9-2022

Motion-Adjustable Neural Implicit Video Representation

Long Mai
Portland State University, mtlong@cs.pdx.edu

Feng Liu
Portland State University, fliu@cs.pdx.edu

Citation Details

Mai, L., & Liu, F. (2022). Motion-Adjustable Neural Implicit Video Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10738-10747).

Motion-Adjustable Neural Implicit Video Representation

Long Mai
Portland State University
mai.t.long88@gmail.com

Feng Liu
Portland State University
fliu@cs.pdx.edu

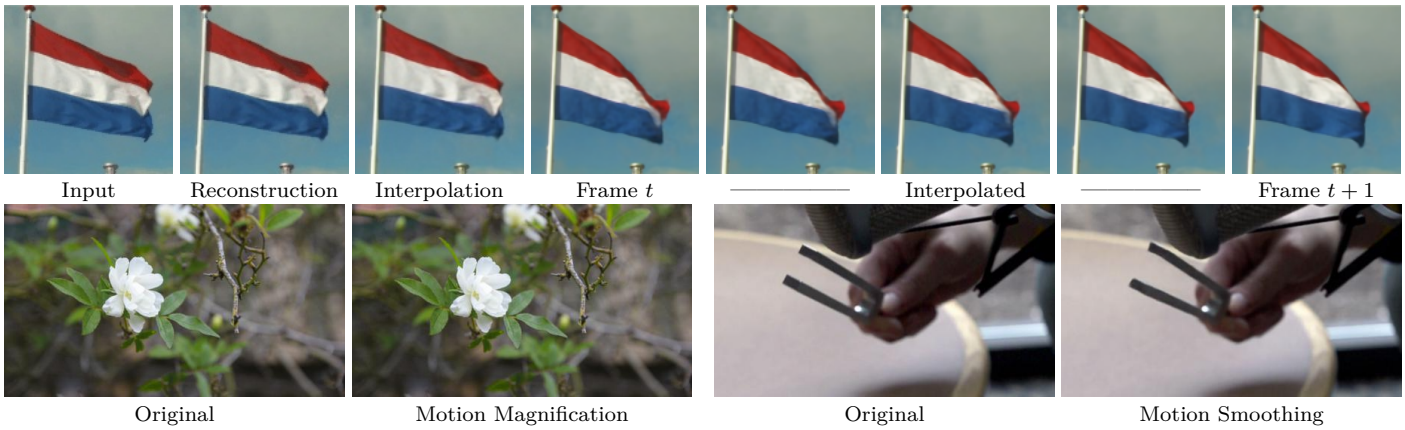


Figure 1. We extend the standard image-based implicit neural representation to a motion-adjustable neural implicit video representation by incorporating temporally varying phase-shift information into Fourier-based positional encoding. By changing the phase-shift values at inference time, our method can not only reconstruct video data but can also re-synthesize videos with modified motion properties. This paper contains **video figures** which can be viewed on our project website https://mai-t-long.com/Phase_NIVR/index.html. We also strongly encourage our readers to download and view the **video-enabled version** of this paper (https://mai-t-long.com/Phase_NIVR/paper.pdf) using Adobe Reader.

Abstract

Implicit neural representation (INR) has been successful in representing static images. Contemporary image-based INR, with the use of Fourier-based positional encoding, can be viewed as a mapping from sinusoidal patterns with different frequencies to image content. Inspired by that view, we hypothesize that it is possible to generate temporally varying content with a single image-based INR model by displacing its input sinusoidal patterns over time. By exploiting the relation between the phase information in sinusoidal functions and their displacements, we incorporate into the conventional image-based INR model a phase-varying positional encoding module, and couple it with a phase-shift generation module that determines the phase-shift values at each frame. The model is trained end-to-end on a video to jointly determine the phase-shift values at each time with the mapping from the phase-shifted sinusoidal functions to the corresponding frame, enabling an implicit video representation. Experiments on a wide range of videos suggest that such a model is capable of learning to interpret phase-varying positional embeddings into the corresponding time-varying content. More importantly, we found that the learned phase-shift vectors tend to capture meaningful temporal and motion information from the video. In particular, manipulating the phase-shift vectors induces meaningful changes in the temporal dynamics of the resulting video, enabling non-trivial temporal and motion editing effects such as temporal interpolation, motion magnification, motion smoothing, and video loop detection.

1. Introduction

Implicit neural representation (INR) has recently emerged as a powerful paradigm for representing visual data [19, 20, 29, 30, 34]. Notably, INR has recently been successfully adopted to represent 2D images for image processing and synthesis [1, 5, 6]. Image-based INR employs coordinate-based multi-layer perceptron (MLP), typically along with Fourier-based positional encoding, to map 2D pixel coordinates to the corresponding color values. Existing works also studied video-based INR and considered it as a natural extension of their image-based counterpart [15, 29]. Such an approach uses time as an additional input coordinate to the coordinate-MLP model, effectively treating a video as a 3D volume without explicitly modeling inherent temporal connec-

tion among video frames.

Alternatively, a video is often considered as a sequence of images evolving over time in computer vision research [23,32]. This work explores a video-based INR from that perspective. We investigate if it is possible to leverage an image-based INR to generate temporally varying video content motivated by two observations. First, image-based INR, with the use of Fourier-based positional encoding [34], operates as a mapping from sinusoidal patterns of different frequencies to 2D image content. Varying the input sinusoids would necessarily cause the generated output to vary accordingly. Therefore, in principle a time-evolving image sequence can be generated from a single image-based INR by varying its sinusoidal functions over time. Second, displacements of sinusoidal functions can be modeled mathematically by the shifts in their phase angles. Time-varying sinusoids can therefore be achieved by assigning different phase shifts at different times.

We develop an implicit neural representation for videos based on these observations. We model the pixel generation process in a frame-wise manner with an image-based INR, and leverage the phase information in its positional encoding to generate temporally varying video content. Our model consists of two components, a frame generation module and a phase-shift generation module. Our frame generation module maps each pixel coordinate $\boldsymbol{c} = (x, y)$ to the color value $M_f(\boldsymbol{c})$ at the corresponding coordinates in the image plane. This frame generation module is a standard image-based INR model with a minimal yet important modification to its positional encoding (PE) operation. Different from a standard INR, each sinusoidal function in our PE is not static but to be shifted at each time t by a phase-shift vector $\boldsymbol{\phi}(t)$. The mapping $\boldsymbol{\phi}$ is generated by the phase-shift generation module M_p, jointly trained end-to-end with M_f to fit the input video. After training, M_p can provide the per-frame phase-shift vector at each corresponding frame in the video. Those learned phase-shift vectors can be externally manipulated before entering the frame generation stage, potentially enabling new generated content with modified dynamics. That makes our neural implicit video representation motion-adjustable.

With the proposed neural implicit video representation, we center our study around two questions. *First, can the model learn to fit a video?* Compared to a standard INR approach where the spatial coordinate encodings are fixed across frames, the input coordinate encodings to our frame generation model constantly change from frame to frame, making it more challenging to memorize pixel value at each location. *Second, does the learned phase space have any meaningful structures?* As the image content at each time is associated with a phase-shift vector, it is interesting to see whether manipulating the learned per-frame phase-shift sequence can result in meaningful changes in the generated video. Our experiments on diverse video content suggest positive answers. We found that the model can learn to interpret the learned phase-varying positional encoding into the corresponding time-varying video content. Interestingly, we found that the resulting phase space corresponds to meaningful information in the video. Manipulating the generated phase-shift vectors can enable different temporal-dynamics effects such as temporal interpolation, motion magnitude adjustment, and motion filtering from the video as shown in Figure 1.

This paper makes the following contributions.

- We introduce a motion-adjustable neural implicit video representation. Instead of treating the time dimension equally as the spatial dimensions, our representation maps time to a driving signal to modulate the frame-generation process, effectively adapting regular image-based INR to generate temporally varying video content.

- We report the interesting finding that the phase information in Fourier-based positional encoding can be flexibly leveraged to capture temporal dynamics in a video. Our work adds to the growing literature on the use of Fourier-based positional encoding in INR, complementing prior works that study the roles of frequency information in Fourier-based positional embeddings.

- We experiment on a variety of real-world videos and demonstrate that our neural implicit video representation can not only represent a video but can also allow for modifying certain temporal-dynamics aspects of the video content, enabling a motion-adjustable neural implicit video representation potentially useful for video processing applications.

2. Related Work

Implicit Neural Representation has been shown a powerful approach to represent visual data, such as 3D data modelling [2–4, 13, 16, 19, 20, 25, 26, 30, 36, 41, 42] and image representation [14, 15, 29, 33, 34]. Image-based INR frameworks have been developed for numerous applications, including image compression [6], super-resolution [5], and image synthesis [1,31]. In this paper, we focus on exploring a motion-adjustable neural implicit video representation. Different from the standard approach which extends image-based INR to

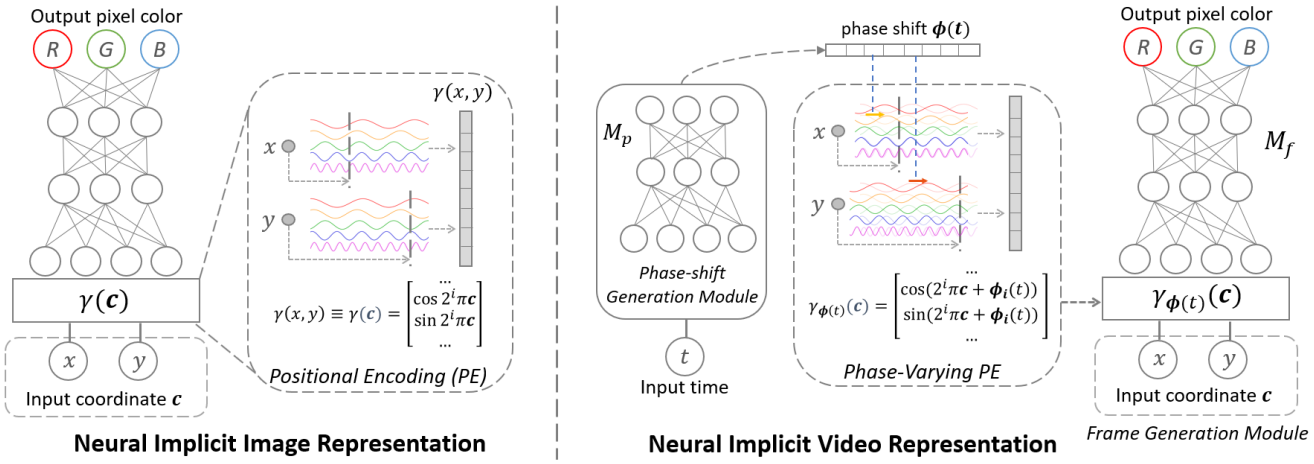


Figure 2. Motion-Adjustable Neural Implicit Video Representation. We extend image-based implicit neural representation (left) to model a video. Our method determines the phase-shift $\phi(t)$ at each time t using the phase-shift generation network M_p. The frame generation network M_f synthesizes the video frames corresponding to the positional embeddings with the phase shifted by $\phi(t)$. At inference time, $\phi(t)$ can be manipulated to generate new videos with modified dynamics.

fit a 3D video volume [15, 29], we leverage the phase information in the Fourier-based positional encoding to learn temporally varying video content with a regular 2D image-based INR.

Implicit Neural Representation for 3D Dynamic Scenes. Following the immense success of Neural Radiance Fields (NeRF) [19], many methods extend NeRF to model temporally varying 3D scenes from video data [8, 9, 12, 21, 24, 35, 40]. Existing works along this line typically treat video frames as the projection of a dynamic 3D scene onto the image plane. These methods explicitly model 3D scenes and per-frame camera poses. This paper works on a more relaxed setting without any 3D scene or camera information and focuses on adapting image-based INR model to capture the temporally evolving content in a video.

Fourier-Feature Based Positional Encoding. Positional encoding (PE) refers to the mechanism to represent position information by mapping low-dimensional input coordinates to higher-dimensional vectors, typically through a collection of sinusoidal functions. Initially made popular by Vaswani *et al.* through their Transformer paper [37], positional encoding has also proved critical for implicit neural representation models [19]. Recent works have studied the importance of the frequency components in PE to the model's fitting quality [29, 34]. Our work adds to the growing literature on Fourier-feature-based positional encoding in INR, demonstrating that besides the frequency information, the phase information in Fourier-based PE can also be used to enable video modelling.

Phase-Based Motion Modelling. Our work is inspired in part from the rich literature on phase-based motion processing [7, 10, 17, 18, 38]. These works built on the connection between motion information in a video and its phase information extracted through frequency domain analysis [39] to enable various motion editing applications such as motion estimation [7, 10], motion magnification [38], and frame interpolation [17, 18]. In this paper we explore the possibility of leveraging phase information embedded in the Fourier-based positional encoding to help implicit neural representation models learn temporal dynamics information in video data.

3. Neural Implicit Video Representation

We first review image-based INR and motivate the use of phase shifts for generating temporally varying content. Image-based INR represents an image as a continuous function $f : \boldsymbol{c} \rightarrow \boldsymbol{v}$, where $\boldsymbol{c} = (x, y)$ are 2D coordinates on the normalized image plane, and $\boldsymbol{v} = (R, G, B)$ is the corresponding color value. The mapping function f is parameterized by the weights of a multi-layer perceptron (MLP) M_f. In practice, the input coordinates $\boldsymbol{c}$ are first mapped to higher-dimension vectors $\gamma(\boldsymbol{c})$ through a positional encoder module γ. M_f then maps the resulting positional encodings to the final color value $\boldsymbol{v}$ (Figure 2 left).

We adopt the widely used Fourier-based positional encoding scheme [13, 19, 37] that forms the encoding by concatenating sinusoidal functions of c

$$\gamma(\boldsymbol{c}) = [\gamma_0(\boldsymbol{c}), ...\gamma_{N-1}(\boldsymbol{c})] \quad (1)$$

$$\gamma_i(\boldsymbol{c}) = [\sin(2^{i-i_0}\pi \boldsymbol{c}), \cos(2^{i-i_0}\pi \boldsymbol{c})] \quad (2)$$

where N denotes the number of frequencies. $\gamma_i(\boldsymbol{c})$ represents the encoding corresponding to the i-th frequency. The sin and cos functions are defined coordinate-wise. i_0 controls the lowest frequency component to use, which is typically set to 0 in most INR models. With the positional encoding incorporated, the resulting model can be viewed as mapping the sinusoidal patterns arranged in 2D planes to the corre-

sponding image content. From that view, we hypothesize that an image-based INR model can be made to generate temporally varying content by displacing its input sinusoidal patterns over time.

3.1. Neural Implicit Video Representation

The displacement of the sinusoidal functions can be achieved by shifting their phase angles. Accordingly, we jointly determine the phase-shift values at each time and the image-based INR model that maps the phase-shifted positional encodings to the corresponding video frames as shown in Figure 2 right. This leads to our Neural Implicit Video Representation. Below we detail its two main components.

Frame Generation. The frame generation module M_f generates each 2D video frame. As in conventional image-based INR, M_f maps each 2D coordinate $\boldsymbol{c}$ to the corresponding pixel value using a coordinate-MLP with Fourier-based positional encoding. To make it generate different video content at different time, we modify its positional encoding module to enable phase-varying positional encoding. Specifically, we incorporate an explicit phase-shift term into each sinusoidal function. As a result, the per-frequency positional embedding in Equation 2 is modified to

$$\gamma_{\phi(t)_i}(\boldsymbol{c}) = [\sin(2^{i-i_0}\pi\boldsymbol{c} + \phi_i(t), \cos(2^{i-i_0}\pi\boldsymbol{c} + \phi_i(t)] \quad (3)$$

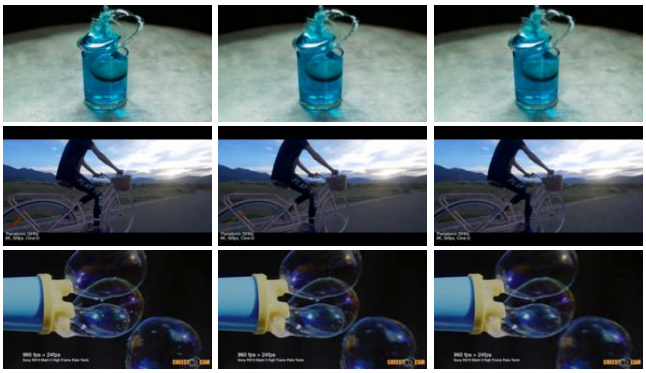
where $\phi_i(t)$ is a two-dimensional vector representing the i-th component of the phase shift at time t. With this minimal change, M_f can generate different values for the same (x, y) coordinate at different time, adapting an image-generation model for video generation.

Phase-Shift Generation. We parameterize the mapping from time t to phase shift $\phi(t)$ with a neural network M_p. As the mapping has continuous nature, we implement M_p as a 1-D implicit neural representation. Specifically, the input t is first mapped to a positional embedding $\gamma(t)$ using the regular positional encoding procedure following the one-dimensional instantiation of Equation 1. The resulting positional embedding is then processed by an MLP to generate the output phase-shift vector $\phi(t)$.

Model Training. At each training iteration, we randomly sample one video frame V_i along with its frame index i, which is normalized to $[-1, 1]$ and passed through our model to generate the frame $\hat{V}_i$. The model is trained with the reconstruction-based loss function

$$L(\hat{V}_i, V_i) = ||\hat{V}_i - V_i||_1 + \lambda||\Phi(\hat{V}_i) - \Phi(V_i)||_2 \quad (4)$$

where $\Phi(.)$ denotes the feature maps extracted from the pre-trained VGG-19 network [28]. The loss function is composed of two loss terms: the conventional L_1 loss and the perceptual loss to encourage preserving better image details. $\lambda = 0.2$ is a weighing factor.



Input Direct-VINR Phase-NIVR (Ours)

Figure 3. Video-fitting examples. Our method can fit video content with comparable visual quality as Direct-VINR. This suggests that incorporating phase-varying positional encoding preserves the ability to fit video data. Please see video results at https://tinyurl.com/NIVR-VidFit.

During training, we found it beneficial to update M_f and M_p in an asymmetric manner. In particular, we update the parameters of both networks only on half the number of frames evenly sampled across the video. For the remaining frames, we only update the parameters of the phase-shift generation network M_p while freezing the parameters of M_f during back-propagation. In that way, M_f is prevented from overfitting to all the frames while still able to guide the update of M_p such that the predicted phase-shifts that can be correctly interpreted to generate the hold-out frames. We found such asymmetric training procedure critical for learning well-structured phase space.

4. Experiments

4.1. Implementation Details

We implement both M_f and M_p as MLPs with 5 hidden layers and 256 neurons per hidden layer. Following [29], we use the sine activation function in all hidden layers. For M_f, the output layer has three neurons, corresponding to the RGB color values. Each neuron has a tanh activation function to constrain the output value to $[-1, 1]$. For M_p, the number of output neurons is equal to twice the number of frequency channels in the positional encoding module of M_f. The number of frequency channels N in positional encoding is determined by the number of samples L along each dimension of the input video as $N = [\log_2(L) + i_0]$ to account for the Nyquist sampling rate. L is taken to be the length of the video for the temporal dimension and the smaller side of the frame for the spatial dimension. We use $i_0 = 1$ in Eq. 3 and 2 for all experiments. We trained our model using the ADAM optimization al-

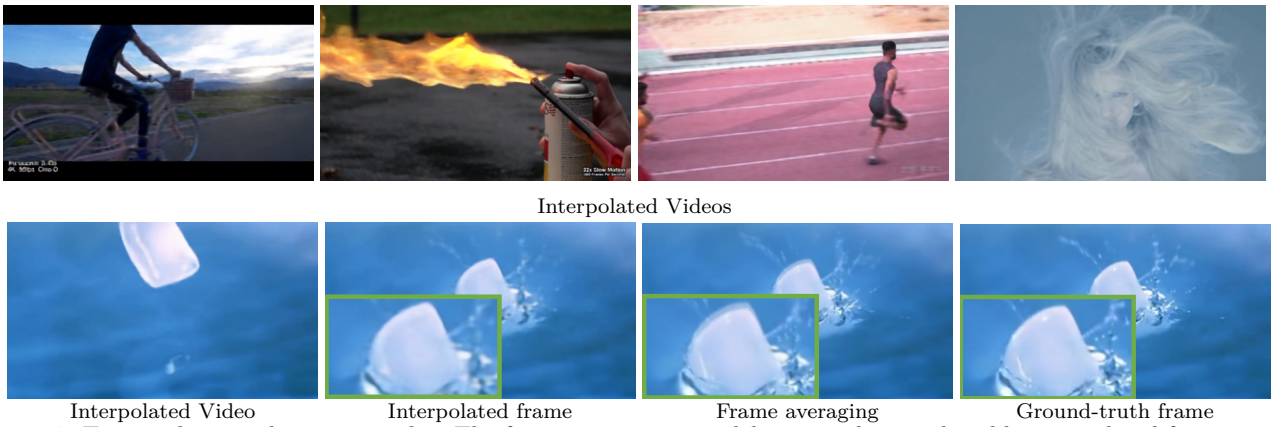


Figure 4. Temporal interpolation examples. The frame generation model can synthesize plausible interpolated frames with interpolated phase-shift vectors during inference time. The interpolation results often show plausible motion transition rather than copying nearby frames or taking frame-wise average (2nd row). Video results can be viewed at https://tinyurl.com/NIVR-VidInterp.

gorithm [11] with learning rate 0.0001 for 6,000 passes over an input video. It takes about 18 hours to train on a video of 120 frames with resolution 256×452 on one NVIDIA 2080Ti GPU.

4.2. Learning to Fit Video Data

We examine whether incorporating the phase-varying positional encoding and the generated phase shifts hurts the ability of the model to fit the video data well. Compared to standard INR formulation, it is more challenging for our model to fit the coordinate-to-color mapping as the positional embeddings of the input spatial coordinates constantly change across frames. We test our neural implicit video representation (Phase-NIVR) on 25 videos from the WAIC-TSR dataset [22] that covers different content and motion types. For each video, we use the first 120 frames and resize them so that the small side is 128-pixel.

For comparison, we also train a direct extension of INR to video, named Direct-VINR, that incorporates t as an additional input coordinate. We use the same model architecture as in our model to experiment with Direct-VINR. We train both models on each video in the dataset and compute the PNSR/SIIM reconstruction scores from their reconstructed videos. The results (32.05/0.905 vs 31.98/0.897) indicate that our method performs comparably with Direct-VINR. This suggests that incorporating phase-varying positional encoding, while making the learning problem more challenging for the mapping network, does not prevent the model from fitting the videos. Figure 3 shows 30-frame segments of some example reconstructed videos. Consistent with the numerical scores, we observe the reconstructed videos from two methods often have comparable visual quality.

4.3. Phase-based Motion Manipulation

The previous experiment shows the ability of our model to map per-frame phase information into the frame content. However, it is not clear whether the learned phase captures meaningful temporal dynamics structure or simply serves as an index for the model to memorize the frame content. In this section, we inspect how manipulating the generated phase-shift sequence $\phi(t)$ influences the change in the output frames.

Temporal Interpolation. We examine if interpolating two phase-shift vectors corresponds to a meaningful interpolation in the video domain. We sample five videos in the WAIC-TSR dataset that cover different scene types and have good reconstruction quality (PSNR ≥ 28.5) from the previous experiment. We retrain our model on 120 frames from each video sampled at half the original frame rate. For this test, we train the model on video frames resized to 256×452 so that more details can be observed. After training, we use M_p to generate the phase shift vectors at each time t and perform interpolation between each pair of consecutive phase-shift vectors to obtain the interpolated phase-shift sequence. We use spherical linear interpolation to account for the circular nature of phases [27]. The resulting phase-shift sequence is used in the frame generation module M_f to generate the final frames.

Figure 4 shows the interpolated video results. The videos were set to be played back at two frames per second in the figure for easier inspection. First, it can be observed that the interpolated frames have comparable visual quality as the original frames. This indicates that the model can indeed interpret the positional embedding from the interpolated phase-shift vectors into plausible video content rather than treating them as out-of-distribution samples. Second, the appearance of the frames continuously changes, indicating that the model can associate the change in the phase-shift vec-

Input Reconstruction Motion Smoothing

Figure 5. Motion filtering. Low-pass filtering the phase-shift sequence $\phi(t)$ at inference time can make the frame generation model to generate a new video with smoother object motion. Note the concrete base becomes more stable while its larger-scale motion is preserved. Video results can be viewed at https://tinyurl.com/NIVR-MoFilt.

tors to the change in the video domain rather than simply copying the content from the nearest frames. Finally, we inspect whether the interpolated frames are the results of the pixel-space average of the corresponding neighboring frames. We found that in general the interpolated frames are different from the frame-wise average results (Notice the ghosting around the ice cube in the averaging result in Figure 4 (bottom row)). We observe that when the motion is sufficiently small, the interpolated video does capture the interpolated motion. However, with a larger inter-frame motion, the model may not identify the corresponding large-moving regions across frames as part of a single motion. In those cases, interpolation tends to reduce to a blending operation, resulting in occasional ghosting artifacts as can be observed in the "running men" sequence (the third examples in Figure 4's top row).

Motion Filtering. The previous interpolation test suggests that the learned phase-shift vectors can be associated with the temporal states of the video content. We furthermore perform a simple experiment to test whether low-pass filtering the learned phase-shift sequence can smooth motion in the video. For this test, we collect videos that have some jiterring object motion on top of a longer-range motion trajectory such as a tunning fork vibrating while moving (Figure 1 (bottom-right)). After training our model on each video, we treat the generated phase-shift sequence as a multi-dimensional time series and apply a median filter with a temporal window-size of 7 to it. The filtered phase-shift sequence is used with the frame generation model M_f to synthesize the new video.

Figure 1 (bottom-right) and Figure 5 show two motion filtering results. More examples can be found in our supplementary video[1]. We observe that filtering the learned phase-shift sequence leads to the resulting videos with reduced high-frequency jittering while the larger-scale motion is preserved. Note the overall up-down motion of the tuning fork in Figure 1 (bottom-right) is retained while its vibration is largely removed. Also, the base concrete platform in Figure 5 is stabilized while its overall motion direction is preserved.

[1]https://tinyurl.com/NIVR-Supp

Motion Magnitude Adjustment. Inspired by phase-based motion processing works [38, 39], we are curious if manipulating the phase-shift vectors in our framework can alter the motion magnitude in videos. Specifically, we test whether adjusting the magnitude of the difference between neighboring phase-shifts can result in motion magnitude change.

We test our method on different videos with object fluctuating in space. For phase-shift adjustment, we first scale the difference between each consecutive phase-shift vectors $\Delta\hat{\phi}(t) = \alpha(\phi(t+1) - \phi(t))$. We then fix the phase-shift vector at the first frame and re-compute the phase-shift sequence with the modified pair-wise difference $\Delta\hat{\phi}(t)$. The new videos are synthesized from the modified phase-shift sequence. Figure 6 and Figure 1 (bottom left) show two example videos with different scaling factor α values. When α is smaller than one, the resulting video shows reduced motion magnitude, leading to the motion minification effect. The magnification effect was obtained with $\alpha > 1$. Note that while the motion magnitude was modified, the overall motion structure such as the direction of motion and different motion stages were preserved. We observe that the motion minification effect can be robustly obtained with any $\alpha < 1$ while the magnification effect tends to be more restricted as the model may not be able to extrapolate far beyond the motion range available in the input video.

Video Loop Detection. Hypothesizing that the phase-shift vectors encode the states of dynamics, we investigate if we can detect loops in videos with repeated motion by analysing the phase-shift sequence.

We adopt a simple approach to detect loops in a video from the learned phase-shift sequence. Let $\{\phi(k)\}_{k=0..N}$ represent the learned phase vectors from the video with N frames. We identify the looping point by determining the pair of frame indices $\hat{i}$ and $\hat{j}$ that minimize the cost function

$$\min_{i,j|j\geq i+\tau} \|\phi(i) - \phi(j)\| + \beta\|\Delta\phi(i) - \Delta\phi(j)\| \qquad (5)$$

where $\|.\|$ denotes the L_1 distance, $\Delta\phi(i) = \phi(i) - \phi(i-1)$ represents the phase-difference vector, and β is a weighing factor to balance between phase matching

Original ($\alpha = 1.0$) $\alpha = 0.5$ $\alpha = 1.5$

Figure 6. Motion magnitude adjustment. Scaling the phase-shift sequence $\phi(t)$ at inference time can alter the motion magnitude in the synthesized video. Varying the scaling factor allows for both motion minification and motion magnification. Video results can be viewed at https://tinyurl.com/NVIR-MoAdj.

and motion matching terms. τ determines the desired minimum length for the extracted loop. The idea is to determine the pair of as-similar-as-possible phase shifts that also have similar phase transition. After solving for $\hat{i}$ and $\hat{j}$, the sub-sequence $\{\phi(k)\}_{k=\hat{i}..\hat{j}}$ forms the candidate loop. The new video synthesized with this sequence would ideally transition from the $\hat{j}$-th frame back to the $\hat{i}$-th frame which has similar dynamics state, forming the illusion of looping.

In practice, we observe that perfect matching is only possible for simple mechanical motion where objects perfectly repeat themselves. For more organic motion such as human action, slight variation in object poses can cause a perceivable jump at the looping point. To address that problem, we further modify the phase-shift sequence with a simple phase blending process. We modify the first l phase-shifts by blending them with $\phi(\hat{j})$ using spherical linear interpolation with the blending weight $\alpha(n)$ of the n-th vector defined as $\alpha(n) = \frac{n}{l}$.

Figure 7 shows example loop extraction results from two potentially looping videos. Please check our supplementary video[2] for more examples. In general, the loop points can be successfully detected by phase matching. This indicates that the similar phase-shift vectors reflect similar scene states reappearing at different times. We note that phase blending helps improve the perceived looping noticeably. The wind-chime example (Figure 7 (bottom)) is particularly challenging to handle as the original video contains small camera motion. For that reason, no perfect loop point exists that can match both the background and the object motion, resulting in the noticeable temporal seam in the looping result. Surprisingly, with phase blending it is possible to achieve a seamless looping video. This indicates that manipulating in the phase-shift space can lead to plausible modification in the video domain.

5. Discussion

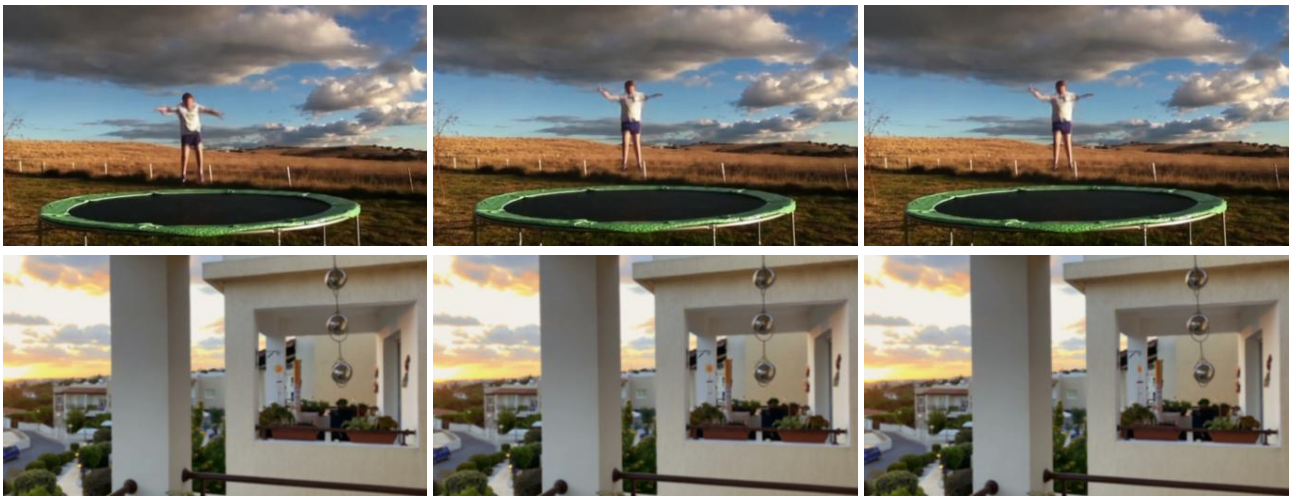
Our experiments suggest that the learned phase-shifts $\phi(t)$ can be associated with meaningful transition in videos. We visualize $\phi(t)$ as a function over time. Figure 8 shows such a visualization for our model trained on a video with structured and symmetric nature while containing some complex motion. We show the phase-shift visualization for 5 out of 14 phase-shift series (corresponding to 14 M_f's frequency channels).

Inspecting the visualized phase shifts, we can see that the phase-shift series evolve smoothly over time rather than forming a sequence of uncorrelated states. More interestingly, we found that the phase-shift series are well structured. The phase-shift plots contain highly symmetric patterns, reflecting the symmetric nature of this particular input video. In most videos that we experimented with we also observe that the transition in the phase-shift series often corresponds in meaningful ways to the transition in the visual domain.

Occasionally, we also observe a localized-control capability. For this example, we found that the fifth phase-shift series (the red curve in Figure 8) correlates with the movement of the hair-lock region. Keeping the fifth phase-shift evolving over the whole temporal range while freezing the phase-shift values of all other frequency channels at certain keyframes results in the re-synthesized scenes frozen at the selected keyframes while keeping the hair lock moving in similar ways (Figure 8 (bottom)). This localized-control behavior is interesting as it suggests the simple MLP networks can potentially learn non-trivial spatial-temporal relations from raw visual data without explicit motion, correspondence, or semantic supervision. However, we would like to note that our current model does not exhibit this localized-control ability on all videos. In general cases, one phase-shift series often correlates with more global motion, and the motion of one visual element is often influenced by multiple phase-shift channels. Explicitly encouraging such localized-control capability by incorporating specialized training strategy would be an interesting direction for future exploration.

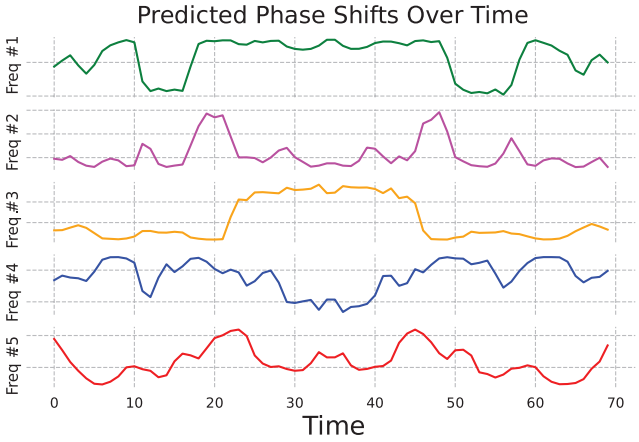
Limitation and Future Work. Our study demonstrates the surprising effectiveness of using phase-varying positional encoding in image-based INR to capturing temporal dynamics. However, our method has several limitations. First, while our model can

[2]https://tinyurl.com/NIVR-Supp



Original Looping w/o Phase Blending

Figure 7. Video loop detection. Potential repeat point in a video can be detected by simple phase-matching strategy in the learned phase-shift sequence $\phi(t)$. Applying phase blending improves the looping results especially for challenging scenarios, such as when both the wind chime and the background move due to subtle camera motion (bottom row). See video results can be viewed at https://tinyurl.com/NIVR-VidLoop.



Reconstructed Video Freq 5 only (keyframe at $t = 0$) Freq 5 only (keyframe at $t = 40$)

Figure 8. We visualize five channels of the learned phase-shift values $\phi(t)$ as a function of time (top). The structure of the phase-shift series reflects the symmetric nature of the video (bottom-left). In addition, the fifth phase-shift series (the red curve) correlates with the hair-lock movement even when other channels are frozen to one keyframe. Please see video results at https://tinyurl.com/NIVR-PhaseViz.

fit a video, the reconstruction quality is not perfect. Our reconstructed videos are often slightly more blurry and sometime noisy compared to input videos, as can be seen from the video results. Incorporating more advanced frequency selection principles in positional encoding layers [29, 34] or employing local implicit function models [5, 14, 15] are promising directions to improve the visual quality. Second, as our framework requires example-specific training, it takes many hours to process one video. Extending our method to multiple-video settings with hyper-networks models [30] or meta-learning [33] can be fruitful directions to explore in future work. Finally, while our motion-adjustable neural implicit video representation shows promising results for various motion editing tasks, our method is not optimized for each specific application and the achievable effects may depend on the content and motion in the original input video. We believe that incorporating application-specific domain knowledge specialized for each task can potentially improve the robustness of our method.

6. Conclusion

In this work we introduced a motion-adjustable neural implicit video representation. We extended image-based implicit neural representation to model video data by making the positional encoding component phase-varying, coupled with the jointly learned phase-shift generator. We showed that such extensions can retain the ability to learn from video data. Interestingly, the resulting representation also allows for modifying temporal dynamics in a video through phase-shift manipulation, enabling meaningful effects such as temporal interpolation, motion-magnitude adjustment, motion filtering, and video-loop extraction. We hope our findings can motivate further research on exploring the potential of Fourier-based positional encoding and its phase information in learning visual dynamics within the context of implicit neural representation.

Acknowledgments We thank Mr. Alessandro Michelazzi for the permission to use his video in our experiments (Figure 8).

References

[1] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021. 1, 2

[2] Jonathan T. Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P. Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5855–5864, October 2021. 2

[3] Mojtaba Bemana, Karol Myszkowski, Hans-Peter Seidel, and Tobias Ritschel. X-fields: Implicit neural view-, light- and time-image interpolation. *ACM Trans. Graph.*, 39(6), nov 2020. 2

[4] Eric R. Chan, Marco Monteiro, Petr Kellnhofer, Jiajun Wu, and Gordon Wetzstein. Pi-gan: Periodic implicit generative adversarial networks for 3d-aware image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5799–5809, 2021. 2

[5] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 1, 2, 8

[6] Emilien Dupont, Adam Golinski, Milad Alizadeh, Yee Whye Teh, and Arnaud Doucet. COIN: COmpression with implicit neural representations. In *Neural Compression: From Information Theory to Applications – Workshop @ ICLR 2021*, 2021. 1, 2

[7] David J. Fleet and A. D. Jepson. Computation of component image velocity from local phase information. *Int. J. Comput. Vision*, 5(1):77–104, 1990. 3

[8] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Niessner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8649–8658, June 2021. 3

[9] Chen Gao, Ayush Saraf, Johannes Kopf, and Jia-Bin Huang. Dynamic view synthesis from dynamic monocular video. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 3

[10] T. Gautama and M.A. Van Hulle. A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Transactions on Neural Networks*, 13(5):1127–1136, 2002. 3

[11] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. 5

[12] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[13] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *IEEE International Conference on Computer Vision*, 2021. 2, 3

[14] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *ACM Trans. Graph.*, 40(4), 2021. 2, 8

[15] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14214–14223, 2021. 1, 2, 3, 8

[16] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2019. 2

[17] Simone Meyer, Abdelaziz Djelouah, Brian McWilliams, Alexander Sorkine-Hornung, Markus Gross, and Christopher Schroers. Phasenet for video frame interpolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[18] Simone Meyer, Oliver Wang, Henning Zimmer, Max Grosse, and Alexander Sorkine-Hornung. Phase-based frame interpolation for video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1410–1418, 2015. 3

[19] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision*, 2020. 1, 2, 3

[20] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 2

[21] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. *ICCV*, 2021. 3

[22] Liad Pollak Zuckerman, Eyal Naor, George Pisha, Shai Bagon, and Michal Irani. Across scales across dimensions: Temporal super-resolution using deep internal learning. In *European Conference on Computer Vision*. Springer, 2020. 5

[23] S.J.D. Prince. *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012. 2

[24] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural Radiance Fields for Dynamic Scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 3

[25] Shunsuke Saito, Tomas Simon, Jason Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned im-

plicit function for high-resolution 3d human digitization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2020. 2

[26] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. In *Advances in Neural Information Processing Systems*, 2020. 2

[27] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, page 245–254, 1985. 5

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 4

[29] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, 2020. 1, 2, 3, 4, 8

[30] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *Advances in Neural Information Processing Systems*, 2019. 1, 2, 8

[31] Ivan Skorokhodov, Savva Ignatyev, and Mohamed Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021. 2

[32] Richard Szeliski. *Computer Vision - Algorithms and Applications*. Texts in Computer Science. Springer, 2011. 2

[33] Matthew Tancik, Ben Mildenhall, Terrance Wang, Divi Schmidt, Pratul P. Srinivasan, Jonathan T. Barron, and Ren Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2846–2855, 2021. 2, 8

[34] 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. *NeurIPS*, 2020. 1, 2, 3, 8

[35] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a dynamic scene from monocular video. In *IEEE International Conference on Computer Vision*, 2021. 3

[36] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15182–15192, October 2021. 2

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017. 3

[38] Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T. Freeman. Phase-based video motion processing. *ACM Trans. Graph.*, 32(4), 2013. 3, 6

[39] Neal Wadhwa, Hao-Yu Wu, Abe Davis, Michael Rubinstein, Eugene Shih, Gautham J. Mysore, Justin G. Chen, Oral Buyukozturk, John V. Guttag, William T. Freeman, and Frédo Durand. Eulerian video magnification and analysis. *Commun. ACM*, 60(1):87–95, dec 2016. 3, 6

[40] Wenqi Xian, Jia-Bin Huang, Johannes Kopf, and Changil Kim. Space-time neural irradiance fields for free-viewpoint video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9421–9431, 2021. 3

[41] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems 32*, pages 492–502. 2019. 2

[42] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv:2010.07492*, 2020. 2