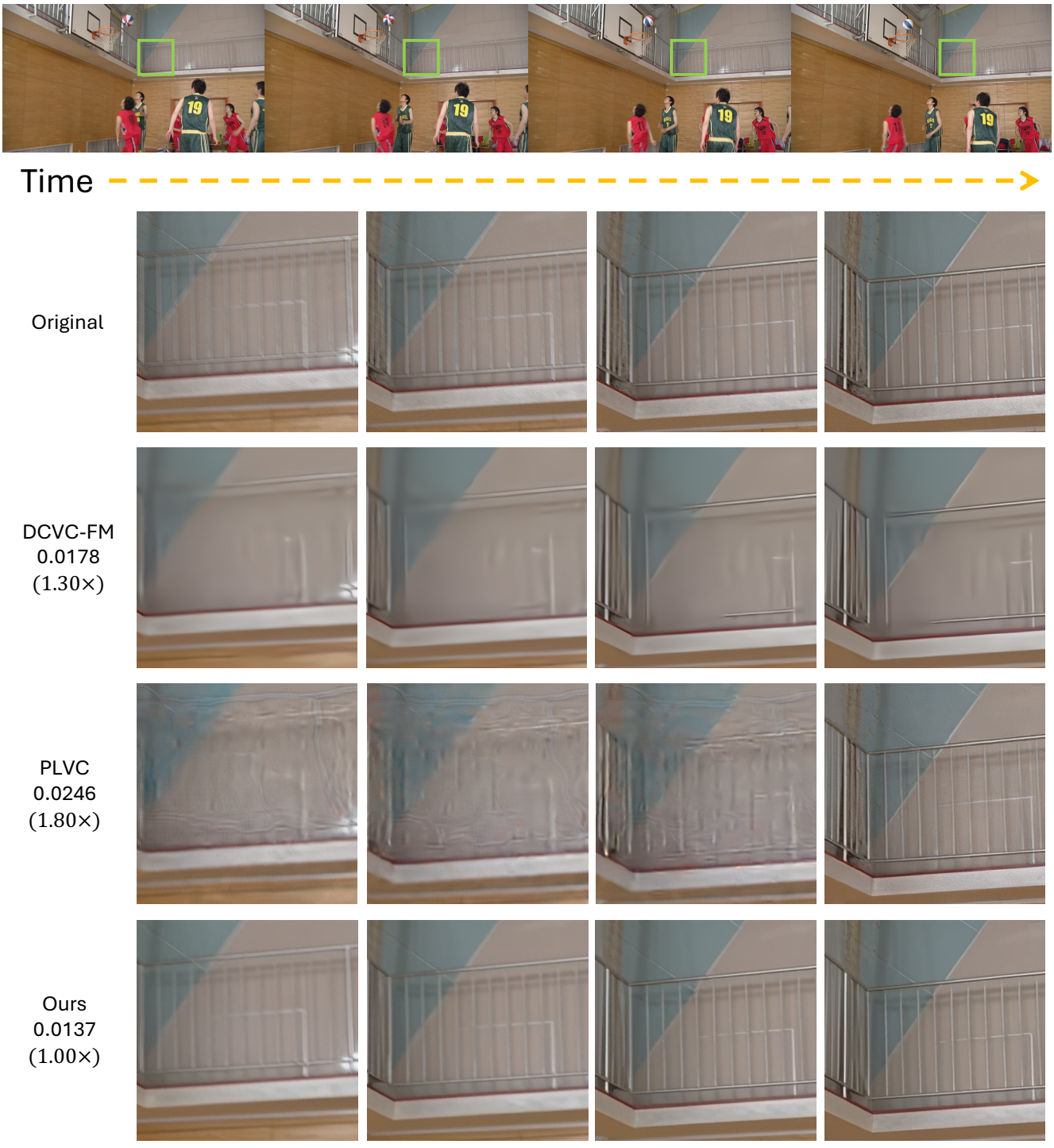


Figure 11. The proposed S$^2$VC accurately reconstructs the railing structure, whereas the neural codec DCVC-FM fails to recover its geometry, and the perceptual codec PLVC introduces additional temporally inconsistent noise.


Figure 12. Additional visual examples in a motion scenario across neural codecs. The proposed S$^2$VC delivers the clearest and most temporally consistent texture on the wall, even as the subject moves over it. In contrast, DCVC-FM generates blurry details, while PLVC produces distorted textures that are inconsistent over time.

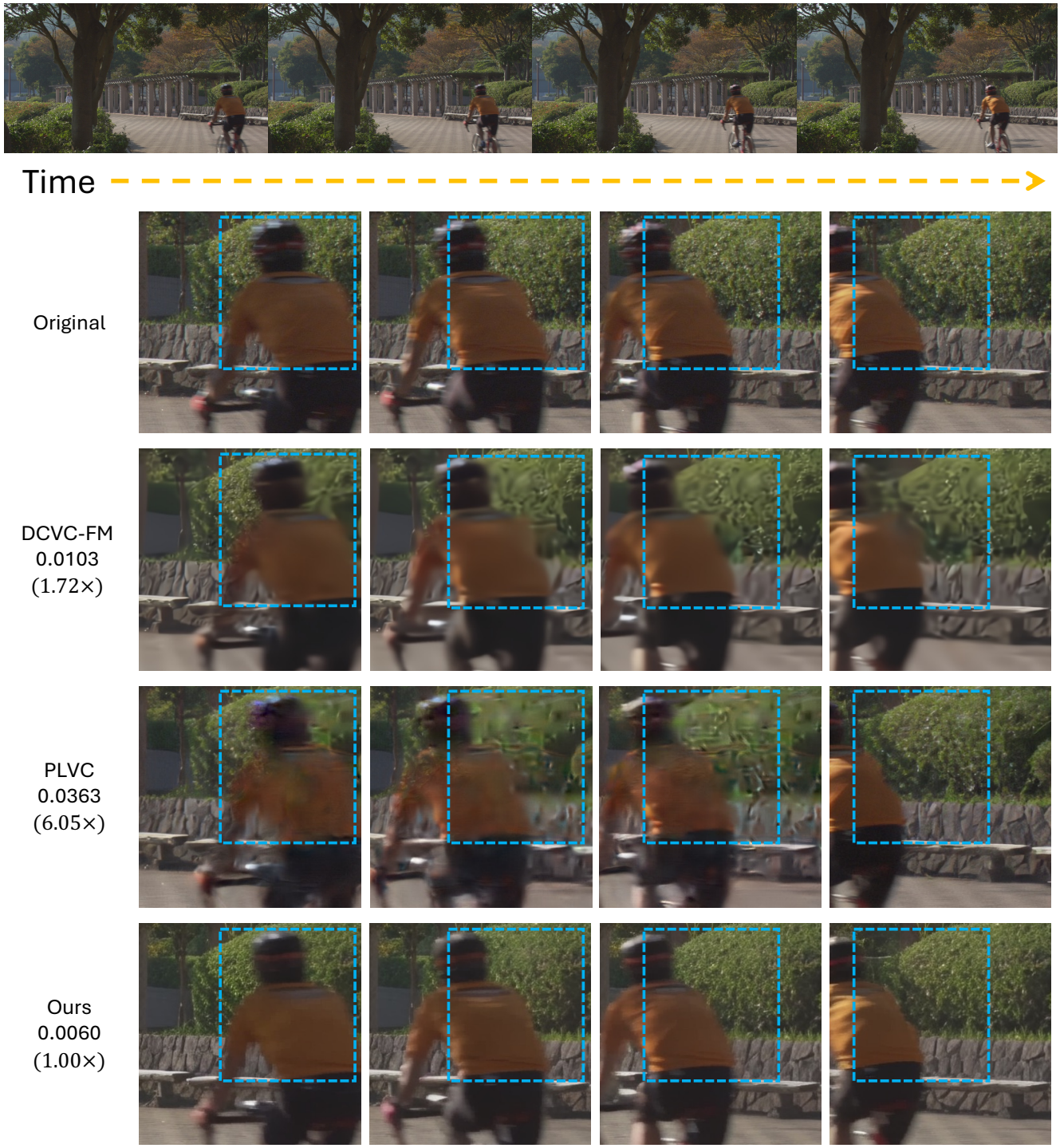


Figure 13. Additional visual examples in a motion scenario across neural codecs. The proposed S$^2$VC synthesizes consistent texture on the background trees, unaffected by the moving subject. In contrast, DCVC-FM generates blurry details, while PLVC produces jitter artifacts on both the subject and background, resulting in temporal inconsistencies.

## References


[1] 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*, 31(10):3736–3764, 2021. 1

[2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[3] Zhaoyang Jia, Bin Li, Jiahao Li, Wenxuan Xie, Linfeng Qi, Houqiang Li, and Yan Lu. Towards practical real-time neural video compression. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12543–12552, 2025. 1, 2, 4

[4] 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. 3

[5] JVET. Explorations: Enhanced compression beyond vvc capability (ecm), 2025. 1

[6] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with diverse contexts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22616–22626, 2023. 1

[7] Jiahao Li, Bin Li, and Yan Lu. Neural video compression with feature modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26099–26108, 2024. 1, 2

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[9] Wenzhuo Ma and Zhenzhong Chen. Diffusion-based perceptual neural video compression with temporal diffusion information reuse. *arXiv preprint arXiv:2501.13528*, 2025. 1

[10] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022. 4

[11] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020. 1

[12] 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. 2

[13] 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. *arXiv preprint arXiv:2407.02371*, 2024. 2

[14] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025. 2

[15] 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*, 22(12):1649–1668, 2012. 2, 3

[16] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 2

[17] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 4

[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. 2

[19] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pages 1398–1402 Vol.2, 2003. 1, 2

[20] Naifu Xue, Zhaoyang Jia, Jiahao Li, Bin Li, Yuan Zhang, and Yan Lu. One-step diffusion-based image compression with semantic distillation. *arXiv preprint arXiv:2505.16687*, 2025. 2, 4

[21] Ren Yang, Radu Timofte, and Luc Van Gool. Perceptual learned video compression with recurrent conditional gan. In *IJCAI*, pages 1537–1544, 2022. 1

[22] Tianwei Yin, Michaël Gharbi, Taesung Park, Richard Zhang, Eli Shechtman, Fredo Durand, and Bill Freeman. Improved distribution matching distillation for fast image synthesis. *Advances in neural information processing systems*, 37:47455–47487, 2024. 2, 4