Spatio-temporal/Video INR

Background

- INR: f(coordinate)=signal value at that coordinate
  - Audio: f(t)=amplitude(t)
  - Image: f(x,y)=(R,G,B)
  - Video: f(x,y,t)=(R,G,B) NERV f(t)--(R,G,B) f(x,y,0/1)=(R,G,B,t)
    - super-resolution 480p—1080p
    - Interpolation 30fps—60fps
    - compression

- A simple MLP doesn’t directly work for video.
  - A simple MLP struggles to capture complex spatiotemporal correlations.
  - We need more expressive and efficient representations to capture these structures.

NeRV: Neural Representations for Videos

Hao Chen[1], **Bo He**[1], **Hanyu Wang**[1], **Yixuan Ren**[1], **Ser-Nam Lim**[2], **Abhinav Shrivastava**[1]

[1]University of Maryland, College Park, [2]Facebook AI

{chenh, bohe, hywang66, yxren, abhinav}@umd.edu, sernamlim@fb.com

Nerv: Neural representations for videos

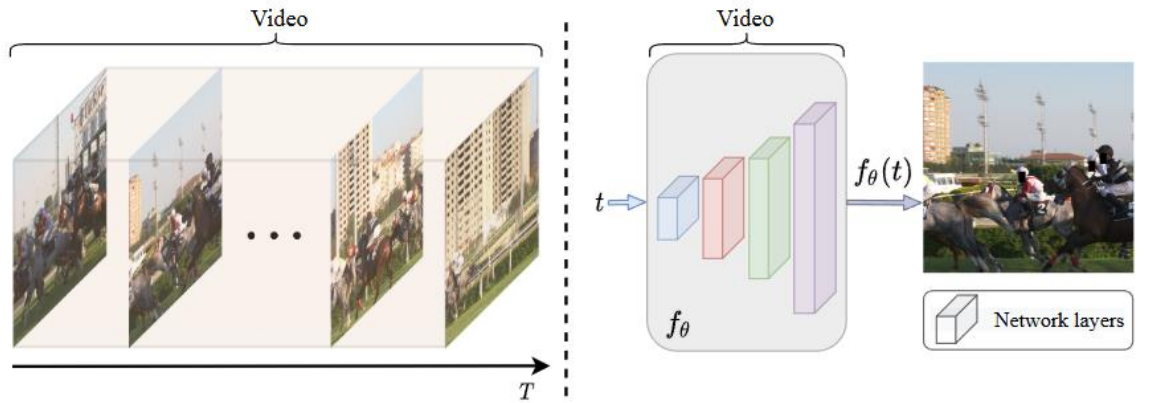
H Chen, B He, H Wang, Y Ren, SN Lim, A Shrivastava

Advances in Neural Information Processing Systems, 2021 · proceedings.neurips.cc

Abstract

We propose a novel neural representation for videos (NeRV) which encodes videos in neural networks. Unlike conventional representations that treat videos as frame sequences, we represent videos as neural networks taking frame index as input. Given a frame index, NeRV outputs the corresponding RGB image. Video encoding in NeRV is simply fitting a neural network to video frames and decoding process is a simple feedforward operation. As an image-wise implicit representation, NeRV output the whole

SHOW MORE



(a) Explicit representations for videos (e.g., HEVC) (b) Neural implicit representations for videos (e.g., NeRV)

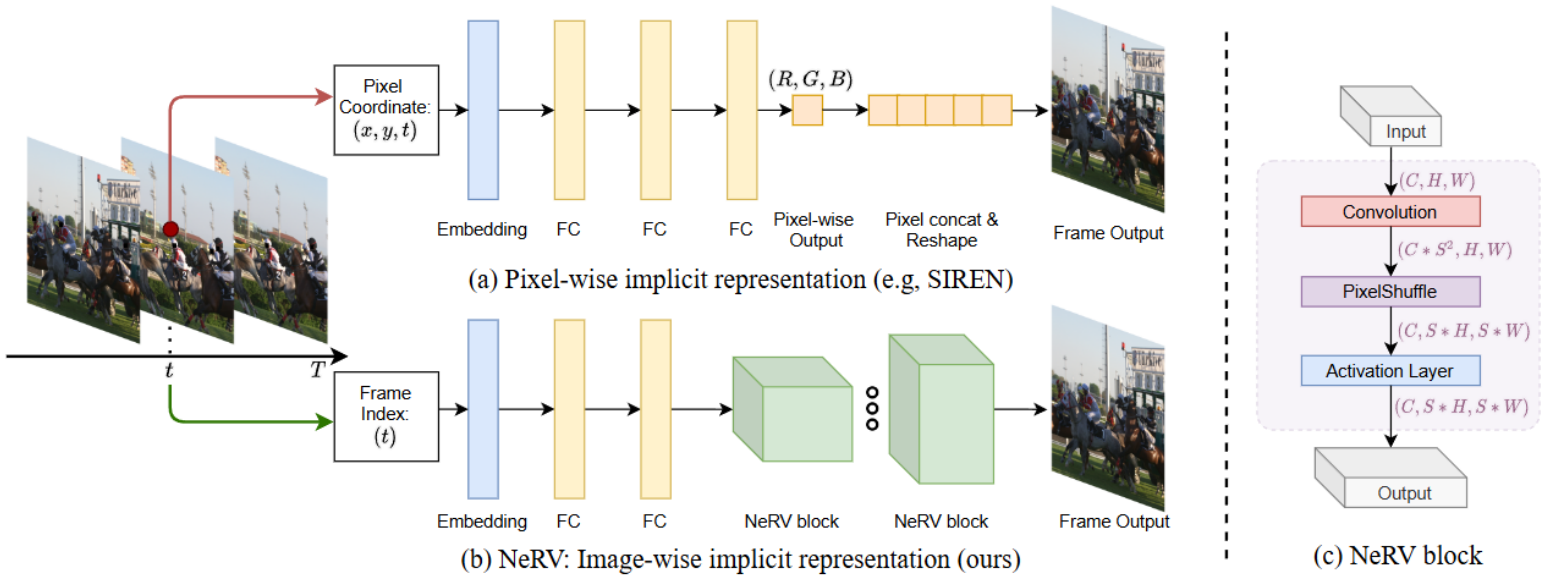
Figure 1: **(a)** Conventional video representation as **frame sequences**. **(b)** NeRV, representing video as **neural networks**, which consists of multiple convolutional layers, taking the normalized frame index as the input and output the corresponding RGB frame.

Main Parts

- Instead of modeling video as a mapping from spatial-temporal coordinates f(x, y, t) → RGB, they directly model f(t) → frame.
  - instead of querying each pixel independently, they generate the entire frame in one forward pass.

Methods	Parameters	Training Speed ↑	Encoding Time ↓	PSNR ↑	Decoding FPS ↑
SIREN [5]	3.2M	1×	2.5×	31.39	1.4
NeRF [4]	3.2M	1×	2.5×	33.31	1.4
NeRV-S (ours)	3.2M	**25**×	**1**×	**34.21**	**54.5**

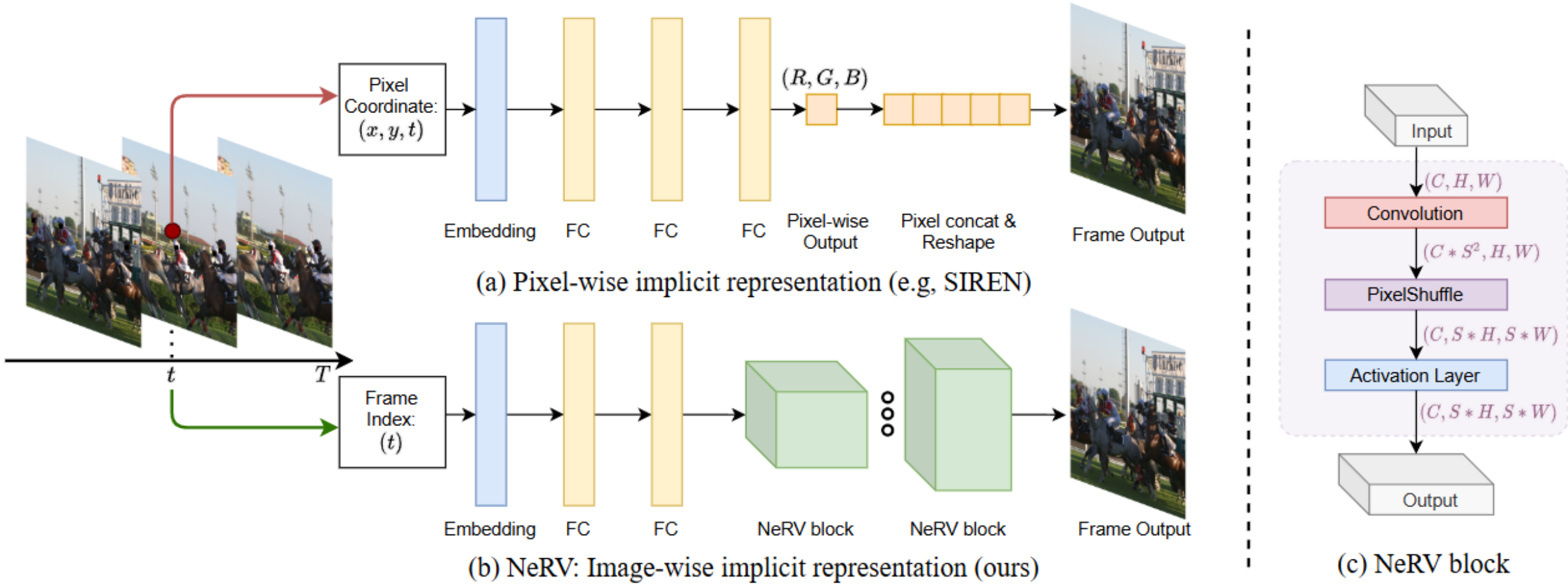
- Architecture



(a) Pixel-wise implicit representation (e.g, SIREN)

(b) NeRV: Image-wise implicit representation (ours)

(c) NeRV block



(a) Pixel-wise implicit representation (e.g, SIREN)

(b) NeRV: Image-wise implicit representation (ours)

(c) NeRV block

Our goal: H×W×C → 2H×2W×C

H×W×C → H×W×4C → process → reshape → 2H×2W×C

instead of

H×W×C → 2H×2W×C→ process

Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network

W Shi, J Caballero, F Huszár, J Totz, AP Aitken, R Bishop, D Rueckert, Z Wang

Proceedings of the IEEE conference on computer vision and pattern ..., 2016 · cv-foundation.org

Abstract

Recently, several models based on deep neural networks have achieved great success in terms of both reconstruction accuracy and computational performance for single image super-resolution. In these methods, the low resolution (LR) input image is upscaled to the high resolution (HR) space using a single filter, commonly bicubic interpolation, before reconstruction. This means that the super-resolution (SR) operation is performed in HR space. We demonstrate that this is sub-optimal and adds computational complexity. In this

SHOW MORE

☆ Save Cite Cited by 9273 Related articles All 14 versions

Other Details

- Input Embedding $\Gamma(t) = \left(\sin\left(b^0\pi t\right), \cos\left(b^0\pi t\right), \ldots, \sin\left(b^{l-1}\pi t\right), \cos\left(b^{l-1}\pi t\right)\right)$
- Loss Function $L = \frac{1}{T}\sum_{t=1}^{T} \alpha \left\| f_\theta(t) - v_t \right\|_1 + (1-\alpha)(1 - \mathrm{SSIM}(f_\theta(t), v_t))$ $\mathrm{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$
- Compression we can use any neural network compression method as a proxy for video compression
  - Pruning

$$\theta_i = \begin{cases} \theta_i, & \text{if } \theta_i \geq \theta_q \\ 0, & \text{otherwise,} \end{cases}$$

  - Quantization

$$\mu_i = \mathrm{round}\left(\frac{\mu_i - \mu_{\min}}{2^{\mathrm{bit}}}\right) * \mathrm{scale} + \mu_{\min}, \quad \mathrm{scale} = \frac{\mu_{\max} - \mu_{\min}}{2^{\mathrm{bit}}}$$

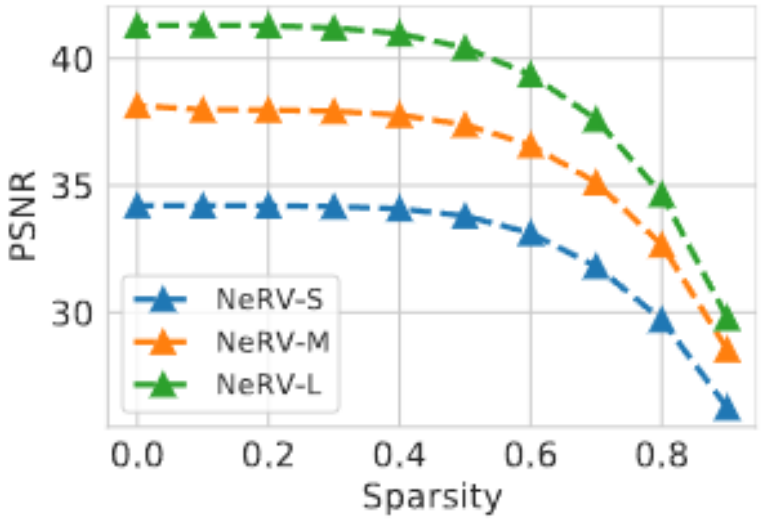


Figure 4: Model **pruning**. Sparsity is the ratio of parameters pruned.

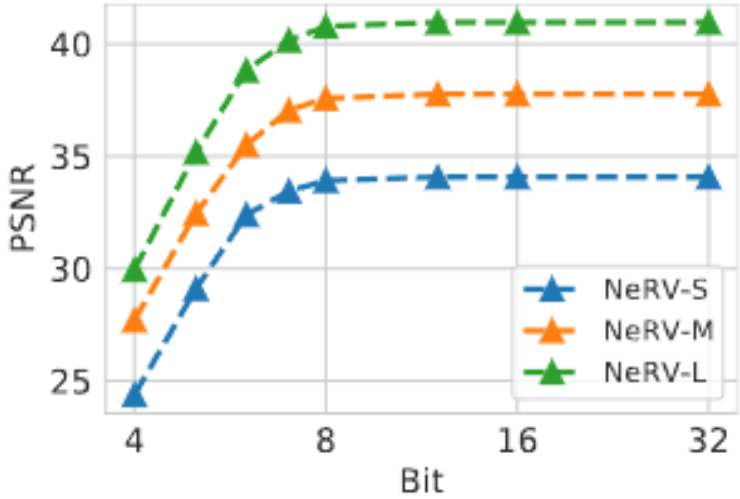


Figure 5: Model **quantization**. Bit is the bit length used to represent parameter value.

Table 6: Input embedding ablation. PE means positional encoding

	PSNR	MS-SSIM
None	24.93	0.769
PE	**37.26**	**0.970**

Table 7: Upscale layer ablation

	PSNR	MS-SSIM
Bilinear Pooling	29.56	0.873
Transpose Conv	36.63	0.967
PixelShuffle	**37.26**	**0.970**

Table 9: Activation function ablation

	PSNR	MS-SSIM
ReLU	35.89	0.963
Leaky ReLU	36.76	0.968
Swish	37.08	0.969
GELU	**37.26**	**0.970**

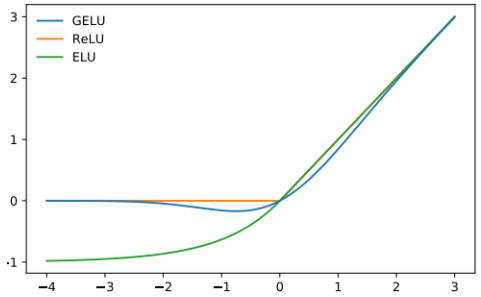


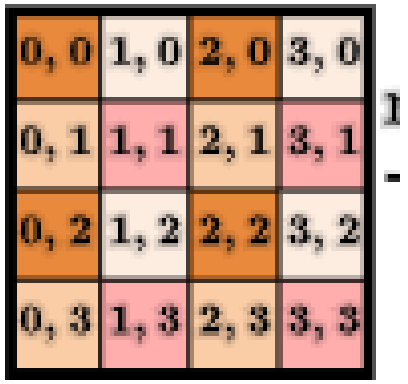
Table 10: Loss objective ablation

L2	L1	SSIM	PSNR	MS-SSIM
✓			35.64	0.956
	✓		35.77	0.959
		✓	35.69	**0.971**
✓	✓		35.95	0.960
✓		✓	36.46	0.970
	✓	✓	**37.26**	0.970

HiNeRV: Video Compression with Hierarchical Encoding-based Neural Representation

Ho Man Kwan†, Ge Gao†, Fan Zhang†, Andrew Gower‡, David Bull†
† Visual Information Lab, University of Bristol, UK
‡ Immersive Content & Comms Research, BT, UK
{hm.kwan, ge1.gao, fan.zhang, dave.bull}@bristol.ac.uk,
andrew.p.gower@bt.com

- **(i,j,t)→patch** instead of **t→frame**



- **learnable grid encoding** instead of **fourier encoding**

- **parameter-free bilinear upsampling** instead of **heavy learned upsampling**

- **adaptive pruning**

$$\theta_i = \begin{cases} \theta_i, & \text{if } \theta_i \geq \theta_q \\ 0, & \text{otherwise,} \end{cases} \longrightarrow \frac{|\theta_p|}{P^\lambda}$$

- **quantization-aware compression**
  - perform a short fine-tuning with Quant-Noise after weight pruning

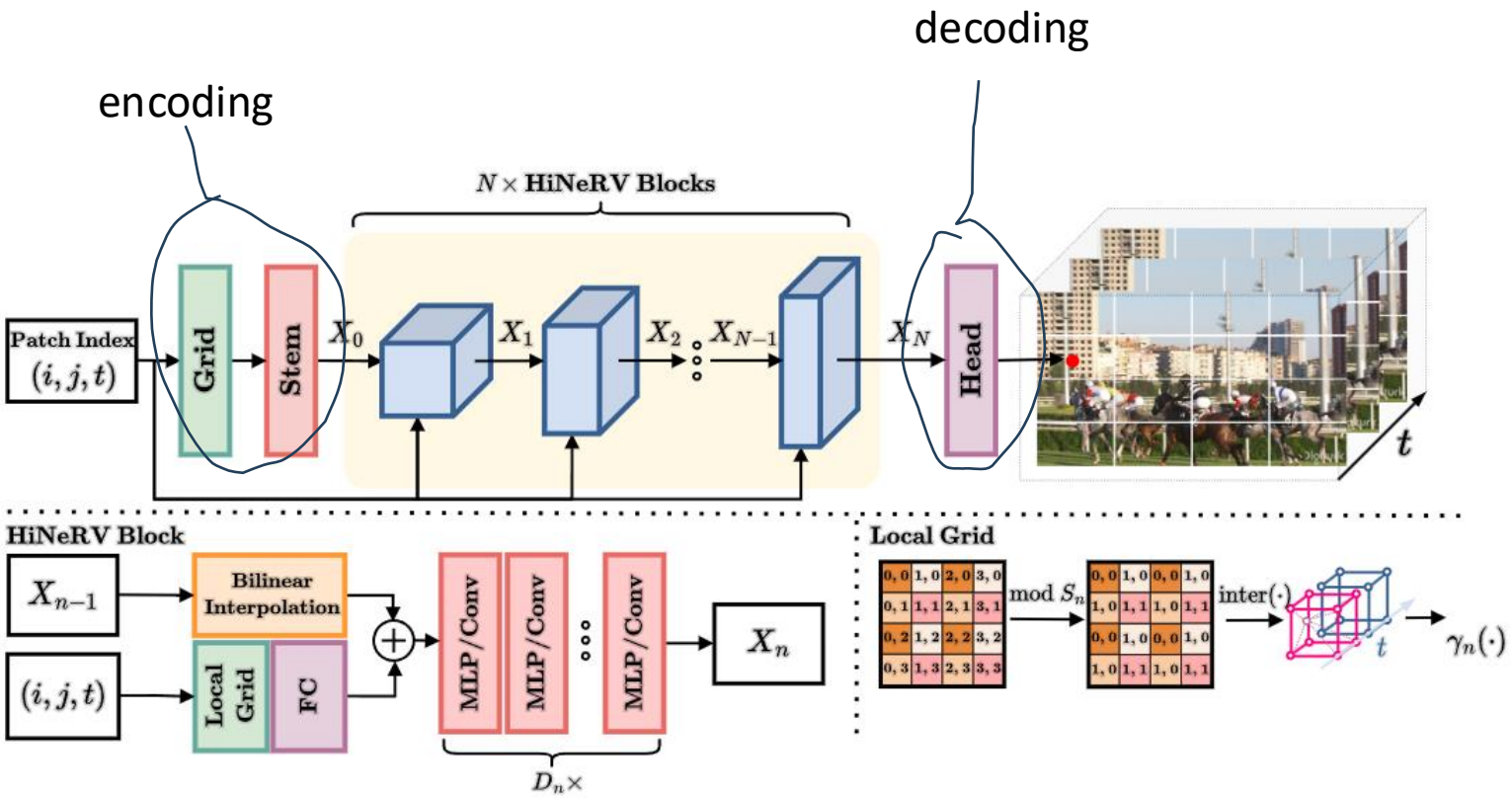


Figure 2: Top: The HiNeRV architecture. Bottom left: The HiNeRV block. HiNeRV block take feature maps X_{n-1} and patch index (i, j, t) as input, upsample the feature maps, enhances it with the hierarchical encoding, then computes the transformed maps X_n. Bottom right: The local grid. In HiNeRV, the hierarchical encoding is computed by performing interpolation from the local grid, where the modulo of the coordinates is being used.

(0,0,0) (1,0,0)

(0,1,0) (1,1,0)

100x100x3
50x50x100
25x25x1000

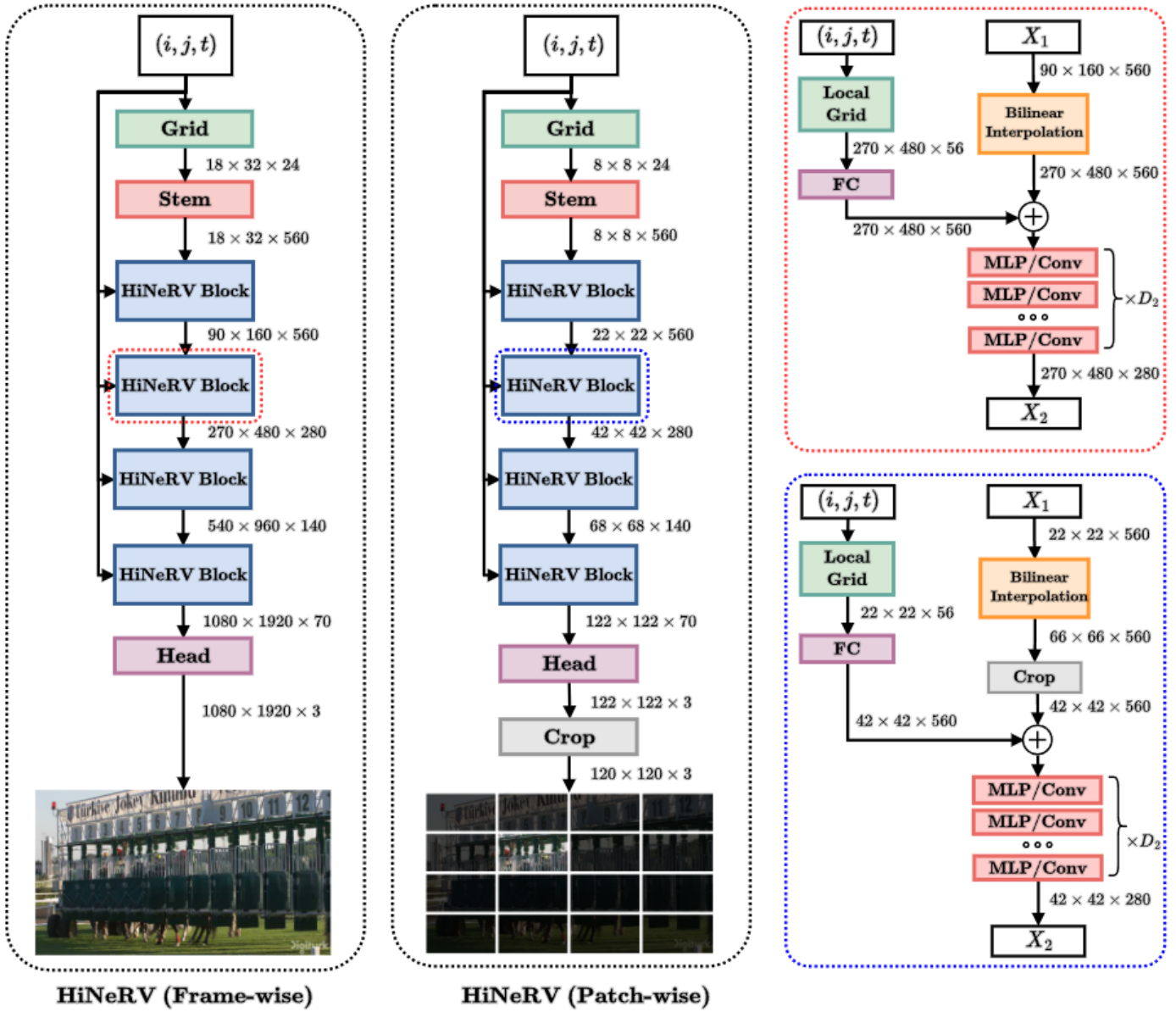
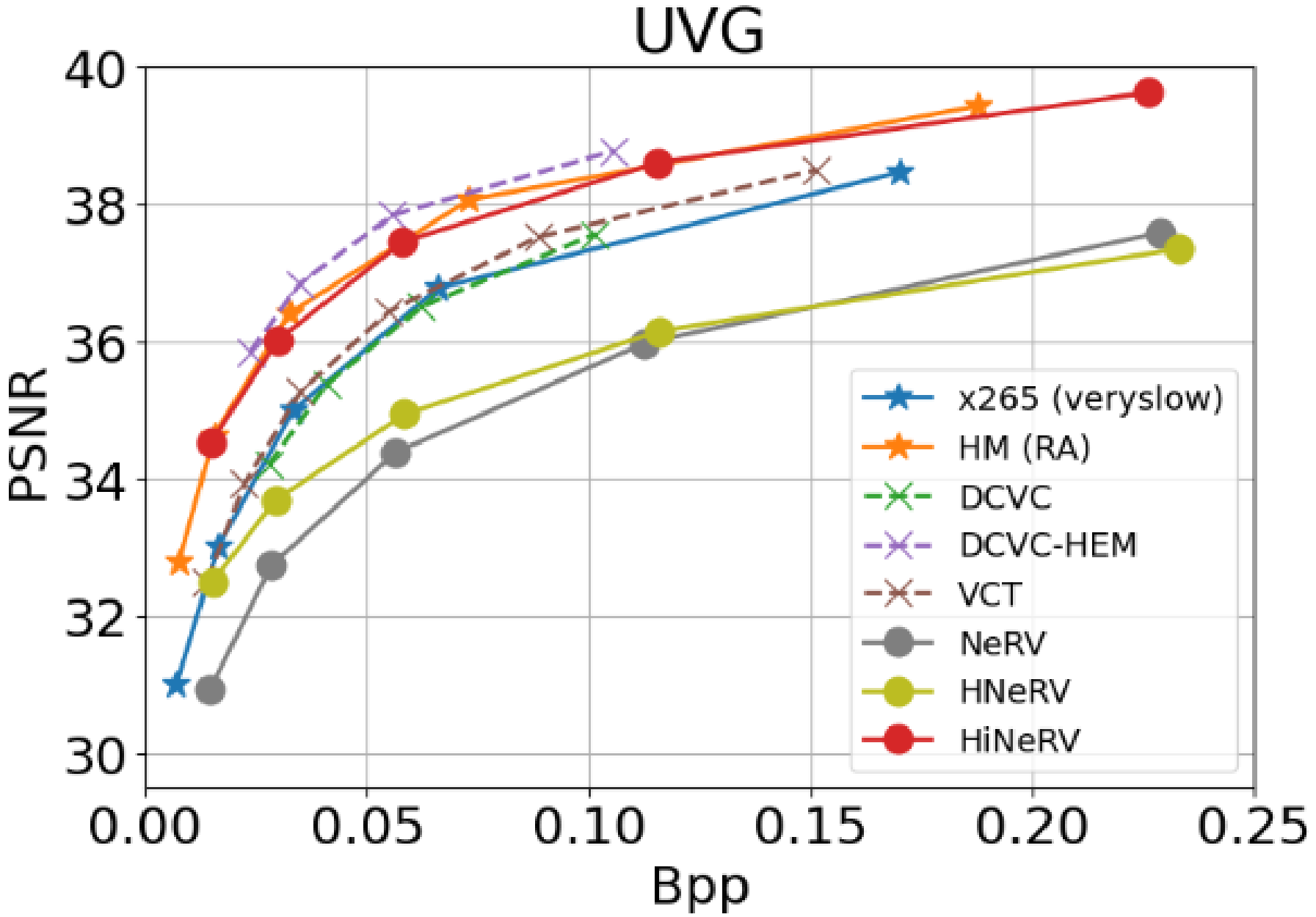


Figure 4: Illustration of the proposed HiNeRV models employing frame-based or patch-based representation.


UVG
PSNR
Bpp
40
38
36
34
32
30
0.00
0.05
0.10
0.15
0.20
0.25
x265 (veryslow)
HM (RA)
DCVC
DCVC-HEM
VCT
NeRV
HNeRV
HiNeRV

Ablation Studies

Table 4: Ablation studies of HiNeRV with the UVG dataset [38]. Results are in PSNR.

Model	Size	Beauty	Bosph.	Honey.	Jockey	Ready.	Shake.	Yacht.	Avg.
NeRV	3.20M	34.03	32.77	39.59	30.39	23.88	33.85	26.88	31.63
HNeRV	3.22M	35.04	35.72	41.11	32.20	25.88	35.75	29.69	33.63
HiNeRV	3.17M	35.67	**39.37**	41.61	**36.94**	**31.98**	36.74	**31.57**	**36.27**
(V1) w/ Sub-Conv1x1	3.16M	35.28	36.63	41.58	34.64	29.12	36.31	29.91	34.78
(V2) w/ Sub-Conv3x3	3.15M	34.96	35.35	41.14	32.80	27.18	35.34	29.14	33.70
(V3) w/o Encoding	3.17M	35.64	39.18	41.58	36.16	30.92	36.68	31.50	35.95
(V4) w/ Fourier enc.	3.17M	35.62	39.07	41.59	36.00	30.91	36.81	31.47	35.92
(V5) w/ Fourier (local) enc.	3.17M	35.59	38.99	41.54	35.77	30.61	36.57	31.30	35.77
(V6) w/ Grid (local) enc.	3.19M	35.65	39.26	41.58	36.17	30.93	36.72	31.55	35.98
(V7) w/ MLP	3.19M	35.10	37.17	41.35	34.77	29.10	35.58	29.76	34.69
(V8) w/ Conv3x3	3.17M	35.35	37.86	41.37	35.13	29.70	36.10	30.31	35.12
(V9) w/ Frame-wise	3.17M	**35.68**	39.22	41.54	36.69	31.49	36.54	31.54	36.10
(V10) w/ Patch-wise	3.17M	35.46	38.30	41.55	35.04	30.06	36.51	30.77	35.38
(V11) w/ Nearest Neighbor	3.17M	35.60	39.12	**41.64**	36.52	31.51	**36.82**	31.33	36.08

NVRC: Neural video representation compression

HM Kwan, G Gao, F Zhang, A Gower, D Bull

Advances in neural information processing systems, 2024 • proceedings.neurips.cc

Abstract

Recent advances in implicit neural representation (INR)-based video coding havedemonstrated its potential to compete with both conventional and other learning-based approaches. With INR methods, a neural network is trained to overfit avideo sequence, with its parameters compressed to obtain a compact representationof the video content. However, although promising results have been achieved, the best INR-based methods are still out-performed by the latest standard codecs, such as VVC VTM, partially

SHOW MORE

Save Cite Cited by 28 Related articles All 6 versions

NVRC: Neural Video Representation Compression

Ho Man Kwan†, Ge Gao†, Fan Zhang†, Andrew Gower‡, David Bull†
† Visual Information Lab, University of Bristol, UK
‡ Immersive Content & Comms Research, BT, UK
{hm.kwan, ge1.gao, fan.zhang, dave.bull}@bristol.ac.uk,
andrew.p.gower@bt.com

Contribution

- Feature grids and network weights use different quantization strategies
  - Feature grids use learned per-block, per-channel quantization scales δ
    - We quantize weights using a scale δ $\hat{w} = \text{round}(w/\delta)$
    - one scale for all weights? -- Cannot adapt to different weight distributions
    - Per-weight δ? – δ itself is also a parameter!
  - Feature grids are modeled as spatio-temporal structured latents, encoded with context-based Gaussian entropy model $z[i,j,k] \sim \mathcal{N}(\mu, \sigma)$
  - Network Weight Compression $\delta[i,j] = \delta_{out}[i] \cdot \delta_{in}[j]$ AxB A+B
- Rate-Distortion loss:
  - Loss = R + λD, where D stands for distortion(reconstruction loss) and rate stands for the consumed bitrate (bits/pixel)

- δ=5 δ=1

- 10 15 11

Neural video compression with context modulation

[PDF] thecvf.com

C Tang, Z Li, Y Bian, L Li, D Liu

Proceedings of the Computer Vision and Pattern Recognition ..., 2025 · openaccess.thecvf.com

Abstract

Efficient video coding is highly dependent on exploiting the temporal redundancy, which is usually achieved by extracting and leveraging the temporal context in the emerging conditional coding-based neural video codec (NVC). Although the latest NVC has achieved remarkable progress in improving the compression performance, the inherent temporal context propagation mechanism lacks the ability to sufficiently leverage the reference information, limiting further improvement. In this paper, we address the limitation

SHOW MORE

Neural Video Compression with Context Modulation

Chuanbo Tang Zhuoyuan Li Yifan Bian Li Li Dong Liu

MOE Key Laboratory of Brain-Inspired Intelligent Perception and Cognition

University of Science and Technology of China, Hefei 230027, China

{cbtang,zhuoyuanli}@mail.ustc.edu.cn, togelbian@gmail.com, {lill,dongeliu}@ustc.edu.cn

VideoINR: Learning Video Implicit Neural Representation for Continuous Space-Time Super-Resolution

Zeyuan Chen[1] Yinbo Chen[2] Jingwen Liu[2] Xingqian Xu[3,6] Vidit Goel[6]
Zhangyang Wang[5] Humphrey Shi[6,5,3†] Xiaolong Wang[2†]
[1]USTC [2]UC San Diego [3]UIUC [4]UT Austin [5]U of Oregon [6]Picsart AI Research (PAIR)

Abstract
Videos typically record the streaming and continuous visual data as discrete consecutive frames. Since the storage cost is expensive for videos of high fidelity, most of them are stored in a relatively low resolution and frame rate. Recent works of Space-Time Video Super-Resolution (STVSR) are developed to incorporate temporal interpolation and spatial super-resolution in a unified framework. However, most of them only support a fixed up-sampling scale, which limits their flexibility and applications. In this work, instead

- Video: f(x,y,t)=(R,G,B)
  - super-resolution
  - Interpolation

- Task: given two frames I_0, I_1, we want to produce an interpolated picture with any resolution at any T

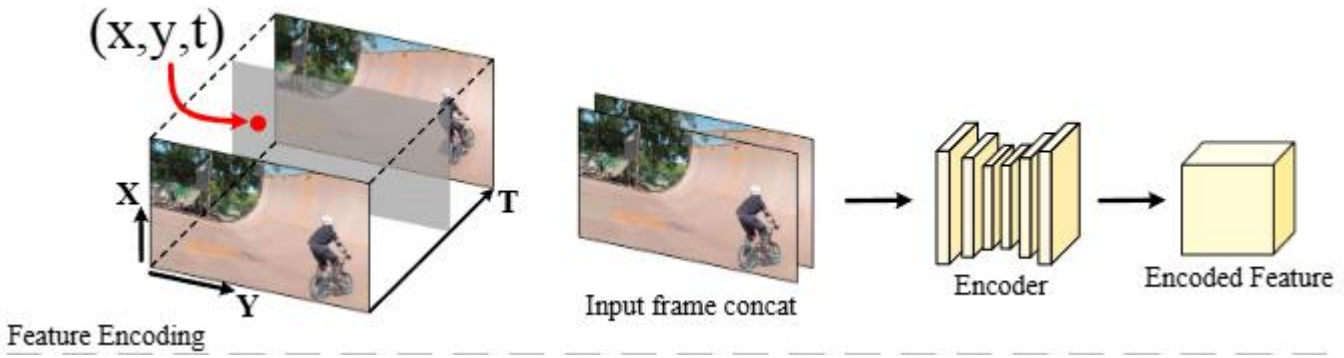
(256,256,5)

Steps

Query (122.155, 122.155,3)

(122, 122,3, 0.155, 0.155)

- Encoding: I_0, I_1 → CNN encoder → feature grid



- Continuous Spatial Representation: F(x, y) → feature **learn the position**

$$\mathcal{F}_s(x_s) = f_s(z^*, x_s - v^*), \qquad (2)$$

where $\mathcal{F}_s$ is the continuous feature domain defined by SpatialINR, z^* is the feature vector nearest to the query coordinate x_s and v^* is the spatial coordinate of the feature vector z^*.

Steps

- Continuous Temporal Representation: **learn the motion (flow)**
  - flow = TemporalINR(f, t)=(dx)
  - Updated coordinates: x' = x + dx

$$\mathcal{M}(x_s, x_t) = f_t(x_t, \mathcal{F}_s(x_s)), \qquad (4)$$

where $\mathcal{F}_s(x_s)$ is the feature domain defined in Eq 2.

- Space-Time Continuous Representation
  - f' = F(x')
  - RGB = decoder(f’)

  - RGB(x,t) = decoder(F(x + flow(x,t)))

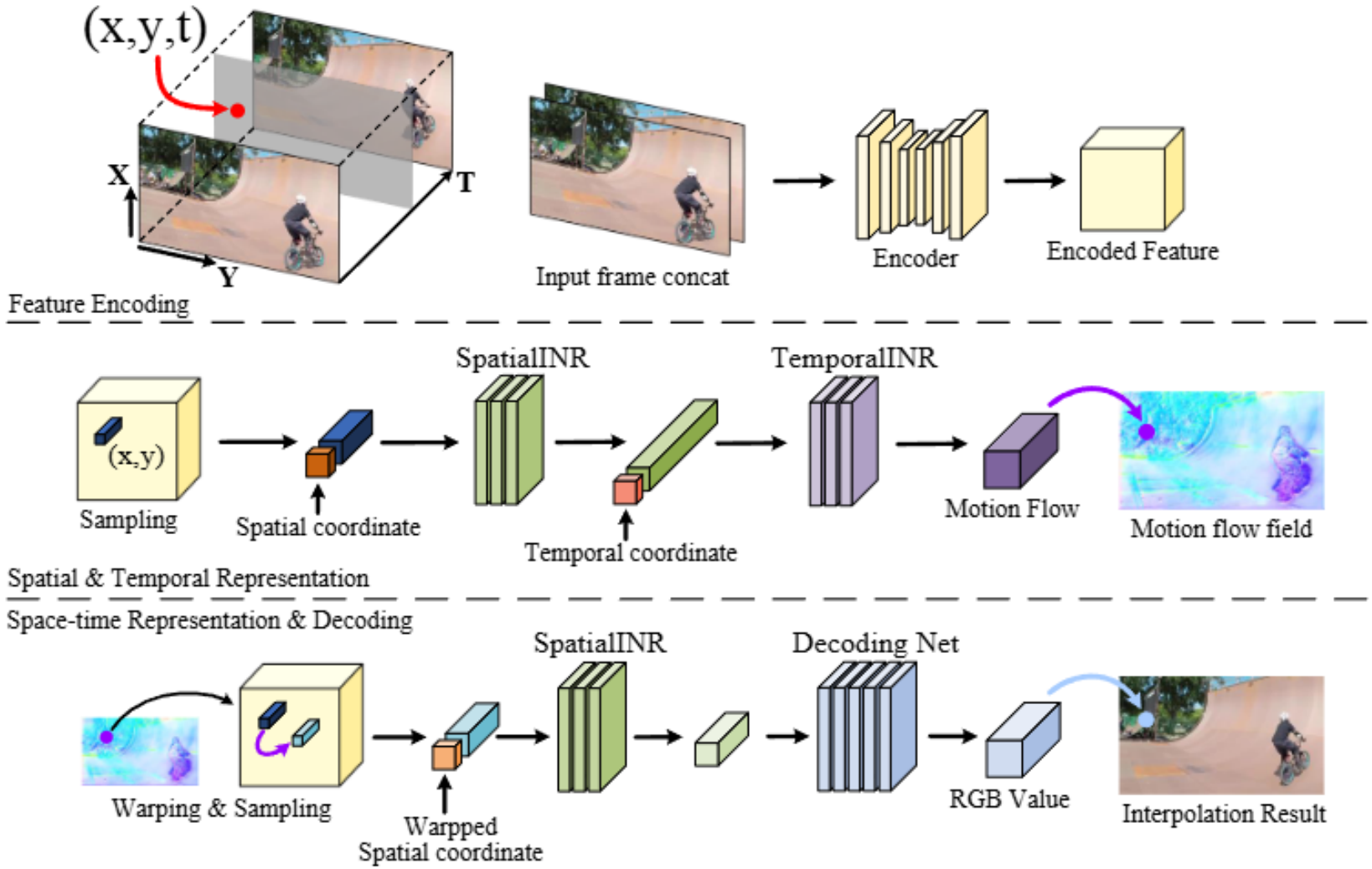


Figure 2. **An overview of our Video Implicit Neural Representation (VideoINR).** Two input frames are concatenated and encoded as a discrete feature map. Based on the feature, the spatial and temporal implicit neural representations decode a 3D space-time coordinate to a motion flow vector. We then sample a new feature vector by warping according to the motion flow, and decode it as the RGB prediction of the query coordinate. We omit the multi-scale feature aggregation part in this figure.