

Neural Residual Flow Fields for Efficient Video Representations

Daniel Rho¹[0000–0002–8568–9489], Junwoo Cho¹, Jong Hwan Ko^{1,2*}, and
Eunbyung Park^{1,2*}

¹ Department of Artificial Intelligence, Sungkyunkwan University

² Department of Electrical and Computer Engineering, Sungkyunkwan University
{daniel231, jwcho000, jhko, epark}@skku.edu

Abstract. Neural fields have emerged as a powerful paradigm for representing various signals, including videos. However, research on improving the parameter efficiency of neural fields is still in its early stages. Even though neural fields that map coordinates to colors can be used to encode video signals, this scheme does not exploit the spatial and temporal redundancy of video signals. Inspired by standard video compression algorithms, we propose a neural field architecture for representing and compressing videos that deliberately removes data redundancy through the use of motion information across video frames. Maintaining motion information, which is typically smoother and less complex than color signals, requires a far fewer number of parameters. Furthermore, reusing color values through motion information further improves the network parameter efficiency. In addition, we suggest using more than one reference frame for video frame reconstruction and separate networks, one for optical flows and the other for residuals. Experimental results have shown that the proposed method outperforms the baseline methods by a significant margin.

1 Introduction

Neural fields [1,2,3,4,5,6] (also known as implicit neural representations or coordinate-based neural representations) are an emerging approach for representing various signals. Signals can be reconstructed using dense sampling once a neural network has been trained to map coordinates to corresponding signal values. Unlike other representation techniques that store discretely sampled data, neural fields use continuous coordinates as inputs, allowing them to represent signals at any resolution and at any arbitrary coordinate. It can accurately express both low and high frequencies of signals thanks to recent breakthroughs in input features [1,2,5]. This innovative representation approach has shown considerable promise in a number of areas, including computer graphics [5,7], physical simulations [2,8], and generative models [9,10], to name a few.

* Corresponding authors.

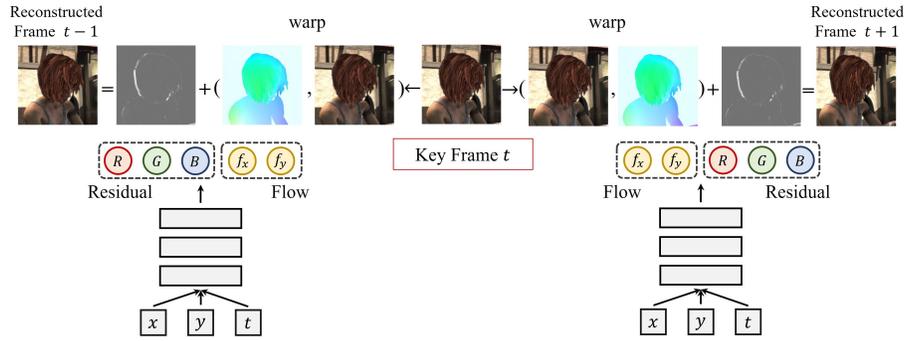


Fig. 1: Neural residual flow fields (NRFF) with a single reference frame. The gray-scale image on each side shows residuals, and the blue image in each parenthesis are optical flows. x, y, t are spatial and temporal coordinates.

Although it has recently received widespread attention, its parameter efficiency has not been thoroughly investigated. In recent studies, neural fields require a large number of parameters to accurately represent signals [2,4,5]. Without enhancing parameter efficiency, transaction costs of neural fields are high because signals are stored as network parameters. This hinders us from utilizing it in many practical applications that may benefit from it.

In this paper, we study how to effectively represent videos using this new representation approach. A naive approach, equivalent to SIREN [2], would be to use a neural field as a function of spatial and temporal coordinates, $(r, g, b) = f_{\theta}(t, x, y)$, with continuous coordinates as inputs and three color channels as outputs. However, this approach does not exploit the spatial and temporal redundancy of video signals, and our goal is to improve the parameter efficiency by explicitly removing the redundancy.

We propose *Neural Residual Flow Fields (NRFF)*, a novel neural field scheme for video representation that leverages optical flows and residuals instead of raw colors. Our proposed scheme was inspired by standard video compression algorithms [11,12,13] that use motion information to deduplicate signals presented across frames. Optical flows allow us to reuse color values from other reference frames, which often preserve fine details. If those delicate patterns are on the surface of subjects inside video frames, reusing color values relieves the burden of learning similar patterns across frames and improves network parameter efficiency, rather than wasting network capacity on storing redundant raw signals for the entire frames.

Many video compression methods use block-wise motion estimations, where each motion vector is applied to all pixels within each block, significantly reducing the total number of required motion vectors. This is based on prior knowledge that smooth, low-frequency motion vector fields are often sufficient for describing movement between video frames. This gave us the idea that substituting optical flows for raw colors may greatly reduce the number of parameters. However,

video frames cannot be completely reconstructed solely by optical flows due to occlusion and dis-occlusion. Thus, we use *residuals* to recover the original signals from video frames with precision. It would not necessitate a large neural network since a substantial number of color values are likely to be reused from the motion information. To summarize, we train the networks to capture optical flows and residuals rather than raw color signals, which greatly improves the efficiency of network parameters.

We also propose to use more than one reference frame for video frame generation. Using multiple reference frames enables the exploitation of visible information over many reference frames, each of which may contain distinct exposed and occluded information.

In addition, we also suggest splitting the network into two subnetworks: one for optical flows and the other for residuals. Separating optical flows and residuals, assuming they have different dynamics, would improve the quality, and the experiment results support this.

Experimental results show that the proposed method significantly outperforms the baseline method, which relies on raw colors. Given similar sizes, *NRFF* reconstructs video frames more clearly and sharply. Quantitative results on the MPI Sintel [14] and UVG [15] datasets reveal that our method significantly outperformed its counterpart in terms of standard image reconstruction metrics (PSNR: 31.2 to 37.4, SSIM: 0.82 to 0.95). Although the proposed method is an initial attempt to improve the parameter efficiency of neural fields for videos, it also performs favorably with H.264 [11], a standard video compression algorithm. With the multi-reference frames method, *NRFF* matches the performance of H.264 using small group of pictures (GOP) on some videos without any model compression techniques such as pruning and entropy coding (Fig. 6).

In summary, our contribution is threefold.

- We show that using optical flows and residuals as output instead of colors can significantly improve video quality.
- We propose to use multiple reference frames for frame reconstruction, and this improves video quality without increasing the network size.
- We demonstrate that parameter efficiency can be improved by using separate neural fields—one for optical flows and the other for residuals—in addition to using a shared network for each group of pictures.

2 Related Works

Neural fields Neural fields map spatial and temporal coordinates to certain physical quantities [16]. Since a wide variety of tasks can be represented as fields, this approach has recently gained popularity and been used in several tasks, such as image representation [1,4,17], audio representation [2], 3D shape [7,18], and novel view synthesis [5,19,20,21,22]. Thanks to recent innovations [1,2,5,17,23], it can faithfully reconstruct even high frequency signals. Neural fields have also been applied to signals having both spatial and temporal dimensions, including video representation and novel view synthesis in 4D space [24,25,26,27,28].

As a new way of representing data, several attempts have been made to compress various signals, such as images [29,30,31] and videos [26,32]. However, the compression performance of neural field-based methods is currently far behind standard state-of-the-art compression algorithms.

Learning based video compression There have been several data-driven attempts to utilize neural networks for efficient video representation. Convolutional neural networks (CNNs) and auto-encoder architectures have been used to compress video signals [33,34]. These works train encoder and decoder networks on large-scale datasets and test them on unseen videos to achieve high video compression rates, assuming decoder networks are already shared and only core video information needs to be stored or sent. DVC [34] has achieved compression rates comparable to or slightly better than standard video compression algorithms, e.g., H.264 and H.265. However, these approaches are inherently vulnerable to the biases of training datasets. SIREN [2] is an attempt to represent various forms of signals, including videos, through neural fields, however, it does not consider its parameter efficiency. NeRV [26] is a variant of neural fields that achieves video compression performance comparable to that of the standard video compression algorithm, H.264. However, NeRV gives up two degrees of freedom and uses only time coordinates t as inputs for efficient rendering. Our proposed method, *NRFF*, explicitly removes the redundancy by combining residuals and flows with the help of key frames. IPF [32] is a concurrent work using optical flows and residuals for video compression. Ours and IPF are different in a number of ways. While IPF employs a separate network for each frame, we use a shared network for each group of pictures and take advantage of temporal redundancy between frames. We also propose to use more than one reference frame, whereas IPF only proposes to use one.

Optical flow estimation Optical flow has been a core component of various computer vision tasks. Since the work of Horn and Schunck [35], many improvements have been proposed to make optical flow more accurate [35]. Recently, employing neural networks to improve optical flow estimation [36,37,38] rather than traditional algorithm-based methods [35] has been successful. To achieve more robust learning based optical flow methods, the ground truth optical flows have been collected by using an animated film and computer graphics [14,36]. Several strategies have been proposed to improve estimating performance by using occlusion masks [39] or transformer-based operations [40,41].

3 Method

Figure 1 illustrates an overview of optical flow-based neural fields for video representation. Our proposed neural fields generate optical flows for a given spatial and temporal coordinates (Sec 3.1) to warp reference frames. In addition to optical flows, the residuals are also generated by neural fields, and these generated

residuals are added to the warped frame to complete the reconstruction (Sec 3.2). To improve video quality, we use more than one reference frame for video frame reconstruction. For each GOP, we use two neural networks: one for optical flows and the other for residuals.

3.1 Dense Optical Flow Estimation

Standard video compression methods use block-wise motion information to improve compression efficiency [11,13]. However, because of this algorithmic nature, results often contain block-shaped distortions or artifacts, which necessitate deblocking filters. We propose using dense, pixel-wise motion vectors through neural fields. It is known that neural networks can efficiently represent smooth signals [1,42]. Thus, we sample motion vectors densely by injecting dense coordinates and using small-size neural networks as neural fields.

3.2 Image Warping and Completion

We use generated flow fields to warp reference frames to predict target video frames. Let I be an image function that takes spatial and temporal coordinates (x, y, t) as inputs and produces corresponding colors (r, g, b) as outputs. A reference frame and a warped image are denoted as \hat{I}_{ref} and \tilde{I} , respectively. The reference frame can be the key frame or a neighboring frame. Then, a warped image at time t , $\tilde{I}(x, y, t)$ can be written as

$$(\Delta x^t, \Delta y^t) = F_{\text{flow}}(x, y, t; \theta), \quad (1)$$

$$\tilde{I}(x, y, t) = \text{Interp}(\hat{I}_{ref}(t), x + \Delta x^t, y + \Delta y^t). \quad (2)$$

$F_{\text{flow}}(t)$, which is parameterized by θ , estimates optical flows between two frames (the video frame to be reconstructed and the reference frame). $\text{Interp}(\cdot)$ warps a video frame by a bicubic interpolation.

Reconstructing a video frame by simply warping the source video frame is likely to contain artifacts. Artifacts can be caused by a variety of factors, including imprecise optical flow estimation, occlusions and disocclusions between video frames, and accumulated errors from the recursive frame generation process. To alleviate the issues, we use residuals along with the optical flows.

The final equation for image completion can be written as

$$\hat{I}(x, y, t) = F_{\text{res}}(x, y, t; \psi) + \tilde{I}(x, y, t) \quad (3)$$

, where F_{res} denotes the residual estimator with its own parameter ψ .

3.3 Key Frames

Inspired by the standard video compression algorithms, we store a key frame per GOP as a standalone frame so that other frames directly or indirectly depend

on the key frame. Regarding key frames, there are two important considerations: the image quality of key frames and the location of key frames.

With a fixed total size, there is a trade-off between the key frame quality and the network size. Given a high-quality key frame, a network can exploit the high-frequency details of the key frame. However, since high-quality key frames require a larger memory size, the network size must be smaller in order to maintain the total size. Having a small-sized network tends to have difficulties covering long, dynamic frames. In contrast, a relatively low-quality key frame relegates expressing fine-details to the network. A large network handles long dynamic frames relatively easily, however, because the reference frames lack fine-details, the network must learn to compromise between learning fine-details and learning the optical flows of long and dynamic frames. We empirically found that the optimal ratio of the network size and the key frame quality (or the size) is related to the total number of frames in the GOP. The larger the GOP is, the larger the network size should be, and vice versa. The detailed experimental results can be found in the supplementary material.

Among possible positions, we chose the middle frame as the key frame to minimize the errors caused by the distance from the key frame. Experimental results showed that selecting the middle frame as the key frame is better in terms of reconstruction quality than selecting the first or last frame in the GOP.

We could use neural fields for key frame compression. However, existing neural field-based image compression methods usually underperform or are at most similar to standard image compression algorithms, such as JPEG, in terms of compression efficiency. Therefore, for keyframe compression, we employ the standard H.264 video compression technique. We still need more sophisticated training algorithms, input feature preprocessing techniques, and new network architectures to completely replace H.264 with neural fields in our proposed method.

3.4 End-to-end Training

Our proposed approach was designed to be differentiable throughout the entire process in order to build an end-to-end framework for video representation learning. We used MSE (mean squared error) loss to minimize the reconstruction errors. Let B be a mini-batch of frame indices excluding the key frame, then the loss function can be written as

$$L(\theta, \phi, \psi) = \frac{1}{Z} \sum_{t \in B} \sum_x \sum_y \|I(x, y, t) - \hat{I}(x, y, t)\|^2. \quad (4)$$

$I(\cdot, \cdot, \cdot)$ is a ground truth color, and $\hat{I}(\cdot, \cdot, \cdot)$ is a reconstructed color. We multiplied the equation with the inverse of the constant Z to get the average loss over both time and space. Note that we excluded the key frame during the training process. The key frame will only be included during the evaluation.

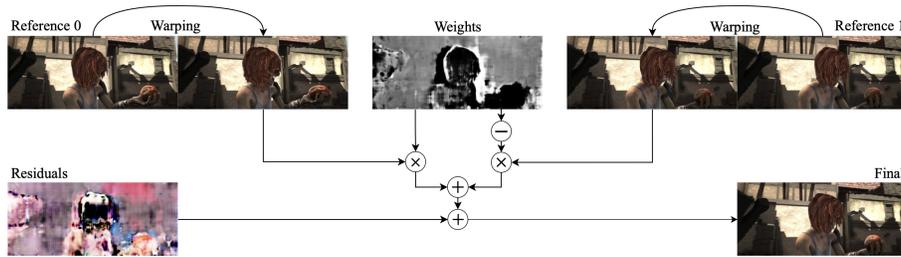


Fig. 2: Neural residual flow fields with multi-reference frames. \otimes and \oplus denote element-wise multiplication and addition, respectively. \ominus in the figure denotes one minus inputs, that is, in the case of weights, one minus weights will be outputted.

3.5 Multi-reference Frames

To further improve the video quality, we propose to use more than one reference frame to reconstruct a video frame. For each video frame, the video representation network learns to warp the nearest two key frames, each of which is unique per GOP, to reconstruct a video frame. The key frame from the GOP, where a particular video frame to be reconstructed is located, is always referred to for frame reconstruction. A frame, preceding the key frame of the GOP in temporal order, uses the key frame from the nearest preceding GOP as the other reference. Likewise, a frame after the key frame will use the key frame of the nearest posterior GOP as another reference frame.

Overall architecture is presented in Fig. 2. First, two reference frames are warped by the generated optical flows. To combine information from multiple frames, we also need learned pixel-wise weights (from zero to one) that selectively aggregate two warped frames. The network would weigh more on closer pixels among two reference frames to aggregate warped images. The flow network now generates optical flow outputs for each reference frame and additional mixing weights. Lastly, the residuals are added to fully reconstruct a video frame.

3.6 Network Split

Simply replacing the output of neural fields from colors with optical flows and residuals means that optical flows and residuals are generated from the shared parameters. If the patterns of optical flows and residuals are similar, then sharing parameters may help to improve parameter efficiency. Otherwise, sharing parameters could degrade the quality. We analyze two different structures in the experimental section.

As proposed in IPF, we can also split in the temporal dimension. That is, we can use each network for each video frame. However, unlike the relationship between optical flows and residuals, neighboring video frames are usually highly correlated. To support our claim that separating the network for each video frame

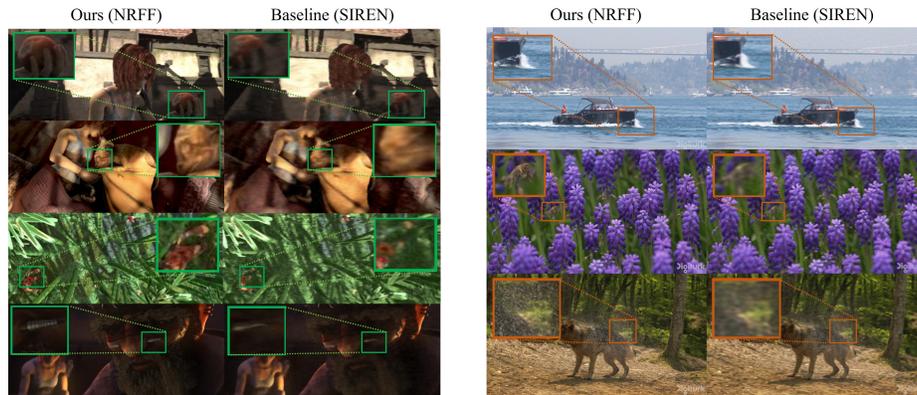


Fig. 3: The qualitative results between the proposed method and the baseline method under similar sizes (best viewed in color electronically). The left column examples are from the MPI Sintel dataset, and the right column examples are from the UVG dataset.

prevents the use of temporal redundancy and, thus, is less parameter efficient, we compare our method with IPF in the experimental section.

4 Experiments

We tested our approach on both synthetic and real-world datasets (MPI Sintel [14], UVG dataset [15]). We used the color-based neural field, SIREN [2], as the baseline. In addition to SIREN, we also compared our approach with the standard video compression algorithm, H.264 [11].

4.1 Dataset

The MPI Sintel [14] dataset was originally designed to evaluate the optical flow performance and it provides challenging and natural video sequences based on an open source animated film. We selected the MPI Sintel dataset for two reasons. First of all, it provides the ground truth optical flow, which allows us to evaluate the accuracy of the learned flows. Flow estimation is a core part of our algorithm, therefore, it is desirable to understand how flow estimation affects the final reconstruction. Second, it has been a good testbed for challenging scenarios, such as long range motion, motion blur, multi-frame analysis, and non-rigid motion. We tested our method on the entire 23 videos supported by MPI Sintel.

We also evaluated our method using seven 1080p videos from the Ultra Video Group (UVG) video collection [15], in order to compare it to other neural field-based video compression methods [26,32]. The dataset includes a variety of videos, ranging from nearly static scenes to fast-moving scenes.

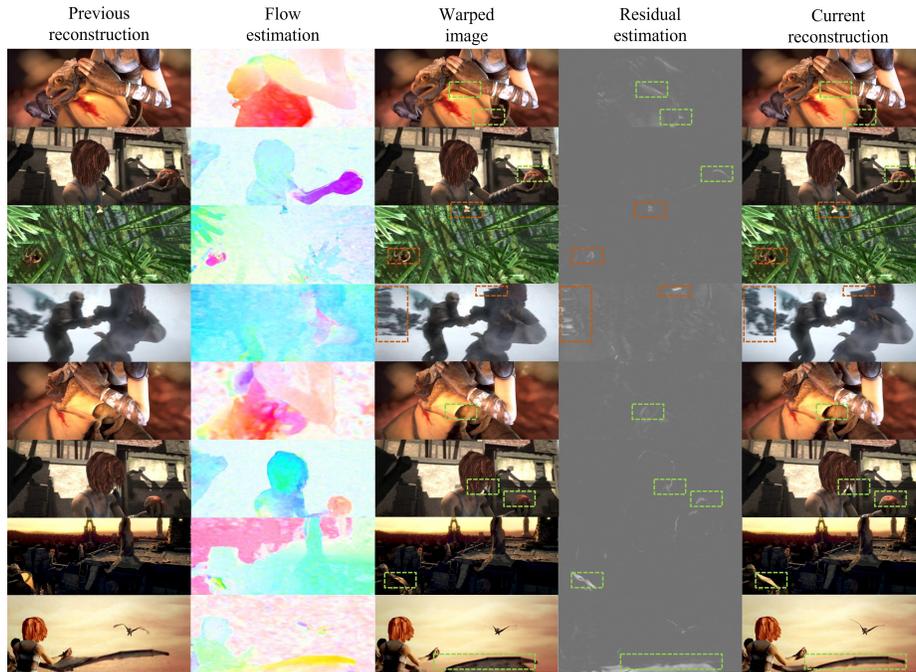


Fig. 4: The qualitative results on the MPI Sintel dataset.

4.2 Single Reference Frame Experiments

Experimental setup First, we used the SIREN backbone model and only changed the output linear layer to evaluate the effect of replacing colors with optical flows and residuals. For frame reconstruction, we applied optical flows recursively; that is, starting from the key frame, the reconstructed frame is used as the reference frame of the next frame. We used 16-bit precision weights for both ours and the baseline (SIREN) to halve the network size without quality degradation.

Our proposed method and the baseline are evaluated using the MPI Sintel video dataset. The resolution of all videos was reduced by half, resulting in 218×512 pixels. We set the size of the GOP to seven, so that each key frame covers six neighboring frames. In the case of a video containing 28 frames, for example, we trained four models for the video. We trained each model for total 30K iterations, and training the whole batch of frames at once was considered a single iteration. We used the Adam optimizer with a learning rate of 0.0005.

Results As shown in Fig. 3, replacing colors with optical flows and residuals significantly improves video quality for similar model sizes. Our method effectively preserves fine details such as a person’s beard, water drops, and bees, whereas the baseline method generates blurry outputs.

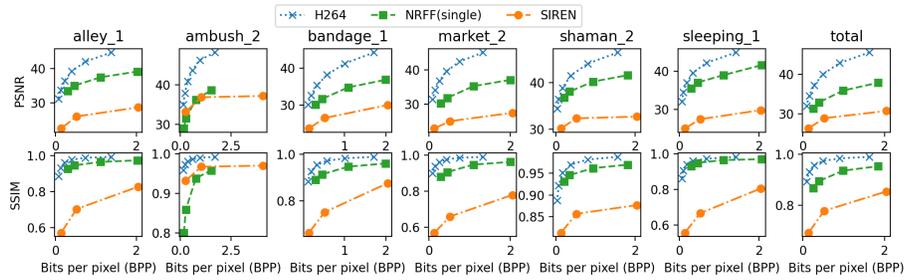


Fig. 5: The quantitative results of single reference frame experiments on the MPI Sintel dataset.

Figure 4 shows the qualitative results of single reference frame experiments in detail, including intermediate steps to complete video frames. The parts that need to be corrected by residuals are highlighted in figure 4. As we expected, the optical flow estimation is not necessarily accurate enough to reconstruct a video frame correctly. In fact, the final optical flow estimation includes many artifacts, particularly in the background regions. The residuals can successfully compensate for those artifacts created by inaccurate optical flows.

Due to the page limits, we presented the results of six videos and the average of all 23 videos in Fig. 5. We used two commonly used metrics in image reconstruction tasks; peak signal noise ratio (PSNR) and structural similarity index measure (SSIM). On average, our method enhanced PSNR from 30 to 37 and SSIM from 0.85 to 0.95 at around 2 bits per pixel. For example, in *alley_1*, we obtained 39.09 PSNR and 0.972 SSIM, as opposed to the baseline performance of 28.71 PSNR and 0.827 SSIM under similar model sizes. In *sleeping_1*, we got 41.56 PSNR and 0.970 SSIM, while the baseline only reached 29.78 PSNR and 0.806 SSIM. As illustrated in Figure 3, this resulted in significant increases in visual quality.

There are a few exceptions where we did not gain much improvement. One example is demonstrated in the fourth row of Figure 4. The scene is foggy and blurry, and the video shows a lot of camera movement. These factors, we believe, are the causes of the slightly lower quality of our approach. We hypothesize that, given a fixed number of parameters, inaccurate optical flow estimation may not be sufficiently compensated by residuals in some scenes. We also reported the performance of the H.264 video compression method [11] for your information.

4.3 Multi-reference Frames and Network Splitting Experiments

Experimental setup This section analyzes the effect of using multi-reference frames and separating networks on video quality. We replaced the activation function of the neural networks from sinusoidal activation [2] to swish function ($f(x) = x \times \text{sigmoid}(\beta x)$) and used positional encoding [5]. Furthermore, we reused hidden layers to improve parameter efficiency. We used H.264 for

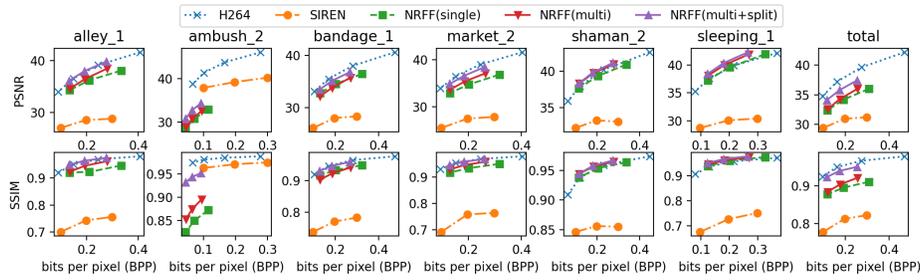


Fig. 6: The quantitative results of multi-reference frames experiments on the MPI Sintel dataset. The last column is the average performance on all 23 videos.

key frame compression, and the middle frame of the GOP was chosen as the key frame. To modulate the total size for video representation, we controlled the quality factor in H.264. As for the network size, each network width was automatically set so that the total size of the network is proportional to the corresponding key frame size.

We ran experiments using full-resolution MPI Sintel videos. We set the size of the GOP to five for every method, including H.264. We also applied 16 bit precision in this experiment to both the baseline (SIREN) and our proposed methods for a fair comparison. We trained each model for 5K epochs, and the batch size was set to one. Since the single reference NRFF becomes unstable when the batch size is one, the whole batch of the GOP was used as the minibatch for the single reference NRFF and the number of epochs was therefore increased to 25K. As for the initial learning rate, we used optimal learning rate for each method: $1e-3$ for SIREN and the single reference NRFF, and $1e-2$ for the multiple reference NRFF. The NRFF network size was set to be equal to one-fourth of the key frame size. For example, a network with a key frame size of 10k bytes was set to have approximately 1,250 parameters of 16-bit precision.

We also evaluated each method using videos from the UVG dataset (the first 100 frames of each). In this experiment, we unbridled the GOP size limitation for SIREN and H.264. That is, we use the automatically chosen GOP size in the case of H.264, which is 100, and we use the same size for SIREN. For our method, the GOP size was set at 15. Since we empirically found that large GOP size requires an increased ratio of the network size and the key frame, we set the ratio to be approximately 0.5.

Results Figs. 6 and 7 show the quantitative results on the MPI Sintel and UVG datasets, respectively. The supplementary materials provide all of the experiment results on the MPI Sintel. First of all, the results indicate that using optical flows can enhance video quality for both synthetic and non-synthetic scenes, including static and dynamic scenes. Second, referencing more than one frame enhances video quality without increasing the network size. Lastly, dividing the network into two subnetworks never degrades video quality and can significantly

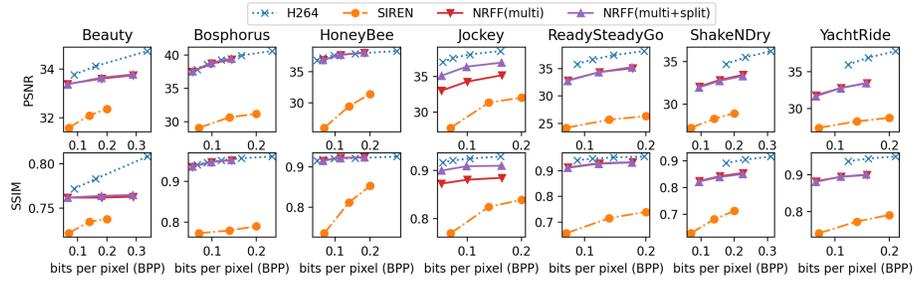


Fig. 7: The quantitative results of multi-reference frames experiments on the UVG dataset.

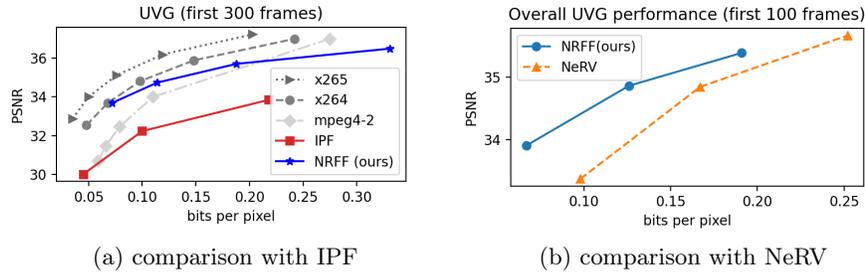


Fig. 8: Overall compression performance on the UVG videos. All results in fig. 8a were copied from IPF [32] except NRFF (ours).

improve quality for some videos. This demonstrates that our assumption that optical flows and residuals have different dynamics is valid. Even without network compression methods, our method performs on par with or better than the H.264 algorithm on some videos, such as *shaman_2* and *sleeping_1* from MPI Sintel, and Bosphorus and HoneyBee from UVG. The optical flow-based method might worsen the video quality in some videos, such as *ambush_2*, which contains fast and large movements. However, the overall performance can be significantly improved by using optical flows instead of raw colors, as shown in Fig. 7, and the rightmost column of Fig. 6

4.4 Comparison with Other Neural-field-based Video Representation

Experimental setup We compared our method (multi-reference model with two subnetworks) to other neural field-based methods (IPF [32], NeRV) using the UVG dataset. Since these two methods are evaluated in different experimental settings, we separately compared them with ours. First, to compare ours with IPF, we used the same experimental settings as in IPF (first 300 frames with the size of the group of pictures (GOP) of five) to train our models. To compare ours with NeRV, we compared the results of publicly available NeRV codes in

our experimental settings (first 100 frames for 1,500 epochs, and no compression techniques other than 16-bit precision).

Results Fig. 8 shows how efficiently our proposed neural fields can express a video compared to other neural field-based methods, including the current state-of-the-art method, NeRV. Our method outperforms a concurrent flow-based neural fields, IPF by a large margin, as demonstrated in fig. 8a. This performance gap is significant considering that IPF has adopted additional compression techniques, such as quantization and entropy encoding, to improve compression performance, while ours does not. We conjecture that this performance gain pertains to using a shared network per GOP.

Fig. 8b shows that replacing raw colors with residuals and flows results in better performance, especially in the low bits per pixel (bpp) region, even without enhancing network structure. Please note that NeRV relies on a neural field architecture that limits spatial sampling and only allows temporal sampling to improve compression performance, while ours permits spatial sampling.

4.5 Spatial and Temporal Interpolation

The neural fields have several advantages that no other video compression method has, one of which being the ability to sample values from arbitrary spatial and temporal coordinates, even at an unobserved point during encoding. To show this advantage of neural fields, we ran two experiments: spatial and temporal interpolation. For spatial interpolation, we first trained a neural network to represent a low-resolution video (with a resolution of (480, 270)). After that, without any post-processing methods, we simply upsampled the resolution four times in both height and width by sampling values in a much more dense grid. This is possible because neural fields take continuous coordinates as inputs. For temporal interpolation, we trained a neural network as in the main experiment with the fixed reference frames as for the multi-reference frame experiments. And then, the intermediate frame (for example, a frame in between the 5th and 6th frame) was sampled simply by injecting the corresponding temporal coordinates.

As shown in Fig. 9, simple dense sampling results in smooth interpolation in both time and space, even without any modifications and extra training techniques. Spatial interpolation of neural fields results in much smoother outputs than bilinear interpolation. For a fast-moving scene, interpolating two adjacent frames results in a blurry frame. However, our proposed method manages to represent the intermediate frame much more clearly. In addition to the fact that NRFF inherits the good properties of neural fields, the proposed method can offer new opportunities to improve the parameter efficiency of neural fields in other domains, such as NeRF [5] and light-field imaging, to name a few.

5 Conclusions and Discussion

We present a way to exploit neural fields for efficient video representations. The video quality was greatly enhanced by explicitly leveraging reference frames

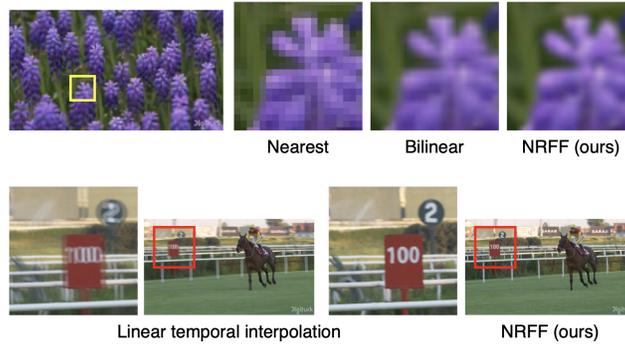


Fig. 9: The spatial and temporal interpolation. The first row shows the results of each method’s spatial super resolution, upscaling from (480, 270) to (1920, 1080). The second row shows the the results of temporal interpolation, which samples an intermediate frame that was not seen during video encoding.

through optical flows. The proposed approach, *Neural Residual Flow Fields (NRFF)* maintains smooth and less complex signals, which allows us to achieve more compact representations while maintaining quality.

Although the results were promising, there is still room for improvement. We observed that neural fields (or implicit neural representations) require a large number of parameters and long training iterations to capture high-frequency details. We believe that resolving this problem would significantly enhance encoding speed and make neural fields more accessible in various cases.

Note that we achieved fairly good compression rates without any model compression techniques, except for 16-bit precision weights. Incorporating unaccommodated network compression techniques into our proposed method would improve the performance much further. Weight pruning, entropy coding, and knowledge distillation could all be promising directions to investigate.

There are a handful of promising research directions that can further make this emerging representation more attractive. We believe we have only scratched the surface. A better understanding of the efficiency of neural network parameters in general might help answer a fundamental question about how a neural network preserves information in its parameters.

Acknowledgements This research was supported by the Ministry of Science and ICT (MSIT) of Korea, under the National Research Foundation (NRF) grant (2021R1F1A1061259, 2022R1F1A1064184), Institute of Information and Communication Technology Planning Evaluation (IITP) grants for the AI Graduate School program (IITP- 2019-0-00421), the ITRC (Information Technology Research Center) support program (IITP-2021-0-02052), the ICT Creative Consilience program (IITP-2020-0-01821), and the Artificial Intelligence Innovation Hub program (IITP-2021-0-02068).

References

1. Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., eds.: *Advances in Neural Information Processing Systems*. Volume 33., Curran Associates, Inc. (2020) 7537–7547
2. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.F., Lin, H., eds.: *Advances in Neural Information Processing Systems*. Volume 33., Curran Associates, Inc. (2020) 7462–7473
3. Tancik, M., Mildenhall, B., Wang, T., Schmidt, D., Srinivasan, P.P., Barron, J.T., Ng, R.: Learned initializations for optimizing coordinate-based neural representations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2021)
4. Martel, J.N.P., Lindell, D.B., Lin, C.Z., Chan, E.R., Monteiro, M., Wetzstein, G.: Acorn: Adaptive coordinate networks for neural scene representation. *ACM Trans. Graph.* **40** (2021)
5. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. (2020)
6. Cho, J., Nam, S., Rho, D., Ko, J.H., Park, E.: Streamable neural fields. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. (2022)
7. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2019)
8. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* **378** (2019) 686–707
9. Skorokhodov, I., Ignatyev, S., Elhoseiny, M.: Adversarial generation of continuous images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2021)
10. Chen, Y., Liu, S., Wang, X.: Learning continuous image representation with local implicit image function. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2021)
11. Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A.: Overview of the h.264/avc video coding standard. *IEEE Transactions on Circuits and Systems for Video Technology* **13** (2003) 560–576
12. Le Gall, D.: Mpeg: A video compression standard for multimedia applications. *Communications of the ACM* **34** (1991) 46–58
13. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on Circuits and Systems for Video Technology* **22** (2012) 1649–1668
14. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.: A naturalistic open source movie for optical flow evaluation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. (2012)
15. Mercat, A., Viitanen, M., Vanne, J.: Uvg dataset: 50/120fps 4k sequences for video codec analysis and development. In: *Proceedings of the 11th ACM Multimedia*

- Systems Conference. MMSys '20, New York, NY, USA, Association for Computing Machinery (2020) 297–302
16. Xie, Y., Takikawa, T., Saito, S., Litany, O., Yan, S., Khan, N., Tombari, F., Tompkin, J., Sitzmann, V., Sridhar, S.: Neural fields in visual computing and beyond. *Computer Graphics Forum* (2022)
 17. Hertz, A., Perel, O., Giryas, R., Sorkine-Hornung, O., Cohen-Or, D.: Sape: Spatially-adaptive progressive encoding for neural optimization. In: *Advances in Neural Information Processing Systems*. (2021)
 18. Atzmon, M., Lipman, Y.: Sal: Sign agnostic learning of shapes from raw data. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2020)
 19. Martin-Brualla, R., Radwan, N., Sajjadi, M.S.M., Barron, J.T., Dosovitskiy, A., Duckworth, D.: Nerf in the wild: Neural radiance fields for unconstrained photo collections. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2021)
 20. Zhang, K., Riegler, G., Snavely, N., Koltun, V.: Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492* (2020)
 21. Yariv, L., Kasten, Y., Moran, D., Galun, M., Atzmon, M., Ronen, B., Lipman, Y.: Multiview neural surface reconstruction by disentangling geometry and appearance. In: *Advances in Neural Information Processing Systems*. (2020)
 22. Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2021)
 23. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (2021)
 24. Li, Z., Niklaus, S., Snavely, N., Wang, O.: 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 (CVPR)*. (2021)
 25. Gao, C., Saraf, A., Kopf, J., Huang, J.B.: Dynamic view synthesis from dynamic monocular video. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. (2021)
 26. Chen, H., He, B., Wang, H., Ren, Y., Lim, S.N., Shrivastava, A.: Nerv: Neural representations for videos. In: *Advances in Neural Information Processing Systems*. (2021)
 27. Peng, S., Zhang, Y., Xu, Y., Wang, Q., Shuai, Q., Bao, H., Zhou, X.: Neural body: Implicit neural representations with structured latent codes for novel view synthesis of dynamic humans. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2021)
 28. Chen, Z., Chen, Y., Liu, J., Xu, X., Goel, V., Wang, Z., Shi, H., Wang, X.: Videoinr: Learning video implicit neural representation for continuous space-time super-resolution. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2022)
 29. Dupont, E., Golinski, A., Alizadeh, M., Teh, Y.W., Doucet, A.: COIN: COMpression with implicit neural representations. In: *Neural Compression: From Information Theory to Applications – Workshop @ ICLR 2021*. (2021)
 30. Strümler, Y., Postels, J., Yang, R., Van Gool, L., Tombari, F.: Implicit neural representations for image compression. *arXiv preprint arXiv:2112.04267* (2021)
 31. Dupont, E., Loya, H., Alizadeh, M., Goliński, A., Teh, Y.W., Doucet, A.: Coin++: Data agnostic neural compression. *arXiv preprint arXiv:2201.12904* (2022)

32. Zhang, Y., van Rozendaal, T., Brehmer, J., Nagel, M., Cohen, T.: Implicit neural video compression. In: ICLR Workshop on Deep Generative Models for Highly Structured Data. (2022)
33. Rippel, O., Nair, S., Lew, C., Branson, S., Anderson, A.G., Bourdev, L.: Learned video compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). (2019)
34. Lu, G., Ouyang, W., Xu, D., Zhang, X., Cai, C., Gao, Z.: Dvc: An end-to-end deep video compression framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
35. Horn, B.K., Schunck, B.G.: Determining optical flow. *Artificial intelligence* **17** (1981) 185–203
36. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., Brox, T.: FlowNet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV). (2015)
37. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: FlowNet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2017)
38. Hur, J., Roth, S.: Iterative residual refinement for joint optical flow and occlusion estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
39. Zhao, S., Sheng, Y., Dong, Y., Chang, E.I.C., Xu, Y.: MaskFlowNet: Asymmetric feature matching with learnable occlusion mask. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
40. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Proceedings of the European Conference on Computer Vision (ECCV). (2020)
41. Jiang, S., Campbell, D., Lu, Y., Li, H., Hartley, R.: Learning to estimate hidden motions with global motion aggregation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). (2021)
42. Rahaman, N., Baratin, A., Arpit, D., Dräxler, F., Lin, M., Hamprecht, F.A., Bengio, Y., Courville, A.C.: On the spectral bias of neural networks. In: International Conference on Machine Learning (ICML). (2019) 5301–5310