PROGRESSIVE FOURIER NEURAL REPRESENTATION FOR SEQUENTIAL VIDEO COMPILATION

Anonymous authors
Paper under double-blind review

A APPENDIX

A.1 DATASETS

1) UVG of 8 Video Sessions: For "Big Buck Bunny" frames collected from the scikit-video, we use the frames provided with the NeRV official code. "Big Buck Bunny" comprises 132 frames of 720×1080 resolution. The frames for the other seven videos, collected from the UVG dataset, are extracted from YUV Y4M videos, and further information can be found in the implementation details. As shown in Table 1, the seven videos have 1920×1080 resolution, with the shaking video comprising 300 frames and the other 6 videos containing 600 frames each. These videos are captured at 120 frames per second (FPS), and the duration of the shaking video is 2.5 seconds, while the duration of the other 6 videos is 5 seconds. For convenience, the video titles in the UVG of 8 Video Sessions are abbreviated, and their corresponding full titles in the UVG dataset are as follows: *1.bunny : Big Buck Bunny, 2.beauty : Beauty, 3.bosphorus : Bosphorus, 4.bee : HoneyBee, 5.jockey : Jockey, 6.setgo : ReadySetGo, 7.shake : ShakeNDry, 8.yacht : YachtRide*.

Table 1: the UVG8 Video Sessions.

	Video Sessions							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Categories	bunny	beauty	bosphorus	bee	jockey	setgo	shake	yacht
FPS	-	120	120	120	120	120	120	120
Length (sec)	-	5	5	5	5	5	2.5	5
Num. of Frames	132	600	600	600	600	600	300	600
Resolutions	720×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080

2) UVG of 17 Video Sessions: Compared to the UVG of 8 video sessions, the other nine videos are all collected from the UVG dataset. The frames for these videos are extracted from YUV RAW videos with a resolution of 1920x1080. Further information can be found in the implementation details. As shown in Table 2, the sunbath video consists of 300 frames at 50 FPS and 12 seconds, the lips video consists of 600 frames at 120 FPS and 5 seconds, and the other seven videos comprised of 600 frames at 50 FPS and 12 seconds. The full names of each video in the UVG dataset are as follows: *2.city : CityAlley, 4.focus : FlowerFocus, 6.kids : FlowerKids, 8.pan : FlowerPan, 10.lips : Lips, 12.race : RaceNight, 14.river : RiverBank, 16.sunbath : SunBath, 17.twilight : Twilight*.

Table 2: the UVG17 Video Sessions.

	Video Sessions								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Categories	bunny	city	beauty	focus	bosphorus	kids	bee	pan	jockey
FPS	-	50	120	50	120	50	120	50	120
Length (sec)	-	12	5	12	5	12	5	12	5
Num. of Frames	132	600	600	600	600	600	600	600	600
Resolutions	720×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080
	Video Sessions								
	10	**11**	**12**	**13**	**14**	**15**	**16**	**17**	
Categories	lips	setgo	race	shake	river	yacht	sunbath	twilight	
FPS	120	120	50	120	50	120	50	50	
Length (sec)	5	5	12	2.5	12	5	6	12	
Num. of Frames	600	600	600	300	600	600	300	600	
Resolutions	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	1920×1080	

A.2 IMPREMENTATION DETAILS

1) 7 Videos in UVG of 8 Video Sessions: To utilize the same video frame with NeRV, we downloaded 7 videos from the UVG dataset and employed the following commands to extract frames from the YUV Y4M videos.

» Download file : **[title] 3840x2160 8bit YUV Y4M**

» Command : **ffmpeg -i [file_name] [path]/f%05d.png**

2) 9 Videos in UVG of 17 Video Sessions: To expand our usage of videos, we acquired an additional 9 videos from the UVG dataset that are exclusively available as YUV RAW videos with a resolution of 3840x2160. We then extracted and resized the frames using the following command.

» Download file : **[title] 3840x2160 10bit YUV RAW**

» Command : **ffmpeg -s 3840x2160 -pix_fmt yuv420p10le -i [file_name] -vf scale=1920:1080 -pix_fmt rgb24 [path]/f%05d.png**

Digiturk provides the video contents of the UVG dataset. The dataset videos are available online at https://ultravideo.fi/#main.

Table 3: PNR's architecture of Fourier Subneural Operator (FSO), f-NeRV$*$. Note that PE denotes positional encoding.

layer	Module	Upscale Factor	Output Size ($C \times W \times H$)	Number of parameters
0	PE (w frame index)	-	$80 \times 1 \times 1$	-
	PE (w video index)	-	$80 \times 1 \times 1$	-
1	STEM of fc1	-	$512 \times 1 \times 1$	81,920
	STEM of fc2	-	$112 \times 16 \times 9$	8,257,536
2	NeRV2 block of conv	5×	$112 \times 80 \times 45$	2,825,200
	f-NeRV2	1×	$112 \times 80 \times 45$	1,605,632
3	NeRV3 block of conv	2×	$96 \times 160 \times 90$	387,456
	f-NeRV3	1×	$96 \times 160 \times 90$	37,847,040
4	NeRV4 block of conv	2×	$96 \times 320 \times 180$	332,160
5	NeRV5 block of conv	2×	$96 \times 640 \times 360$	332,160
6	NeRV6 block of conv	2×	$96 \times 1280 \times 720$	332,160
7	Multi-head layer for a video session	-	$3 \times 1280 \times 720$	291

EWC (Kirkpatrick et al., 2017). We trained EWC on the two novel benchmark datasets as a regularized baseline. When training with a new video, the EWC penalty was adopted as a regularization term to alleviate catastrophic forgetting. The importance of the parameter was calculated through the diagonal component of the Fisher Information matrix, and the EWC penalty increased as the difference in the vital parameter increased as follows:

$$L_E(v_t^s) = L(v_t^s) + \frac{\lambda}{2} \sum_i F_i(\theta_i - \theta_{p,i}^*)^2, \tag{1}$$

where L_E is the total loss for EWC learning, F represents the Fisher information matrix, λ is a hyperparameter to determine the importance of the previous video, i denotes each parameter, and θ_p^* denotes the parameter after training with the previous video. We randomly sampled 10 frames per video to compute the Fisher diagonal and stored them in a replay buffer. The hyperparameter λ was experimentally set to 2e6 to scale the EWC penalty.

iCaRL (Rebuffi et al., 2017). We trained iCaRL as a rehearsal-based baseline on the two novel benchmark datasets. To replay previous videos, we store a total of $m = 800$ frames in the replay buffer as an exemplar set, and as the training progresses, we save m/s frames per video. The exemplar management method is similar to that in iCaRL, where we compute the average feature map within

the video and select m/s video frames that approximate the average feature map. For knowledge distillation, we randomly sample frames from the exemplar set of previous videos at each learning step and performed training with the current video, as follows:

$$L_C(v_t^s) = L(v_t^s) + \lambda \frac{1}{t-1} \sum_{i}^{t-1} L(v_i^*), \tag{2}$$

where v_i^* is a frame sampled from example set of ith video, and λ is a hyperparameter experimentally set to 0.5.

ESMER (Sarfraz et al., 2023). We trained ESMER on the two novel benchmark datasets as a current strong rehearsal-based baseline. Error-Sensitive Reservoir Sampling (ESMER) maintains episodic memory, which leverages the error history to pre-select low-loss samples as candidates for the buffer of 800 samples. At this time, we didn't use any noisy labels when training ESMER. We observed that the ESMER could not reduce forgetting in representations. It seems better suited for retaining information in image classification tasks rather than neural implicit representations. Lastly, compared with iCaRL, ESMER replies buffer at each iteration, leading to ineffective training cost and performance as shown in Table 4.

A.3 Architecture

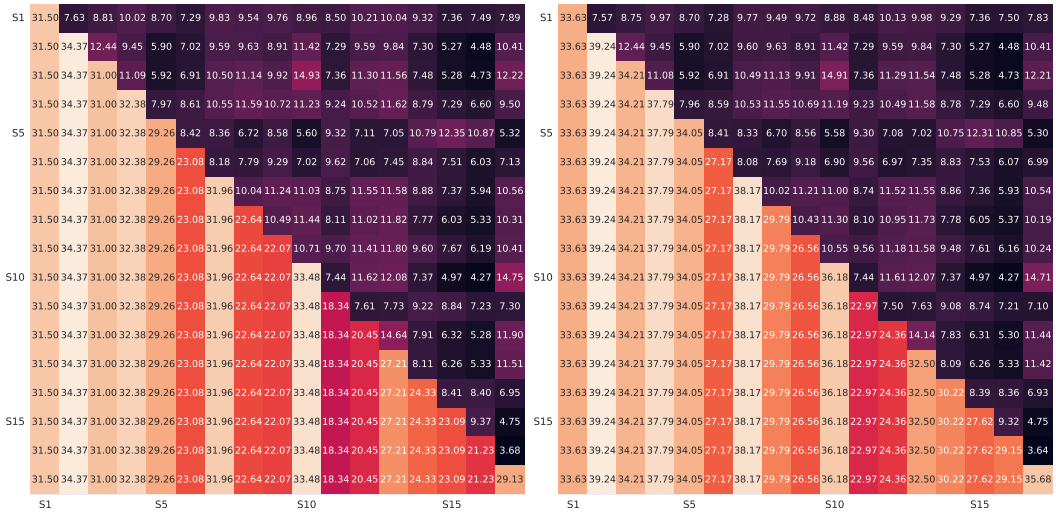
In this paper, we set NeRV as the baseline architecture, and we present a detailed overview of our architecture, as shown in Table 3. Our architecture has two differences compared to the previous NeRV. Firstly, to incorporate the outputs from the positional encoder of both the video index and frame index, we expand the input size of the first layer in the MLP from 80 to 160. Secondly, to enable Multi-Task Learning (MTL), we employ distinct head layers (multi-heads) for each video after the NeRV block. Apart from these modifications, our architecture remains consistent with the previous NeRV architecture. We stack five NeRV blocks with upscaling factors of 5, 2, 2, 2, and 2, respectively. The lowest channel width for output feature maps in the NeRV block is also set to 96. Comparing our architecture to the baseline NeRV, the number of parameters increases by 0.3433% to 12,567,560 for eight videos. Similarly, for seventeen videos, the number of parameters increases by 0.3642% to 12,570,179.

A.4 Additional Results

PFNR's Structure. We investigate the most expressive, effective, and efficient structure with Fourier Subneural Operator (FSO, **??**) for progressive neural implicit representations. To do so, we prepare the layer-wise FSO to maintain the output size of the baseline layer as shown in Table 3: the number of parameters of a spectral layer depends on input and output size, so as the layer increases, the parameters also increase. Table 4 shows the effectiveness of PFNR with spectral layers in terms of PSNR, BWT, and CAP. To acquire neural implicit representations, PFNR explores more diverse parameters than WSN regarding the same capacity.

Table 4: PSNR results with Fourier Subneural Operator (FSO) layer (f-NeRV$*$) (NeRV block, Table 3) on UVG8 Video Sessions with average PSNR and Backward Transfer (BWT), and Capacity (CAP). Note that $*$ denotes our reproduced results.

Method	Video Sessions								Avg. PSNR / BWT	CAP
	1	2	3	4	5	6	7	8		
STL, NeRV Chen et al. (2023)	39.63	36.06	37.35	41.23	38.14	31.86	37.22	32.45	36.74 / -	800.00 %
STL, NeRV*	39.66	36.28	38.14	42.03	36.58	29.22	37.27	31.45	36.33 / -	800.00 %
EWC Kirkpatrick et al. (2017)*	10.19	11.15	14.47	8.39	12.21	10.27	9.97	23.98	12.58 / -17.59	100.00 %
iCaRL Rebuffi et al. (2017)*	30.84	26.30	27.28	34.48	20.90	17.28	30.33	24.64	26.51 / -3.90	100.00 %
ESMER Sarfraz et al. (2023)*	31.71	23.09	24.15	28.03	17.30	13.81	12.45	24.57	21.92 / -9.99	100.00 %
WSN*, c = 10.0 %	27.81	30.66	29.30	33.06	22.16	18.40	27.81	22.97	26.52 / 0.0	28.00
WSN*, c = 30.0 %	31.37	32.19	29.92	33.62	22.82	18.96	28.43	23.40	27.59 / 0.0	59.00
WSN*, c = 50.0 %	34.05	32.28	29.98	32.88	22.15	18.61	27.68	23.64	27.66 / 0.0	77.00
WSN*, c = 70.0 %	35.62	32.08	29.46	31.37	21.60	18.13	27.33	22.61	27.28 / 0.0	91.00
PFNR, c = 10.0 %, f-NeRV2	28.49	32.30	30.30	35.12	24.10	19.82	29.89	24.76	28.10 / 0.0	33.83 %
PFNR, c = 30.0 %, f-NeRV2	31.99	33.56	31.82	36.61	25.28	20.97	31.07	25.73	29.63 / 0.0	76.76 %
PFNR, c = 50.0 %, f-NeRV2	34.46	33.91	32.17	36.43	25.26	20.74	30.18	25.45	29.82 / 0.0	102.64 %
PFNR, c = 70.0 %, f-NeRV2	36.04	33.46	31.05	32.57	23.40	19.41	28.31	24.31	28.57 / 0.0	111.66 %
MLT (upper-bound)	34.22	32.79	32.34	38.33	25.30	22.44	33.73	27.05	30.78 / -	100.00 %



(a) WSN, $c = 30.0\%$ (b) PFNR, $c = 30.0\%$, f-NeRV3

Figure 1: Transfer Matrixes of WSN v.s. PFNR on the UVG17 dataset measured by PSNR of source and target.

Transfer Matrix. We prepare the transfer matrix to prove our PFNR's forget-freeness and to show video correlation among other videos, as shown in Figure 1 on the UVG17 dataset; lower triangular estimated by each session subnetwork denotes that our PFNR is a forget-free method and upper triangular calculated by current session subnetwork denotes the video similarity between source and target. The PFNR proves the effectiveness from the lower triangular of Figure 1 (a) and (b). Nothing special is observable from the upper triangular since they are not correlated.

Video Generation.

We prepared some results of video generation as shown in Figure 2. We showed the quantized and encoded PFNR's video generation results in Figure 3. We demonstrated that a compressed sparse solution in FP8 (PFNR with $c = 30.0\%$, f-NeRV2) generates video representations sequentially without a significant quality drop. The results of the FP4 showed that the compressed model cannot create image pixels in detail.

Table 5: PSNR results with Fourier Subneural Operator (FSO) layer (f-NeRV$*$) (detailed in Table 3) on UVG17 Video Sessions with average PSNR and Backward Transfer (BWT) of PSNR. Note that $*$ denotes our reproduced results.

Method	Video Sessions																	Avg. MS-SSIM BWT
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
STL, NeRV Chen et al. (2023)	39.63	-	36.06	-	37.35	-	41.23	-	38.14	-	31.86	-	37.22	-	32.45	-	-	- / -
STL, NeRV*	39.66	44.89	36.28	41.13	38.14	31.53	42.03	34.74	36.58	36.85	29.22	31.81	37.27	34.18	31.45	38.41	43.86	36.94 / -
EWC Kirkpatrick et al. (2017)*	11.15	9.21	12.71	11.40	15.58	9.25	7.06	12.96	6.34	10.31	9.55	13.39	5.76	8.67	10.93	10.92	28.29	11.38 / -16.13
iCaRL Rebuffi et al. (2017)*	24.31	28.25	22.19	22.74	22.84	16.55	29.37	17.92	16.65	27.43	13.64	16.42	24.02	21.60	19.40	18.60	26.46	21.67 / -6.23
ESMER Sarfraz et al. (2023)*	30.77	26.33	22.79	21.35	23.76	13.64	28.25	15.22	16.71	23.78	13.35	15.23	18.21	19.22	24.59	20.61	22.42	20.95 / -15.23
WSN*, c = 10.0 %	27.68	31.31	30.29	31.63	28.66	22.57	31.62	22.04	21.05	32.71	17.85	20.09	27.07	23.84	22.98	20.50	28.56	25.91 / 0.0
WSN*, c = 30.0 %	31.50	34.37	31.00	32.38	29.26	23.08	31.96	22.64	22.07	33.48	18.34	20.45	27.21	24.33	23.09	21.23	29.13	26.80 / 0.0
WSN*, c = 50.0 %	34.02	34.93	31.04	31.74	28.95	23.07	31.26	22.32	21.93	33.35	18.22	20.34	26.88	24.22	22.72	21.30	28.86	26.77 / 0.0
WSN*, c = 70.0 %	35.64	34.36	30.26	30.27	27.99	22.55	29.88	21.46	20.79	32.37	17.63	20.00	26.68	23.79	22.34	20.69	28.68	26.20 / 0.0
PFNR, c = 10.0 %, f-NeRV2	28.31	33.57	31.92	33.67	29.98	23.99	34.39	24.8	23.94	35.08	19.70	22.03	29.56	26.57	24.79	24.10	31.35	28.10 / 0.0
PFNR, c = 30.0 %, f-NeRV2	32.01	35.84	32.97	35.17	31.24	24.82	36.01	25.85	24.83	35.76	20.50	22.79	30.40	27.37	25.52	25.40	32.70	29.36 / 0.0
PFNR, c = 50.0 %, f-NeRV2	34.49	37.13	33.21	35.50	30.87	24.72	34.36	24.79	24.73	35.65	20.33	22.65	29.78	27.05	25.18	25.18	32.39	29.29 / 0.0
PFNR, c = 70.0 %, f-NeRV2	36.02	36.50	32.09	32.15	28.67	23.35	30.63	22.86	23.18	34.90	19.08	21.30	27.87	25.86	24.12	23.47	30.34	27.79 / 0.0
PFNR, c = 10.0 %, f-NeRV3	30.40	36.94	32.92	35.64	31.90	25.25	36.86	28.36	24.46	35.68	20.84	22.86	31.61	28.8	26.07	26.31	33.68	29.92 / 0.0
PFNR, c = 30.0 %, f-NeRV3	33.64	39.24	34.21	37.79	34.05	27.17	38.17	29.79	26.56	36.18	22.97	24.36	32.50	30.22	27.62	29.15	35.68	31.72 / 0.0
PFNR, c = 50.0 %, f-NeRV3	36.59	39.95	34.46	38.01	34.21	27.26	37.47	29.07	26.72	36.17	22.65	24.46	32.43	30.07	27.62	29.30	35.26	31.65 / 0.0
PFNR, c = 70.0 %, f-NeRV3	38.24	39.47	33.72	35.18	31.58	25.65	33.64	26.23	23.92	35.75	20.81	22.73	30.54	28.55	26.17	26.84	33.15	30.13 / 0.00
MLT (upper-bound)	32.39	34.35	31.45	34.03	30.70	24.53	37.13	27.83	23.80	34.69	20.77	22.37	32.71	28.00	25.89	26.40	33.16	29.42 / -

A.5 Layer-wise Representations

To investigate the property of FSO, we prepare the layer-wise representations, as shown in Figure 4. The representations of PFNR (f-NeRV3) focus on textures rather than objects at the NeRV3 layer in

Table 6: MS-SSIM results with Fourier Subnueral Operator (FSO) layer (f-NeRV$*$) (detailed in Table 3) on UVG17 Video Sessions with average MS-SSIM and Backward Transfer (BWT) of MS-SSIM. Note that $*$ denotes our reproduced results.

Method	Video Sessions																	Avg. MS-SSIM BWT
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
STL, NeRV*	0.99	0.99	0.95	0.98	0.98	0.96	0.99	0.98	0.97	0.95	0.96	0.96	0.98	0.98	0.96	0.99	0.99	0.97 / -
EWC Kirkpatrick et al. (2017)*	0.26	0.24	0.44	0.24	0.40	0.29	0.15	0.17	0.26	0.26	0.17	0.34	0.04	0.30	0.33	0.31	0.91	0.30 / -0.55
iCaRL Rebuffi et al. (2017)*	0.74	0.88	0.67	0.67	0.64	0.48	0.91	0.53	0.37	0.82	0.35	0.53	0.75	0.70	0.61	0.60	0.87	0.65 / -0.20
ESMER Sarfraz et al. (2023)*	0.85	0.86	0.64	0.63	0.66	0.46	0.89	0.51	0.42	0.79	0.30	0.51	0.43	0.68	0.82	0.6	0.63	0.62 / -0.37
WSN*, c = 10.0 %	0.90	0.94	0.88	0.92	0.87	0.75	0.96	0.74	0.69	0.91	0.57	0.72	0.86	0.80	0.74	0.69	0.92	0.82 / 0.0
WSN*, c = 30.0 %	0.96	0.97	0.89	0.93	0.88	0.77	0.97	0.77	0.73	0.91	0.60	0.74	0.86	0.81	0.76	0.72	0.93	0.84 / 0.0
WSN*, c = 50.0 %	0.98	0.97	0.89	0.92	0.88	0.77	0.96	0.75	0.73	0.91	0.60	0.74	0.85	0.80	0.75	0.73	0.92	0.83 / 0.0
WSN*, c = 70.0 %	0.98	0.97	0.88	0.91	0.85	0.75	0.95	0.70	0.68	0.91	0.55	0.72	0.85	0.80	0.73	0.70	0.92	0.82 / 0.0
PFNR, c = 10.0 %, f-NeRV2	0.92	0.97	0.90	0.94	0.90	0.81	0.98	0.86	0.79	0.93	0.69	0.79	0.91	0.87	0.82	0.84	0.96	0.88 / 0.0
PFNR, c = 30.0 %, f-NeRV2	0.97	0.98	0.92	0.95	0.92	0.84	0.98	0.88	0.82	0.93	0.73	0.82	0.92	0.89	0.84	0.87	0.97	0.90 / 0.0
PFNR, c = 50.0 %, f-NeRV2	0.98	0.99	0.92	0.95	0.92	0.83	0.98	0.86	0.81	0.93	0.72	0.82	0.91	0.88	0.84	0.87	0.97	0.89 / 0.0
PFNR, c = 70.0 %, f-NeRV2	0.99	0.98	0.91	0.93	0.87	0.78	0.96	0.77	0.77	0.93	0.66	0.77	0.89	0.85	0.80	0.82	0.95	0.86 / 0.0
PFNR, c = 10.0 %, f-NeRV3	0.96	0.99	0.92	0.96	0.94	0.86	0.99	0.94	0.82	0.93	0.76	0.83	0.93	0.92	0.87	0.90	0.98	0.91 / 0.0
PFNR, c = 30.0 %, f-NeRV3	0.98	0.99	0.93	0.97	0.96	0.91	0.99	0.96	0.87	0.94	0.84	0.87	0.94	0.94	0.90	0.94	0.98	0.94 / 0.0
PFNR, c = 50.0 %, f-NeRV3	0.99	0.99	0.93	0.97	0.96	0.91	0.99	0.95	0.87	0.94	0.83	0.88	0.94	0.94	0.9	0.95	0.98	0.94 / 0.0
PFNR, c = 70.0 %, f-NeRV3	0.99	0.99	0.93	0.96	0.93	0.87	0.98	0.90	0.81	0.93	0.76	0.83	0.93	0.92	0.87	0.91	0.97	0.91 / 0.0
MLT (upper-bound)	0.97	0.97	0.90	0.94	0.91	0.82	0.99	0.92	0.80	0.92	0.75	0.81	0.94	0.90	0.85	0.89	0.97	0.90 / -

the video session of bosphorus. In the video session of bee, PFNR (f-NeRV3) also tends to capture local textures broadly at the NeRV3 layer. This behavior of f-NeRV3 leads to better generalization at final prediction than others, such as WSN and PFNR (f-NeRV2). Moreover, we conducted an ablation study to inspect the best sparsity of f-NeRV3, as shown in Figure 5. Specifically, the performances of PFNR, c=50.0% depend on the sparsity of f-NeRV3. We observed that f-NeRV3 with c=50.0% was the best sparsity.

A.6 Current Limitations and Future Work

Since the parameters of FSO depend on the input/output feature map size, in this task, the deeper the FSO layer, the larger the parameters increase. Nevertheless, we found the most parameter-efficient FSO structure through layer-wise inspection and layer sparsity to describe the best neural implicit representations. In future work, we will design a more parameter-efficient FSO layer in continual tasks such as neural implicit representation and task/class incremental learnings.

A.7 Broader Impacts

As the most popular media format nowadays, videos are generally viewed as frames of sequences. Unlike that, our proposed PFNR is a novel way to represent sequential videos as a function of video session and time, parameterized by the neural network firstly in Fourier space, which is more efficient and might be used in many video-related tasks, such as sequential video compression, sequential video denoising, complex sequential physical modeling (Li et al., 2020a;b; Kovachki et al., 2021; Tran et al., 2021), and so on. Hopefully, this can save bandwidth and fasten media streaming, enriching entertainment potential. Unfortunately, like many advances in deep learning for videos, this approach could be used for various purposes beyond our control.

A.8 Public Codes of our PFNR

We provide the core parts of PFNR to understand better. Please refer to the attached file. We will provide all training and inference codes soon.

References

Hao Chen, Matt Gwilliam, Ser-Nam Lim, and Abhinav Shrivastava. Hnerv: A hybrid neural representation for videos. *arXiv preprint arXiv:2304.02633*, 2023. 3, 4

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis,

Table 7: PFNR (f-NeRV2) of Quantization + Compression on the UVG17 Video Sessions with average PSNR/MS-SSIM and Backward Transfer (BTW). Note that bit is the bit length used to represent parameter value.

Method	Video Sessions																	Avg. * / BWT
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
C = 10.0%,		PSNR,	MS-SSIM															
bit=4	17.73	30.45	30.74	30.56	26.15	23.29	23.16	19.75	23.80	26.19	19.53	20.36	24.99	26.28	24.61	23.86	29.95	24.79 / -0.80
bit=8	28.21	33.56	31.92	33.66	29.97	23.98	34.35	24.78	23.95	35.07	19.70	22.02	29.57	26.58	24.79	24.09	31.34	28.09 / -0.01
bit=16	28.31	33.57	31.92	33.67	29.98	23.99	34.39	24.80	23.94	35.08	19.70	22.03	29.56	26.57	24.79	24.10	31.35	28.10 / 0.00
bit=32	28.31	33.57	31.92	33.67	29.98	23.99	34.39	24.80	23.94	35.08	19.70	22.03	29.56	26.57	24.79	24.10	31.35	28.10 / 0.00
bit=4	0.60	0.94	0.90	0.93	0.87	0.80	0.89	0.73	0.79	0.89	0.69	0.76	0.87	0.86	0.82	0.83	0.95	0.83 / -0.02
bit=8	0.92	0.97	0.90	0.94	0.90	0.81	0.98	0.86	0.79	0.93	0.69	0.79	0.91	0.87	0.82	0.84	0.96	0.88 / 0.00
bit=16	0.92	0.97	0.90	0.94	0.90	0.81	0.98	0.86	0.79	0.93	0.69	0.79	0.91	0.87	0.82	0.84	0.96	0.88 / 0.00
bit=32	0.92	0.97	0.90	0.94	0.90	0.81	0.98	0.86	0.79	0.93	0.69	0.79	0.91	0.87	0.82	0.84	0.96	0.88 / 0.00
C = 30.0%,		PSNR,	MS-SSIM															
bit=4	10.96	27.43	24.65	25.04	24.05	22.92	16.24	23.08	23.91	33.51	20.10	22.47	28.79	26.58	24.94	24.87	31.61	24.19 / -2.13
bit=8	31.72	35.81	32.95	35.11	31.22	24.81	35.82	25.84	24.84	35.76	20.49	22.79	30.40	27.37	25.52	25.40	32.69	29.33 / -0.02
bit=16	32.01	35.84	32.97	35.17	31.24	24.82	36.01	25.85	24.83	35.76	20.50	22.79	30.40	27.37	25.52	25.40	32.70	29.36 / 0.00
bit=32	32.01	35.84	32.97	35.17	31.24	24.82	36.01	25.85	24.83	35.76	20.50	22.79	30.40	27.37	25.52	25.40	32.70	29.36 / 0.00
bit=4	0.47	0.92	0.85	0.91	0.84	0.81	0.74	0.82	0.80	0.93	0.72	0.81	0.91	0.88	0.83	0.86	0.96	0.83 / -0.04
bit=8	0.97	0.98	0.92	0.95	0.92	0.84	0.98	0.88	0.82	0.93	0.73	0.82	0.92	0.89	0.84	0.87	0.97	0.90 / 0.00
bit=16	0.97	0.98	0.92	0.95	0.92	0.84	0.98	0.88	0.82	0.93	0.73	0.82	0.92	0.89	0.84	0.87	0.97	0.90 / 0.00
bit=32	0.97	0.98	0.92	0.95	0.92	0.84	0.98	0.88	0.82	0.93	0.73	0.82	0.92	0.89	0.84	0.87	0.97	0.90 / 0.00
C = 50.0%,		PSNR,	MS-SSIM															
bit=4	7.19	23.92	20.83	24.71	24.77	21.91	28.55	23.03	23.33	32.96	19.22	21.80	25.86	22.00	23.85	22.82	29.59	22.92 / -3.91
bit=8	34.03	37.08	33.20	35.47	30.86	24.71	34.34	24.78	24.73	35.64	20.33	22.65	29.78	27.04	25.17	25.17	32.39	29.26 / -0.03
bit=16	34.49	37.13	33.21	35.50	30.87	24.72	34.36	24.79	24.73	35.65	20.33	22.65	29.78	27.05	25.18	25.18	32.39	29.29 / 0.00
bit=32	34.49	37.13	33.21	35.50	30.87	24.72	34.36	24.79	24.73	35.65	20.33	22.65	29.78	27.05	25.18	25.18	32.39	29.29 / 0.00
bit=4	0.25	0.87	0.79	0.87	0.85	0.78	0.95	0.81	0.79	0.92	0.68	0.78	0.88	0.80	0.81	0.82	0.94	0.80 / -0.07
bit=8	0.98	0.99	0.92	0.95	0.92	0.83	0.98	0.86	0.81	0.93	0.72	0.82	0.91	0.88	0.84	0.87	0.97	0.89 / 0.00
bit=16	0.98	0.99	0.92	0.95	0.92	0.83	0.98	0.86	0.81	0.93	0.72	0.82	0.91	0.88	0.84	0.87	0.97	0.89 / 0.00
bit=32	0.98	0.99	0.92	0.95	0.92	0.83	0.98	0.86	0.81	0.93	0.72	0.82	0.91	0.88	0.84	0.87	0.97	0.89 / 0.00
C = 70.0%,		PSNR,	MS-SSIM															
bit=4	7.05	15.54	16.27	18.14	25.63	21.95	24.75	20.50	20.97	17.84	18.34	19.18	23.49	14.20	22.55	22.04	17.19	19.15/ -3.48
bit=8	35.52	36.43	32.08	32.14	28.66	23.35	30.61	22.86	23.18	34.89	19.08	21.29	21.86	25.86	24.13	23.47	30.30	27.75 / -0.03
bit=16	36.02	36.50	32.09	32.15	28.67	23.35	30.63	22.86	23.18	34.90	19.08	21.30	27.87	25.86	24.12	23.47	30.34	27.79 / 0.00
bit=32	36.02	36.50	32.09	32.15	28.67	23.35	30.63	22.86	23.18	34.90	19.08	21.30	27.87	25.86	24.12	23.47	30.34	27.79 / 0.00
bit=4	0.28	0.56	0.68	0.76	0.82	0.74	0.88	0.69	0.71	0.75	0.61	0.71	0.79	0.55	0.75	0.78	0.88	0.70 / -0.08
bit=8	0.98	0.98	0.91	0.93	0.87	0.78	0.96	0.77	0.77	0.93	0.65	0.77	0.89	0.85	0.80	0.82	0.95	0.86 / 0.00
bit=16	0.99	0.98	0.91	0.93	0.87	0.78	0.96	0.77	0.77	0.93	0.66	0.77	0.89	0.85	0.80	0.82	0.95	0.86 / 0.00
bit=32	0.99	0.98	0.91	0.93	0.87	0.78	0.96	0.77	0.77	0.93	0.66	0.77	0.89	0.85	0.80	0.82	0.95	0.86 / 0.00

Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. 2017. 2, 3, 4, 5

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces. *arXiv preprint arXiv:2108.08481*, 2021. 5

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a. 5

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020b. 5

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017. 2, 3, 4, 5

Fahad Sarfraz, Elahe Arani, and Bahram Zonooz. Error sensitivity modulation based experience replay: Mitigating abrupt representation drift in continual learning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zlbci7019Z3. 3, 4, 5

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. *arXiv preprint arXiv:2111.13802*, 2021. 5

Figure 2: PFNR's Video Generation (from t=0 to t=3) with $c = 30.0\%$ on the UVG17 dataset.

Figure 3: PFNR's Qunatizationed and Compresssed Video Generation (t=0) with $c = 30.0\%$, f-NeRV2 on the UVG17 dataset. Note that GT: ground-truth and PRED: PFNR's prediction in FP32,16,8, and 4.

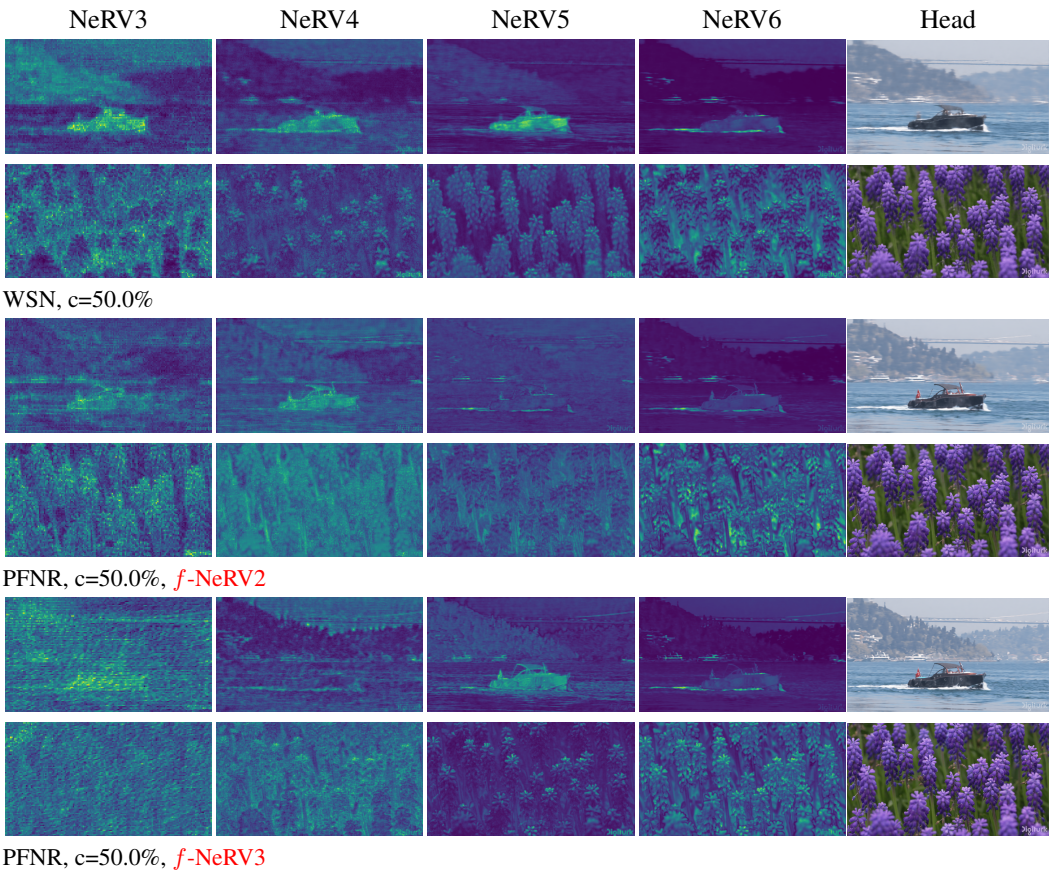


Figure 4: PFNR's Representations of NeRV Blocks with $c = 50.0\%$ on the UVG17 dataset.

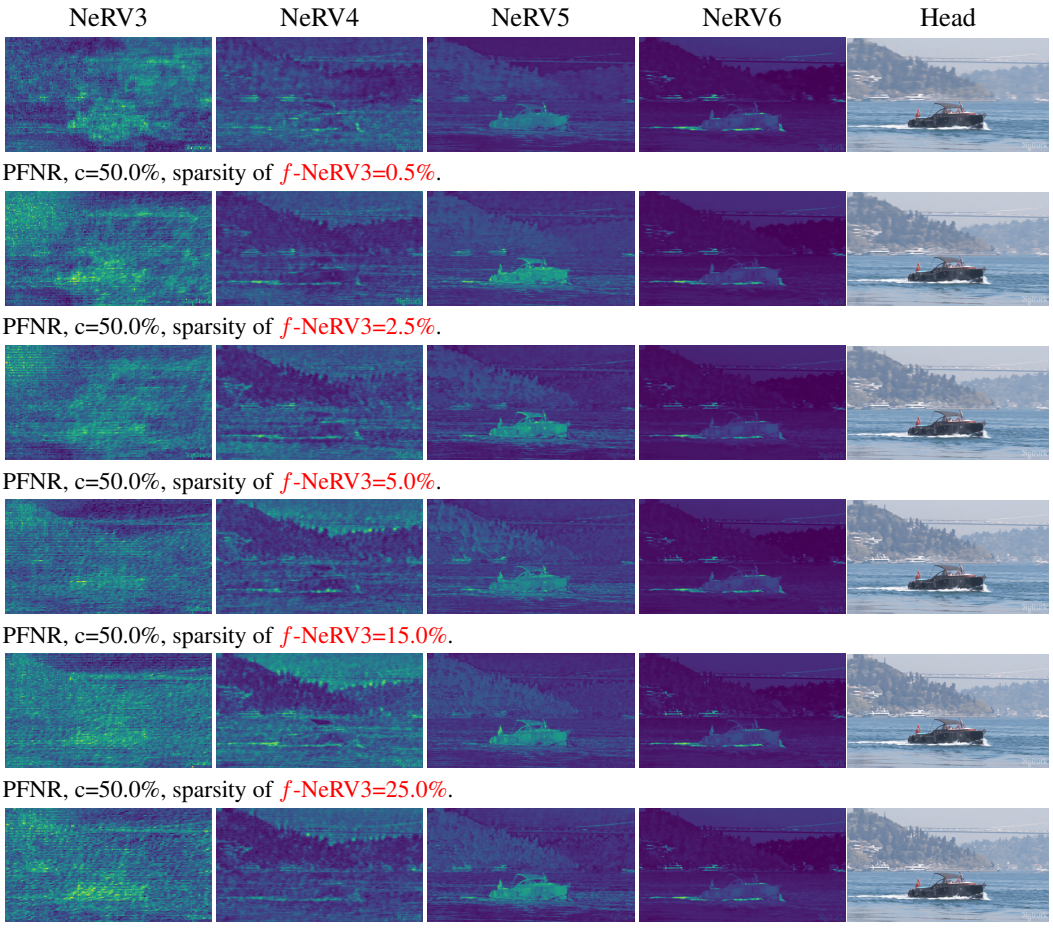


Figure 5: Various sparsity of f-NeRV3 ranging from 0.05 % (top row) to 50.0 % (bottom row) on the UVG17 dataset.