PNeRV: Enhancing Spatial Consistency via Pyramidal Neural Representation for Videos

Supplementary Material

A. More Discussions of Universal Approximation Theory (UAT) Analysis on NeRV

We provide more analysis and discussions of UAT analysis on the NeRV system. We define the problem that current NeRV systems are attempting to address and provide a comparison with existing video neural coding pipelines.

A.1. Implicit Neural Video Coding Problem

Following the pipeline of Implicit Neural Video Coding (INVC) presented in Sec. 4.2 , we recall the proposed Implicit Neural Video Coding Problem (INVCP) as follows.

Problem A.1. *(INVC Problem). The goal of Implicit Neural Video Coding is to find out the optimal design of the decoder $\mathcal{D}$ and encoder $\mathcal{E}$ in pursuit of minimal parameter quantity $\mathsf{Param}(\mathcal{D})$ and embeddings $\{e_t = \mathcal{E}(t) \in \mathbb{R}^{d_{in}^t}\}_{t=1}^T$ (where $d = d_{in}^t$ is often the same for all t in existing NeRV systems) under a certain approximation error ϵ between the reconstruction $\tilde{V}$ and a given video sequence V,*

$$\arg\min_{\mathcal{D},\mathcal{E}} \ \mathsf{Param}\left(\mathcal{D}\right) + \sum_{t=1}^{T} d_{in}^t,$$
$$\text{s.t. } L_{\mathcal{D}}, w_{\mathcal{D}} \in [1,\infty), \ \sup \sum \|\tilde{V}_t - V_t\| \leq \epsilon, \ t \in [1,T].$$

In the practice of INVC research, we usually use the dual problem of A.1 to determine the optimal architecture of a model to achieve a certain level of accuracy for fitting the video. We name it the Dual Implicit Neural Video Coding Problem (DINVCP).

Problem A.2. *(Dual INVC Problem). Given a certain parameter quantity μ, the Dual INVC problem aims to determine the optimal design of decoder $\mathcal{D}$ and encoder $\mathcal{E}$ to minimize the minimal approximation error between the reconstruction $\tilde{V}$ and the given video sequence V,*

$$\arg\min_{\mathcal{D},\mathcal{E}} \ \sup \sum \|\tilde{V}_t - V_t\|,$$
$$\text{s.t. } L_{\mathcal{D}}, w_{\mathcal{D}} \in [1,\infty), \mathsf{Param}\left(\mathcal{D}\right) + \sum_{t=1}^{T} d_{in}^t \leq \mu, \ t \in [1,T].$$

In practice, when using a NeRV model to represent a given video within a certain model size limit μ through end-to-end training, it is trying to solve the DINVCP.

A.2. Comparison between DINVCP and Previous Neural Coding Pipelines

Distribution-Preserving Lossy Compression (DPLC) is proposed by [30] motivated by GAN-based image compression [2]. It is defined as follows:

$$\min_{E,D} \ \mathbb{E}_{X,D}[d(X, D(E(X)))] + \lambda d_f(p_X, p_{\tilde{X}}),$$

where $E, D, X, \tilde{X}$ are encoder, decoder, given input and reconstruction, d_f is a divergence which can be estimated from samples. DPLC emphasizes the importance of maintaining distribution consistency for effective compression and reconstruction.

[26] proposes Rate-Distortion Optimization (RDO). Later, [3] reveals the importance of perceptual quality and proposes the Perception-Distortion Optimization (PDO) as

$$\min_{p_{\tilde{X}|Y}} \ d(p_X, p_{\tilde{X}}) \ \ s.t. \ \mathbb{E}[\Delta(X, \tilde{X})] \leq D,$$

where Δ is the distortion measure and d is the divergence between distributions. Furthermore, [4] defines the Rate-Distortion-Perception Optimization (RDPO) as

$$\min_{p_{\tilde{X}|X}} I(X, \tilde{X}) \ \ s.t. \ \mathbb{E}[\Delta(X, \tilde{X})] \leq D, \ d(p_X, p_{\tilde{X}}) \leq P,$$

where I denotes mutual information.

The primary objective, which also serves as the main obstacle in the aforementioned pipelines, is that density estimation is not only costly but also challenging to estimate accurately. Different from DPLC, PDO, or RDPO, *DINVCP does not need to model the distribution of the given signal explicitly.* In fact, the distribution of input images or videos is difficult to approximate. Whether it is approached by minimizing ELBO or through adversarial training [8, 13], there is always a certain gap or mismatch. Besides, other density estimation methods, such as flow-based or diffusion models, suffer from huge computational costs [11, 23]. In contrast, *NeRV system implicitly models the unknown distribution of a given signal via specific decoding computation process under certain model parameter quantity constraints. The calculation process per se is regarded as the side information* [12, 34].

This approach of implicitly modeling distributions through computational processes under parameter quantity constraints aligns with some current perspectives that suggest the intelligence of Large Language Models (LLM) emerges from data compression [7, 22]. LLMs such as GPT

aim to transfer as much data as possible to models of the same size for learning (and continue to increase the model size after learning) to achieve information compression and efficient information coding. However, the NeRV system strives to compress the model size as much as possible for a given video, emerging with robust representations with generalized capability.

The improvement of PNeRV in terms of perceptual quality confirms this conjecture. By upgrading the model structure and training with only MSE loss, PNeRV emerges better perceptual performance without having to estimate the signal's unobtainable prior distribution.

A.3. Proof of Theorem 1

Following the definitions given in Sec. 4, the width w of $\mathcal{N}$ is named as $\max d_i, \{d_i \in \mathbb{N}\}_{i=1}^{L}$. Once the minimal width $w^* = w_{min}\left(d_{in}, d_{out}\right)$ is estimated by d_{in}, d_{out}, such that, for any continuous function $f : [0,1]^{d_{in}} \to \mathbb{R}^{d_{out}}$ with $\epsilon \geq 0$, there exists a $\mathcal{N}$ with input dimension d_{in}, hidden layer widths at most w^*, and output dimension d_{out} that $\epsilon-$approximates f:

$$\sup_{x \in [0,1]^{d_{in}}} \| f\left(x\right) - \mathcal{N}\left(x\right) \| \leq \epsilon.$$

The goal of Theorem 1 is to determine the minimum parameter demand when $\epsilon-$approximates the implicit $\mathcal{F}$ which represents the given video. We recall Theorem 1 as Theorem A.1 as follows for better illustration.

Theorem A.1. *For a cascaded NeRV system to ϵ-approximate a video V which is implicitly characterized by a certain unknown L-Lipschitz continuous function $\mathcal{F} : K \to \mathbb{R}^{d_{out}}$ where $K \subseteq \mathbb{R}^{d_{in}}$ is a compact set, then the upper bound of the minimal parameter quantity $\mathsf{Param}(\mathcal{D})$ is given as*

$$\mathsf{Param}_{\min}(\mathcal{D}) \leq d_{out}^2 \left(\frac{O\left(diam\left(K\right)\right)}{\omega_{\mathcal{F}}^{-1}\left(\epsilon\right)} \right)^{d_{in}+1}.$$

Before we start, we will recall the setup and demonstrate some mathematic concepts and lemmas.

Definition A.1. A function $g : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ is a max-min string of length $L \geq 1$ on d_{in} input variables and d_{out} output variables if there exist affine functions $\ell_1, \ldots, \ell_L : \mathbb{R}^{d_{in}} \to \mathbb{R}^{d_{out}}$ such that

$$g = \sigma_{L-1}(\ell_L, \sigma_{L-2}\left(\ell_{L-1}, \cdots, \sigma_2\left(\ell_3, \sigma_1\left(\ell_1, \ell_2\right)\right)\cdots\right).$$

The definition of max-min string and DMoC (Def. 3) are first introduced in [9] and [10]. We introduce two lemmas, which were presented as Propositions 2 and 3 in [10].

Lemma A.1. *[10] For every compact $K \subseteq \mathbb{R}^{d_{in}}$, any continuous $f : K \to \mathbb{R}^{d_{out}}$ and each $\epsilon \geq 0$, there exists a max-min string g on d_{in} input variables and d_{out} output variables with length*

$$\left(\frac{O\left(diam\left(K\right)\right)}{\omega_f^{-1}\left(\epsilon\right)} \right)^{d_{in}+1},$$

for which

$$\|f - g\|_{C^0(K)} \leq \epsilon.$$

Lemma A.2. *[10] For every max-min string g on d_{in} input variables and d_{out} output variables with length L and every compact $K \subseteq \mathbb{R}^{d_{in}}$, there exists a* RELU *net $\mathcal{N}$ with input dimension d_{in}, hidden layer width $d_{in} + d_{out}$, and depth L that computes $x \mapsto g(x)$ for every $x \in K$.*

Lemma A.3. *[21] For any $p \in [1, \infty)$,* RELU *nets of width w are dense in $L^P(\mathbb{R}^{d_{in}}, \mathbb{R}^{d_{out}})$ if and only if $w \geq \max\{d_{in}+1, d_{out}\}$.*

The proofs of Lemma A.1 and A.2 can be found in the Sec. 2.1 and Sec. 2.2 of [10]. Lemma A.3 is the Theorem 1 demonstrated in [21] with its proof. Now we provide the proof of Theorem A.1 as follows.

Proof. From Lemma A.1, the implicit function $\mathcal{F}_{\mathcal{V}}$ which represents the video $\mathcal{V}$ can be approximated by one max-min string g. It is worth mentioning that $\mathcal{F}_{\mathcal{V}}$ is supposed to be continuous because video can be considered as a slice of the real world. The length of this max-min string g is given by Lemma A.1. According to Lemma A.2, there exists a RELU net $\mathcal{N}_g$ with the same input and output dimensions that fit this max-min string. So, the minimal parameters of $\mathcal{N}_g$, also the sum of weights for each layer, is

$$\mathbf{Param} = \sum_{l=1}^{L} w_l w_{l-1},$$

where w_l is width in each hidden layer and L is given in Lemma A.1. Noticed that the *whole width* w of a model is the upper bound of all hidden layer widths $\{w_l\}_{l=0}^{L}$. $w_{\min}$ is the minimum estimate for this upper bound, $w_l \leq w_{\min} \leq w$. $w_{\min}$ is further contracted from $d_{in}+d_{out}$ to $\max\{d_{in}+1, d_{out}\}$ by [21] (Lemma A.3).

Thus, the minimal parameters of $\mathcal{N}_g$ under a certain error is no longer than

$$\begin{aligned}\mathbf{Param}_{\min} &\leq w_{\min}^2 \left(\frac{O\left(diam\left(K\right)\right)}{\omega_f^{-1}\left(\epsilon\right)} \right)^{d_{in}+1} \\ &= d_{out}^2 \left(\frac{O\left(diam\left(K\right)\right)}{\omega_f^{-1}\left(\epsilon\right)} \right)^{d_{in}+1},\end{aligned}$$

where $w_{\min} = d_{out}$ for video $\mathcal{V} : \mathbb{N} \to \mathbb{R}^{d_{out}}$. Equality is reached when each layer width reaches the upper bound of minimal width, the worst case. □

Although the upper bound of $\mathsf{Param}(\mathcal{D})$ is fixed regardless of the detailed architecture, the actual performance of serial NeRV will be influenced by structure design, parameter initialization, activation functions, loss functions, and optimizer.

B. More Related Works

Comparison with Other Subpixel-based Upsampling Operators. The NeRV system aims at reconstructing high-resolution videos through decoding low-dim embeddings. Therefore, proper upsampling operators are crucial for its performance. Existing subpixel-based upsampling operators are not efficient enough for the NeRV system. Deconv [36] pads the subpixels with zeros and passes them through a Conv layer, resulting in block artifacts [20]. PixelShuffle [24] first expands the feature map channels through a CONV and then rearranges them into the target subpixels. However, the desired subpixels of a given position are only related to the expanding channels of the same position, ignoring contextual information, as shown in Fig. **??** of the main text. Additionally, PixelShuffle encounters an exponential explosion of required channels when the upsampling ratio is large.

Comparison with INR on Images. [25] (SIREN) uses sine as a periodic activation function to model the high-frequency information of a given image [29] and performs a sinusoidal transformation before input [35] tries to directly modify an INR without explicit decoding. The main difference between these methods and ours is that we consider the input coordinate-pixel pairs to be *dense* for the INR on image coding. In a natural image, the RGB value at a specific position is often closely related to its neighboring positions. However, for high-resolution videos, the gap between adjacent frames can be much larger, both in terms of pixels and semantic terms. This situation is akin to only observing partial pixels from a given image.

Comparison with Self-attention Module. Self-attention (SA) and Multi-head Self-attention (MSA) modules [17, 28, 31, 32] compute the response at a position by attending to all positions, which is similar to KFC. The major defect of SA and MSA when adopted in NeRV is that the computational complexity and the space complexity are too high to efficiently compute the global correlations between arbitrary positions, especially the computational cost ($O(n^2d)$) between queries and keys for high-resolution feature maps. KFC not only captures long-range dependencies but also achieves low-cost rescaling, both of which are significant for NeRV.

C. Additional Results

Unless otherwise specified, all models utilized in the additional results are trained on a 3M model for 300 epochs.

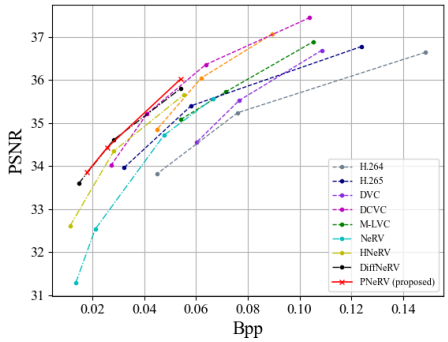

Figure C.1. PSNR of video compression on UVG.

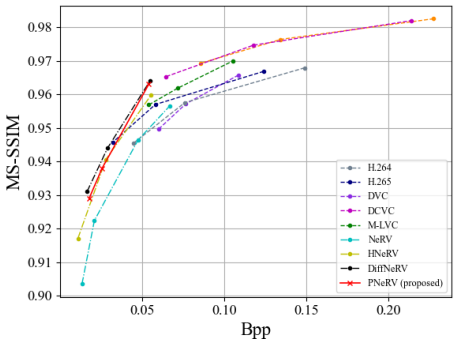

Figure C.2. SSIM of video compression on UVG.

C.1. Comparison of Generalization Ability by Video Interpolation Results

Indeed, the concepts of approximation and generalization are distinct topics within the field of deep learning theory [1, 19]. Understanding the causal relationship between overfitting and the generalization capacity of NeRV necessitates further investigation. Existing NeRV models always focus on the models' approximation capabilities through overfitting training.

Nonetheless, we also evaluate the generalization performance of our proposed PNeRV through a video interpolation experiment. Adhering to the experimental methodology employed in [6] and [37], the model is trained using odd-numbered frames and then tested with unseen even-numbered frames. The results, presented in Table C.1, indicate that PNeRV surpasses most baseline methods. Future research will focus on the theoretical analysis and enhancement of PNeRV's generalization abilities.

C.2. Comparison of Video Compression and Discussion of Training Difficulties

The video compression comparison of PNeRV with other NeRV models in terms of PSNR and MS-SSIM is shown in Fig. C.1 and Fig. C.2. Following the same settings utilized

	Beauty	Bospho	Honey	Jockey	Ready	Shake	Yacht	Avg.
NeRV [5]	28.05	30.04	36.99	20.00	17.02	29.15	24.50	26.54
E-NeRV [15]	27.35	28.95	38.24	19.39	16.74	30.23	22.45	26.19
H-NeRV [6]	31.10	34.38	38.83	23.82	20.99	32.61	27.24	29.85
DiffNeRV [37]	35.99	35.10	37.43	30.61	24.05	35.34	28.70	**32.47**
PNeRV	33.64	34.09	39.85	28.74	23.12	31.49	27.35	31.18

Table C.1. Video interpolation results on 960 × 1920 UVG in PSNR.

	Bmx-B	Camel	Dance-J	Drift-C	Elephant	Parkour	Scoo-G	Scoo-B	Avg.
HNeRV	20.39	21.85	21.73	28.81	17.35	19.97	**24.49**	19.76	21.79
DiffNeRV	**22.95**	23.72	21.78	**30.37**	26.02	21.55	22.78	21.00	23.77
PNeRV	21.69	**24.28**	**25.21**	30.01	**27.32**	**22.61**	22.84	**22.61**	**24.57**

Table C.2. Video inpainting results using center mask on 960 × 1920 DAVIS in PSNR.

	Bmx-B	Camel	Dance-J	Drift-C	Elephant	Parkour	Scoo-G	Scoo-B	Avg.
HNeRV	0.665	0.733	0.677	0.650	0.489	0.650	0.859	0.789	0.725
DiffNeRV	0.767	0.815	0.667	**0.949**	0.817	0.754	0.852	**0.844**	0.808
PNeRV	**0.802**	**0.844**	**0.792**	0.947	**0.862**	**0.801**	**0.874**	0.812	**0.842**

Table C.3. Video inpainting results using center mask on 960 × 1920 DAVIS in SSIM.

	Bmx-B	Camel	Dance-J	Drift-C	Elephant	Parkour	Scoo-G	Scoo-B	Avg.
HNeRV	23.16	20.94	26.54	31.70	17.36	21.32	**26.89**	21.05	23.62
DiffNeRV	**25.70**	**24.71**	26.59	34.74	25.93	24.51	26.61	**24.27**	**26.63**
PNeRV	24.96	24.18	**26.62**	**34.84**	**27.50**	**24.98**	26.85	22.13	26.51

Table C.4. Video inpainting results using disperse mask on 960 × 1920 DAVIS in PSNR.

	Bmx-B	Camel	Dance-J	Drift-C	Elephant	Parkour	Scoo-G	Scoo-B	Avg.
HNeRV	0.728	0.661	0.779	0.957	0.490	0.685	0.889	0.794	0.748
DiffNeRV	0.819	0.832	0.795	0.972	0.827	0.799	0.892	**0.897**	0.854
PNeRV	**0.843**	**0.854**	**0.806**	**0.975**	**0.877**	**0.836**	**0.910**	0.866	**0.871**

Table C.5. Video inpainting results using disperse mask on 960 × 1920 DAVIS in SSIM.

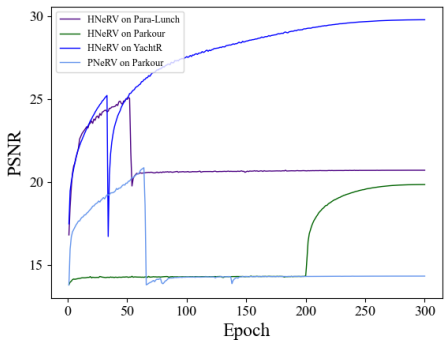

Figure C.3. Example of training difficulty of different NeRV methods in 3M size.

in [6, 37], we evaluate the video compression comparison with 8-bit quantization for both embeddings and the model without model pruning.

PNeRV has demonstrated remarkable performance, notably outperforming conventional encoding pipelines like H264 [33] and H265 [27], and possesses substantial advantages over several traditional neural video coding models [14, 16, 18], particularly at low bit rates. Compared to INR-based methods, PNeRV has also achieved competitive results and outperforms other NeRV methods [5, 6, 37] in terms of PSNR.

For detailed experimental settings, PNeRV adjusts the size of the decoder and the dimensions of the input diff embedding to validate the encoding performance of the proposed method across various bit rates. At low bit rates, the encoding performance of the model may experience some

degradation. We believe this is due to the diversity and complexity of the modules required by PNeRV. Maintaining a certain amount of parameters (such as the number of channels in convolutional layers) is crucial for preserving performance. This ensures that the model has sufficient capacity to handle the challenges posed by low-bit rate encoding.

It is worth noting that all implicit models encounter significant training challenges when dealing with large parameters, such as those exceeding 5M. As a result, these models often converge to local minima, leading to trivial outputs. This issue poses a significant obstacle to the compression performance of all NeRV methods, particularly when the Bpp value increases. Some examples of training failure are shown in Fig. C.3, where models are 3M under the same conditions.

C.3. Comparison of Robustness by Video Inpainting Results

We evaluate the robustness of different methods using video inpainting tasks following the same setting as in [6] and [37], which use a center mask and disperse mask. The center mask uses a rectangular area that occupies one-fourth of the width and height of the original frame, positioned at its center. The disperse mask comprises five square areas, each measuring 100×100 pixels, positioned in the four corners and the center of the frame. The pixel value of areas in the masks is reset to 0. The trained models in video regression tasks will be directly utilized for inpainting without any fine-tuning. Models take the masked frames as input and try to predict the original ones.

The results using the center mask are provided in Tab. C.2 and Tab. C.3. The dispersed ones are in Tab. C.4 and Tab. C.5. PNeRV acquires competitive results with both the center mask and the disperse mask, indicating robust modeling capability.

C.4. More Visualization Examples for Perceptual Quality

We show some more examples of qualitative comparisons between different models.

Shown in Fig. D.4, the results of PNeRV are smoother and less noisy. For instance, in "Lucia" and "Horse-low", PNeRV pays more attention to the geometric pattern of the main objects and ignores those high-frequency details of the background scene. Other baseline methods cannot reconstruct frames at such a semantic level. Due to the lack of high-level information guidance and a global receptive field, baseline methods are hard to reasonably allocate model weights to more important objects, e.g., red waterpipe in "Breakdance-flare" and patterns in "Cows".

Shown in Fig. D.5, the comparison at different timestamps of the same video indicates some specific common issues of different models. Overlapping and noisy patterns have occurred in the results of DiffNeRV [37] and HNeRV [6], such as the grass and hands in "Hike". ENeRV [15] and NeRV [5] often result in color deviation and blurring, e.g., backpack in "Hike" and motor in "motorbump". PNeRV achieves a balance between preserving details and maintaining semantic consistency. Compared to DiffNeRV, which also uses the difference between frames as input, the latter's reconstruction of details is unbiased. However, human attention to visual elements under different semantics should be different. Improving the reconstruction results through high-level information is one of PNeRV's pursuits.

C.5. Discussion on the Failure Cases

As shown in Table 2 , PNeRV fails in the "Dog" which is blurred and mixed with jitter and deformation. Also, the "Soapbox" video, which comprises two clips from entirely different scenes connected by a few frames where the camera rotates through a large angle, poses a challenge. So far, PNeRV has not been able to handle severe temporal inconsistency effectively.

C.6. Video Examples

We provide some video examples from DAVIS as follows. From the video comparison, it can be seen that the reconstructions of NeRV have lost spatial details, and it is difficult for DNeRV to reconstruct videos containing pervasive scattered high-frequency details. Whether there is large motion or high-frequency details in the given videos, PNeRV is more robust in modeling the spatial consistency, leading to better perceptual quality in reconstructions. The links to the examples are presented as follow.

Dance-jump: https://drive.google.com/file/d/18JZq1BCkBJWCkZs-71OB7wI6j_Vma0vP/view?usp=drive_link

Elephant: https://drive.google.com/file/d/1rnPEsEtfA5UADU6BnwEDOPRG9hO9uPuM/view?usp=drive_link

Kite-surf: https://drive.google.com/file/d/1DDGw1zc2iJWcJHdBS4DOnfUQVf2H04Bs/view?usp=drive_link

Parkour: https://drive.google.com/file/d/1jWbJuoc-GCz2N_dXAJSER0PSy7ThrMr-/view?usp=drive_link

Scooter-grey: https://drive.google.com/file/d/1vs22Ru-AwAQuG710qbF72lwdHS1ABy83/view?usp=drive_link

D. Additional Ablation Studies

D.1. Ablation Results of Model Structure Details

We ablate the structure details of PNeRV in 3M on "Rollerblade" in 480×960 from DAVIS, given in Tab. C.7,

Models	Bmx-B	Camel	Dance-J	Dog	Drift-C	Parkour	Soapbox	Avg.	A.P.G
NeRV [5]	29.42/0.864	24.81/0.781	27.33/0.794	28.17/0.795	36.12/0.969	25.15/0.794	27.68/0.848	28.38/0.835	-
E-NeRV [15]	28.90/0.851	25.85/0.844	29.52/0.855	30.40/0.882	39.26/0.983	25.31/0.845	28.98/0.867	29.75/0.875	-
HNeRV [6]	29.98/0.872	25.94/0.851	29.60/0.850	30.96/0.898	39.27/0.985	26.56/0.851	29.81/0.881	30.30/0.874	-
DiffNeRV [37]	30.58/0.890	27.38/0.887	29.09/0.837	**31.32/0.905**	40.21/0.987	25.75/0.827	**31.47/0.912**	30.84/0.892	-
Ablation Study									
Bilinear + Concat	24.85/0.783	24.49/0.793	28.32/0.806	26.19/0.723	31.92/0.943	25.09/0.793	29.23/0.872	27.16/0.816	-4.07
Bilinear + GRU	29.86/0.874	25.00/0.811	29.16/0.830	27.11/0.753	32.09/0.945	26.43/0.845	29.10/0.874	28.39/0.847	-2.84
Bilinear + LSTM	26.22/0.792	26.87/0.871	27.85/0.788	26.71/0.741	33.65/0.946	25.82/0.820	29.42/0.881	28.07/0.834	-3.16
Bilinear + **BSM**	29.97/0.877	27.35/0.881	29.49/0.838	27.14/0.756	34.34/0.968	26.15/0.835	29.14/0.876	29.08/0.862	-2.15
DeConv + Concat	28.06/0.840	24.07/0.774	27.86/0.792	25.16/0.693	34.97/0.961	22.13/0.683	29.33/0.877	27.37/0.803	-3.86
DeConv + GRU	27.52/0.827	28.16/0.900	29.09/0.825	25.76/0.706	37.91/0.980	25.09/0.793	29.54/0.882	29.00/0.845	-2.23
DeConv + LSTM	30.15/0.882	26.49/0.859	28.30/0.805	25.94/0.712	34.91/0.956	26.35/0.842	30.26/0.895	28.91/0.850	-2.32
DeConv + **BSM**	31.56/0.906	27.18/0.878	29.77/0.847	30.09/0.868	36.03/0.971	26.09/0.831	29.00/0.872	29.96/0.881	-1.27
KFc + Concat	27.51/0.826	25.02/0.816	29.02/0.831	28.80/0.831	36.82/0.974	25.12/0.796	28.53/0.864	28.68/0.848	-2.55
KFc + GRU	**31.69/0.910**	25.88/0.848	28.32/0.805	28.47/0.813	33.25/0.942	26.68/0.853	30.89/0.903	29.31/0.868	-1.92
KFc + LSTM	29.16/0.862	27.24/0.878	28.90/0.825	29.28/0.842	32.73/0.935	26.62/0.839	29.35/0.879	29.04/0.866	-2.19
KFc + BSM (PNeRV)	31.05/0.896	**27.89/0.892**	**30.45/0.873**	31.08/0.898	**40.23/0.987**	**27.08/0.867**	30.85/0.902	**31.22/0.902**	+0

Table C.6. Ablation results on DAVIS subset in PSNR and MS-SSIM, where Avg. is the average PSNR and A.P.G is the average PSNR gap. Every result is reported by corresponding model trained in 300 epoch and 3M size.

	40×80	20×40	10×20
PSNR	**31.94**	31.33	30.50
SSIM	**0.960**	0.954	0.947

Table C.7. Embedding size in PNeRV-L.

	1×1	3×3	5×5
PSNR	31.92	**31.94**	31.90
SSIM	0.960	0.960	**0.961**

Table C.8. Kernel size in BSM.

	ReLU	Leaky	GeLU	w/o BN
PSNR	31.80	31.86	**31.94**	31.53
SSIM	0.959	**0.961**	0.960	0.959

Table C.9. Activation and BN in KFc.

Tab. C.8 and Tab. C.9. The alternation of kernel size or activation has little influence. Encoding more information into embeddings will help the decoder reconstruct better and also increase the overall size.

D.2. Ablation Results of Proposed Modules on DAVIS

To verify the contribution of different modules in PNeRV, we conduct ablation studies on (1) upscaling operators and (2) gated memory mechanisms. We compare KFc with two upscaling layers, Deconv and Bilinear, where "Deconv" is implemented by "nn.ConvTranspose2d" from PyTorch, and "Bilinear" is the combination of bilinear upsampling and Conv2D. KFc achieves better performance due to the global receptive field regardless of what fusion module it is combined with.

Also, to illustrate the importance of adaptive feature fusion and improvement of BSM, we compare BSM with Concat, GRU and LSTM, where "Concat" means directly concatenating two features from different domains together. The ablation results suggest that the adaptive fusion of features from different domains significantly improves performance, and BSM outperforms other memory cells due to the disentangled feature learning. The last row is the final PNeRV and the last column shows PSNR gaps when changing modules in PNeRV.

D.3. Visualization of Feature Maps

To verify the effectiveness of hierarchical information merging via KFc and BSM, we visualize some feature maps in PNeRV-L which was pretrained on "Parkour" as examples. Those feature maps shown in Fig. D.6 are from different channels and layers using the same frame as input. Those in Fig. D.7 are all from the 4-th layer but using different frames as input. The feature maps from 4-th layer are in 480×960, and the original frames are in 960×1920. For each lower layer, the height and width are halved compared to the upper layer. "Before" and "After" refer to the feature maps before and after passing through BSM or after.

Fig. D.6 illustrates how the coarse features are refined by BSM. Different channels respond to distinct spatial patterns of video frames, including factors like color, geometric structure, texture, brightness, motion, and so on. Before being processed by the BSM, the vanilla features are semantically mixed and entangled. However, the BSM is able to decouple these features and distinguish their specific effects, resulting in more refined and distinct outputs.

Additionally, for imperfect feature maps, BSM can add details or balance the focus of the reconstruction across various areas in the frames. These phenomena are commonly observed in the 4-th layer, which is responsible for preparing for fine-grained reconstruction, as demonstrated in Figure. D.7. This shows the effectiveness of BSM in enhancing the quality of feature maps and improving the overall reconstruction.

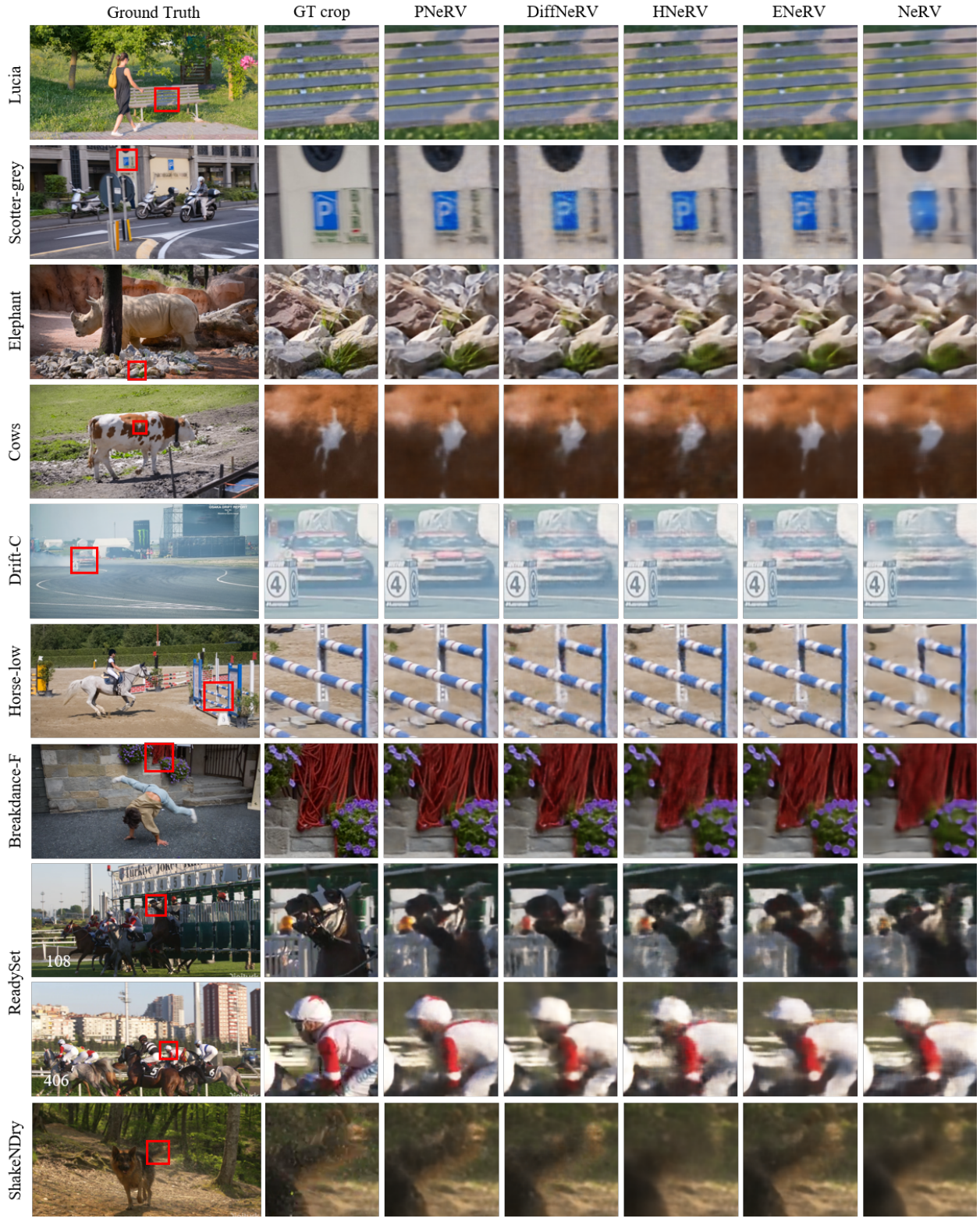

Figure D.4. Visual comparison examples on various videos.

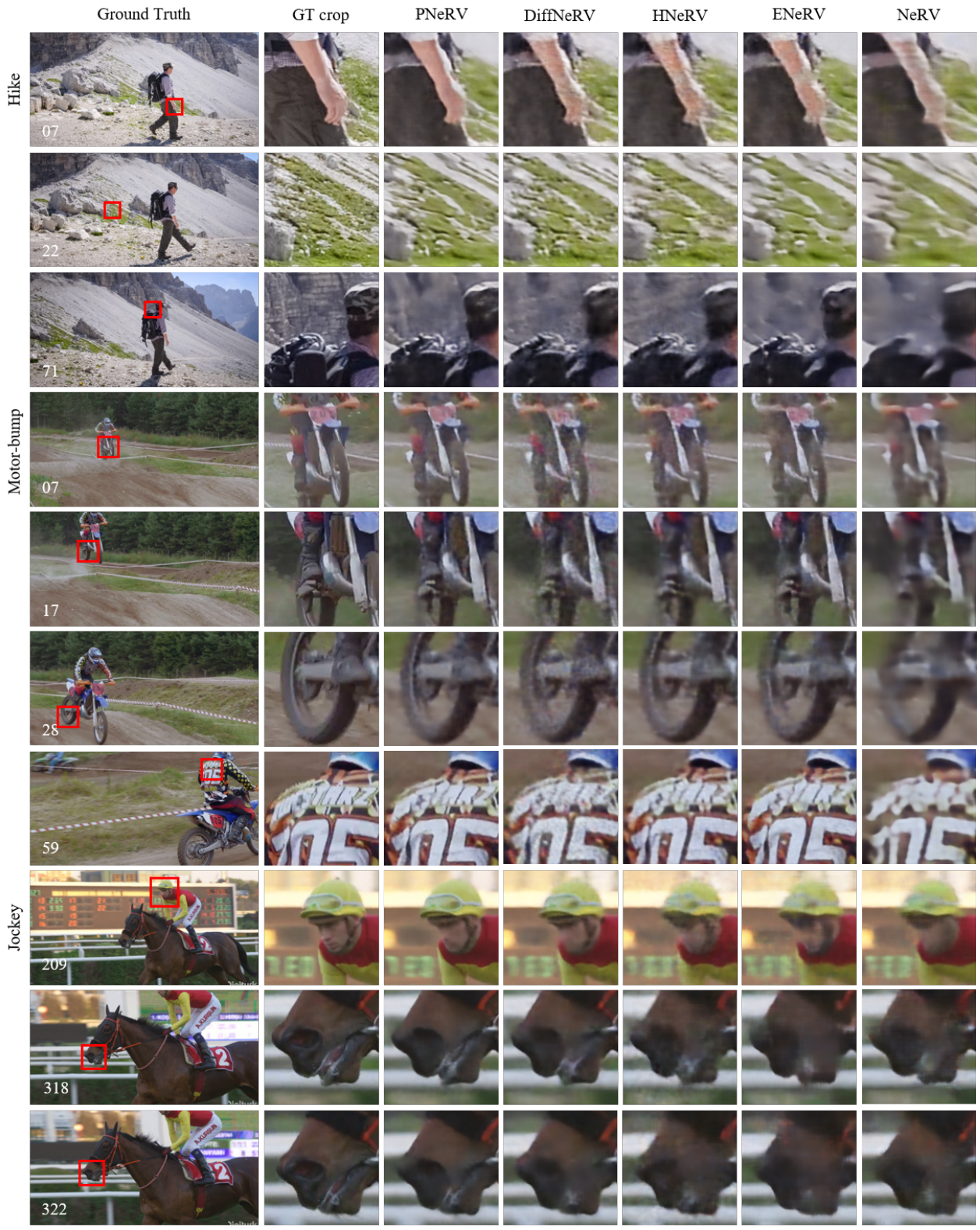

Figure D.5. Visual comparison examples on the same video by same models. Corresponding time stamps are shown in the bottom left.

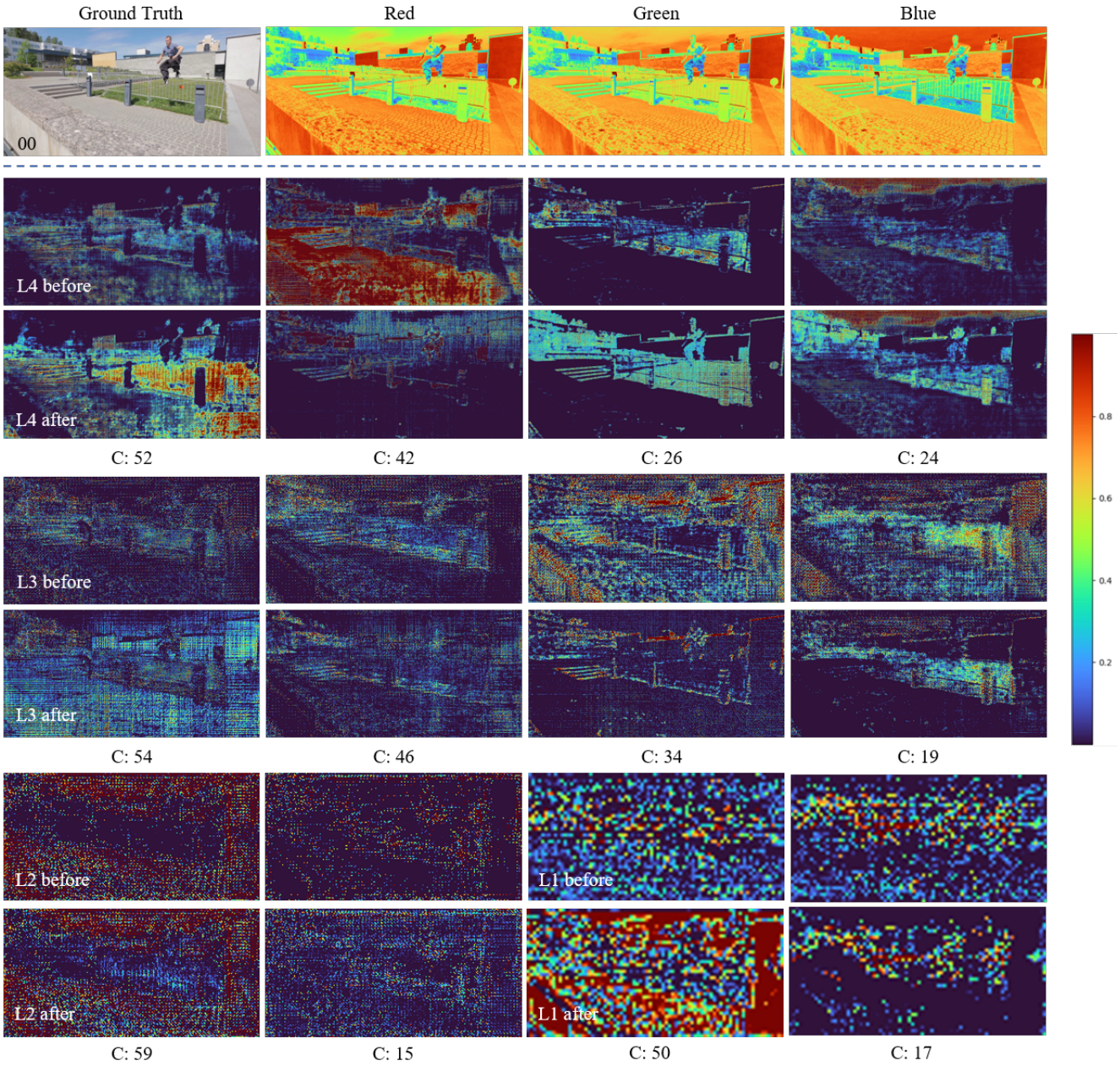

Figure D.6. Visualization examples of feature maps in different layers."C" refers to the channel number and "L" is the layer number.

References

[1] Madhu S. Advani and Andrew M. Saxe. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 2017. 3

[2] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2018. 1

[3] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Computer Vision and Pattern Recognition*, 2017. 1

[4] Yochai Blau and Tomer Michaeli. Rethinking lossy compression: The rate-distortion-perception tradeoff. *Proceedings of the 36th International Conference on Machine Learning*, 2019. 1

[5] Hao Chen, Bo He, Hanyu Wang, Yixuan Ren, Ser-Nam Lim, and Abhinav Shrivastava. Nerv: Neural representations for videos. In *NeurIPS*, 2021. 4, 5, 6

[6] Hao Chen, M. Gwilliam, Ser Nam Lim, and Abhinav Shrivastava. Hnerv: A hybrid neural representation for videos.

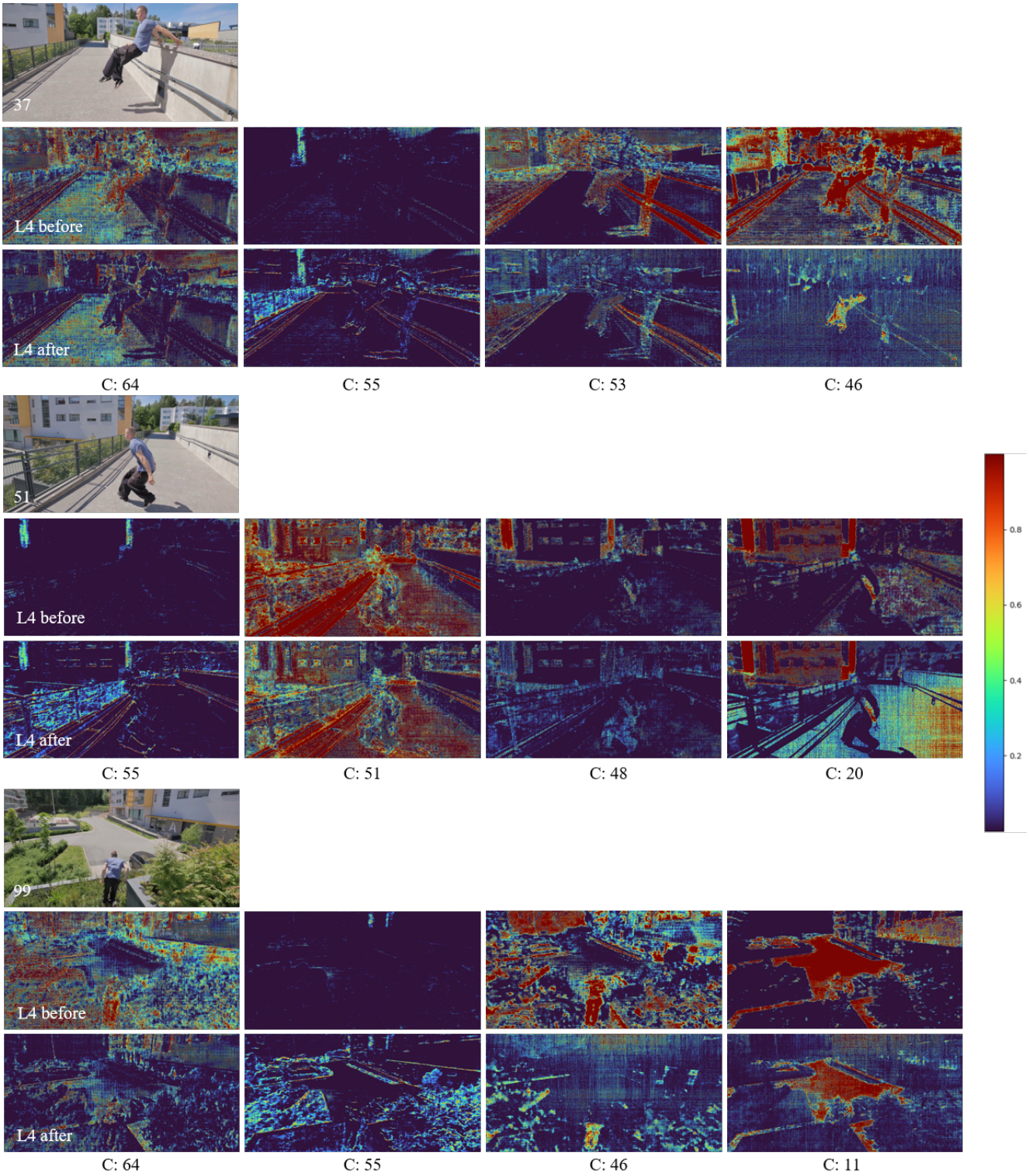

Figure D.7. Visualization examples of feature maps for different frames."C" refers to the channel number and "L" is the layer number.

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023. 3, 4, 5, 6

[7] Gr'egoire Del'etang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern,

Jordi Grau-Moya, Wenliang Kevin Li, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. Language modeling is compression. *ArXiv*, abs/2309.10668, 2023. 1

[8] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2014. 1

[9] Boris Hanin. Universal function approximation by deep neural nets with bounded width and relu activations. *ArXiv*, abs/1708.02691, 2017. 2

[10] Boris Hanin and Mark Sellke. Approximating continuous functions by relu nets of minimal width. *ArXiv*, abs/1710.11278, 2017. 2

[11] Jonathan Ho, Ajay Jain, and P. Abbeel. Denoising diffusion probabilistic models. *ArXiv*, abs/2006.11239, 2020. 1

[12] Rico Jonschkowski, Sebastian Hofer, and Oliver Brock. Patterns for learning with side information. *arXiv: Learning*, 2015. 1

[13] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013. 1

[14] Jiahao Li, Bin Li, and Yan Lu. Deep contextual video compression. *NeurIPS*, 2021. 4

[15] Zizhang Li, Mengmeng Wang, Huaijin Pi, Kechun Xu, Jianbiao Mei, and Yong Liu. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. *arXiv:2207.08132*, 2022. 4, 5, 6

[16] Jianping Lin, Dong Liu, Houqiang Li, and Feng Wu. M-lvc: Multiple frames prediction for learned video compression. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 4

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 3

[18] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. 2019. 4

[19] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: where bigger models and more data hurt. *Journal of Statistical Mechanics: Theory and Experiment*, 2019. 3

[20] Augustus Odena, Vincent Dumoulin, and Christopher Olah. Deconvolution and checkerboard artifacts. 2016. 3

[21] Sejun Park, Chulhee Yun, Jaeho Lee, and Jinwoo Shin. Minimum width for universal approximation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. 2

[22] Jack Rae. Compression for agi. *YouTube*, https://youtu.be/dO4TPJkeaaU, 2023. 1

[23] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 1

[24] Wenzhe Shi and Jose Caballero et al. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 3

[25] Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 3

[26] Gary J. Sullivan and Thomas Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Process. Mag.*, 1998. 1

[27] Gary J. Sullivan, Jens-Rainer Ohm, Woojin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.*, 2012. 4

[28] Mukund Varma T, Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, and Zhangyang Wang. Is attention all that neRF needs? In *The Eleventh International Conference on Learning Representations*, 2023. 3

[29] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *NeurIPS*, 2020. 3

[30] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. Deep generative models for distribution-preserving lossy compression. In *Neural Information Processing Systems*, 2018. 1

[31] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 3

[32] X. Wang, Ross Girshick, Abhinav Kumar Gupta, and Kaiming He. Non-local neural networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 3

[33] Thomas Wiegand, Gary J. Sullivan, Gisle Bjøntegaard, and Ajay Luthra. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.*, 2003. 4

[34] Aaron D. Wyner and Jacob Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inf. Theory*, 1976. 1

[35] Dejia Xu, Peihao Wang, Yifan Jiang, Zhiwen Fan, and Zhangyang Wang. Signal processing for implicit neural representations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[36] Matthew D. Zeiler, Dilip Krishnan, Graham W. Taylor, and Rob Fergus. Deconvolutional networks. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. 3

[37] Qi Zhao, M. Salman Asif, and Zhan Ma. Dnerv: Modeling inherent dynamics via difference neural representation for videos. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3, 4, 5, 6