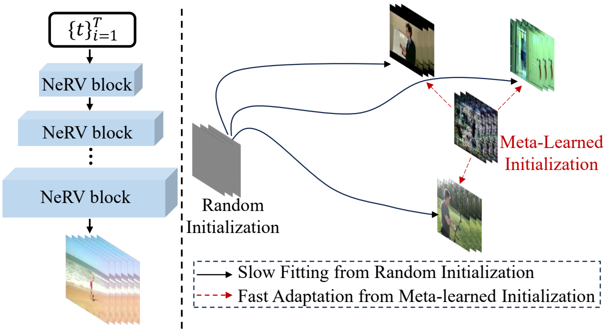
MetaNeRV: Meta Neural Representations for Videos with Spatial-Temporal Guidance

Jialong Guo[1*], **Ke Liu**[1*], **Jiangchao Yao**[2], **Zhihua Wang**[1], **Jiajun Bu**[1], **Haishuai Wang**[1†]
[1]Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems,
College of Computer Science, Zhejiang University, China
[2]Cooperative Medianet Innovation Center, Shanghai Jiaotong University, China
{jialongguo, keliu99, zhihua_wang, bjj, haishuai.wang}@zju.edu.cn, Sunarker@sjtu.edu.cn

Abstract

Neural Representations for Videos (NeRV) has emerged as a promising implicit neural representation (INR) approach for video analysis, which represents videos as neural networks with frame indexes as inputs. However, NeRV-based methods are time-consuming when adapting to a large number of diverse videos, as each video requires a separate NeRV model to be trained from scratch. In addition, NeRV-based methods spatially require generating a high-dimension signal (i.e., an entire image) from the input of a low-dimension timestamp, and a video typically consists of tens of frames temporally that have a minor change between adjacent frames. To improve the efficiency of video representation, we propose Meta Neural Representations for Videos, named MetaNeRV, a novel framework for fast NeRV representation for unseen videos. MetaNeRV leverages a meta-learning framework to learn an optimal parameter initialization, which serves as a good starting point for adapting to new videos. To address the unique spatial and temporal characteristics of video modality, we further introduce spatial-temporal guidance to improve the representation capabilities of MetaNeRV. Specifically, the spatial guidance with a multi-resolution loss aims to capture the information from different resolution stages, and the temporal guidance with an effective progressive learning strategy could gradually refine the number of fitted frames during the meta-learning process. Extensive experiments conducted on multiple datasets demonstrate the superiority of MetaNeRV for video representations and video compression.

code — https://github.com/jialong2023/MetaNeRV



(a) NeRV Network (b) Gradient Descent When Fitting New Videos

Figure 1: (a) NeRV Network takes the frame index as input and outputs an image of that index. Querying a sequence of frame indexes results in a list of sequences, which can represent a video. (b) The network with random initialization necessitates optimization through numerous steps for new videos, whereas meta-learned initialization enables swift adaptation to new videos.

Introduction

In recent years, Implicit Neural Representations (INR) have emerged as a powerful tool for continuously representing data in various computer vision tasks. The core idea of INR is to represent a signal as a function that can be effectively approximated by a neural network (Pinkus 1999). This network encodes the signal's values implicitly within its structure and parameters during the training or fitting process, allowing for the subsequent retrieval of these values through corresponding coordinates. Notably, INR has recently gained significant attention in the context of neural representations for videos, with notable examples including NeRV (Chen et al. 2021) and E-NeRV (Li et al. 2022). In contrast to traditional coordinate-based neural representations, NeRV-based approaches take the frame index as input and directly generate the desired frame image, which is called image-wise implicit neural representations 1, resulting in significantly faster training and inference speeds compared to their coordinate-based counterparts.

Although existing NeRV-based methods demonstrate impressive capabilities, their limitation to encoding a single video at a time restricts their applicability in real-world scenarios, as each new video typically requires optimization through numerous gradient descent steps.

D-NeRV (He et al. 2023) memorize keyframes of videos within the training set, aiming to enhance generalization by reconstructing transition frames from these keyframes. Nevertheless, such methodologies are constrained to representing frames within the trained videos, potentially hindering

---

[*] These authors contributed equally.

[†] Corresponding author: Haishuai Wang

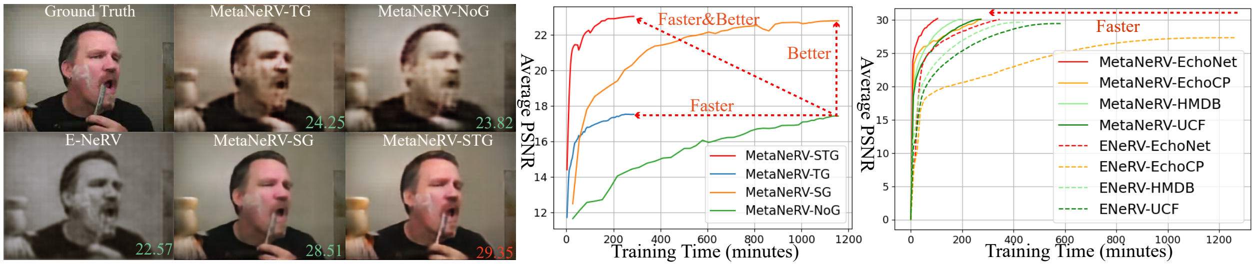


Figure 2: (a) The visualization results with three-step inference in video representation tasks between the E-NeRV method and our method of the different guidance, where TG, SG, NoG, and STG respectively represent Temporal Guidance, Spatial Guidance, No Guidance, and Spatio-Temporal Guidance. (b) The average PSNR and training time curves on the UCF dataset under different guidance, where the model trains faster and performs better under spatio-temporal guidance. (c) The average PSNR and training time curves of E-NeRV and our method on four datasets, given a target PSNR value of 30.

their ability to generalize to previously unseen videos.

To accelerate the adaptation of NeRV-based models on unseen videos, we present **MetaNeRV**, a meta-learning framework designed to learn optimal initial weights for neural representations of videos. Compared to traditional random initialization techniques, MetaNeRV learns effective initialization weights across a series of videos, acting as a powerful prior that accelerates convergence during optimization and enhances generalization capabilities for unseen videos. Our methodology employs the optimization-based meta-learning algorithm MAML (Finn, Abbeel, and Levine 2017), using a diverse meta-training set of videos to generate initial weight configurations ideally suited for representing unseen videos, leading to faster convergence and better generalization. As depicted in Fig. 1(b), each new video necessitates its optimization process, which can be less efficient. However, with proper initialization, the number of iteration steps can be significantly reduced.

Relying solely on meta-learning may yield poor performance. To address this, we propose two enhancements: Spatial Guidance: We introduce a multi-resolution loss function and add header modules to each NeRV block layer. These modules output video frames at different resolutions, improving the meta-learning framework's representation. Temporal Guidance: To handle convergence issues and inefficiency with complex videos, we adopt a progressive training strategy. During meta-learning, we gradually increase subtask difficulty, allowing the framework to smoothly transition from easier to more complex tasks.

Experimental results on multiple datasets demonstrate that MetaNeRV outperforms other frame-wise methods in both video representations. Additionally, we explore various applications of our method, including video compression and video denoising tasks. With quantization-aware training and entropy coding, MetaNeRV outperforms widely-used video codecs such as H.264 (Wiegand et al. 2003) and HEVC (Sullivan et al. 2012) and performs comparably with state-of-the-art video compression algorithms.

The contributions of this work are summarized as follows:

- We introduce **MetaNeRV**, a meta-learning-driven framework with spatial-temporal guidance, tailored for NeRV-based video reconstruction. MetaNeRV enhances its performance by optimizing the initialization parameters.
- We propose a progressive training strategy as temporal guidance and incorporate a multi-resolution loss as spatial guidance within the meta-learning framework, providing precise supervision for better representation capabilities and improving training efficiency.
- Comprehensive experiments on various video datasets show that: (i) optimized initialization parameters lead to a significant acceleration in the convergence of the NeRV-based model, achieving a remarkable **9x** increase in speed; (ii) the incorporation of our proposed guidance enhances the training efficacy of the meta-learning framework, e.g. resulting in a notable improvement of **+16** PSNR in a single step on the EchoNet-LVH dataset; (iii) excellent performance in several video-related applications, including video compression and denoising.

Related Work

Implicit neural representations. Neural networks can be used to approximate the functions that map the input coordinates to various types of signals. It has brought great interest and has been widely adopted to represent 3D shape (Sitzmann, Zollhöfer, and Wetzstein 2019; Mescheder et al. 2019; Park et al. 2019; Liu et al. 2023b), novel view synthesis (Mildenhall et al. 2021; Yu et al. 2021) and so on. These approaches train a neural network to fit a single scene or object such that the network weights encode it. Implicit neural representations have also been applied to represent signals (Hinton, Osindero, and Teh 2006; Vaswani et al. 2017; Tancik et al. 2020; Sitzmann et al. 2020b; Peng, Zeng, and Zhao 2022a,b; Zhang et al. 2025), images (Wang, Simoncelli, and Bovik 2003; Hsu et al. 2019; Chen, Liu, and Wang 2021; Dupont et al. 2021; Chen et al. 2024; Liu et al. 2023a), videos (Chen et al. 2021; Lai 2021; Rho et al. 2022; He 2022; Tong 2022), and time series (Li et al. 2024).

Image-wise implicit neural representations. The first image-wise implicit neural representation for videos is proposed by NeRV (Chen et al. 2021), which takes the frame index and outputs the corresponding RGB frame. Compared to the pixel-wise implicit neural representation (Sitzmann et al. 2020b), NeRV improves the encoding and decoding speed

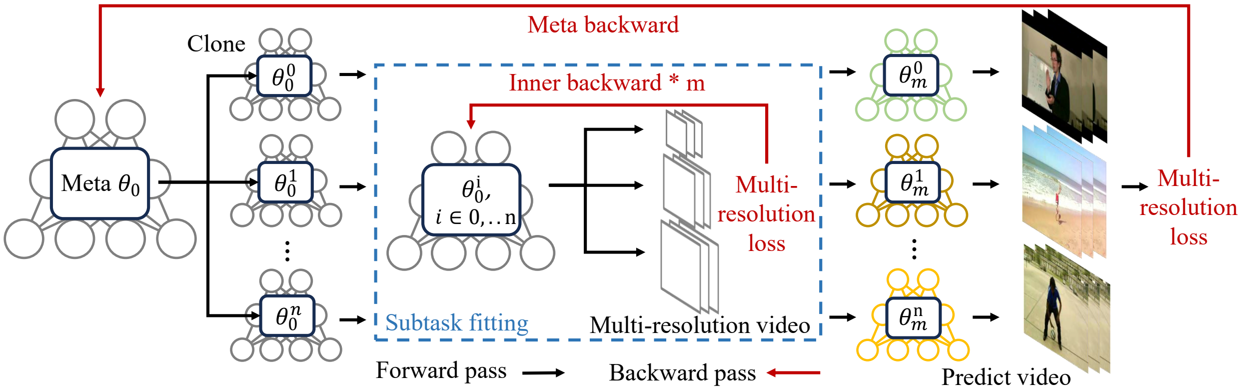


Figure 3: Framework for MetaNeRV. A meta-learner is utilized to sample tasks of learning video and learns an initialized weight that can quickly fine-tune to a new video. The initialized weights will be cloned and then optimized m steps for n subtask in their corresponding video.

greatly and achieves better video reconstruction quality. Based on NeRV, E-NeRV (Li et al. 2022) boosts the video reconstruction performance via decomposing the image-wise implicit neural representation into separate spatial and temporal contexts. CNeRV (Chen et al. 2022) proposes a hybrid video neural representation with content-adaptive embedding to introduce internal generalization further. NRFF (He et al. 2023) introduces a visual content encoder to encode the clip-specific visual content from the sampled key-frames and a motion-aware decoder to output video frames. FFNeRV (Lee et al. 2023) introduces the multi-resolution temporal grids to combine different temporal resolutions. HNeRV (Chen et al. 2023) and HiVeRV (Kwan et al. 2024) proposed a hybrid neural representation approach, employing a VAE-shaped deep network to address these concerns.

Video compresion Visual data compression, a cornerstone of computer vision and image processing, has been extensively studied over several decades. Traditional video compression algorithms like H.264 (Wiegand et al. 2003), and HEVC (Sullivan et al. 2012) have achieved remarkable success. Some works have approached video compression as an image interpolation problem, introducing competitive interpolation networks (Wu, Singhal, and Krahenbuhl 2018), generalized optical flow to scale-space flow for enhanced uncertainty modeling (Agustsson et al. 2020; Yang et al. 2020b), and employed temporal hierarchical structures with neural networks for various components (Yang et al. 2020a). However, these methods are still constrained by the traditional compression pipeline. Alternatively, NeRV adopts the INR method, transforming video compression into model compression and demonstrating substantial potential. Given that videos are typically encoded once but decoded multiple times, INR methods like NeRV excel due to their high decoding efficiency and facilitate parallel decoding, contrasting with sequential decoding requirements in other video compression methods post key frame reconstruction.

Meta-learning INRs. Meta-learning typically addresses the problem of "few-shot learning", where some example tasks are used to train an algorithm that has a great generalization ability on new similar tasks. Some previous works on meta-learning have focused on few-shot learning (Ravi and Larochelle 2016; Mishra et al. 2017; Patravali 2021; Liu et al. 2023a) and reinforcement learning (Finn, Abbeel, and Levine 2017; Sitzmann et al. 2020a), where a meta-learner allows fast adaptation for new observations and better generalization with few samples. Optimization-based meta-learning algorithms such as Model-Agnostic Meta-Learning (MAML) (Finn, Abbeel, and Levine 2017; Li et al. 2017; Antoniou, Edwards, and Storkey 2018; Flennerhag et al. 2019; Rajeswaran et al. 2019; Hospedales et al. 2021; Tancik et al. 2021; Guo et al. 2024; Liu et al. 2023a) are relevant to this work. Given a network architecture for performing a task, these methods use an outer loop of gradient-based learning to find a weight initialization that allows the network to optimize more efficiently for new tasks at test time. These methods assume the use of a standard gradient-based optimization method such as stochastic gradient descent or Adam (Kingma and Ba 2014) at test time. Liu et al. (Liu et al. 2023a) propose partition methods for learning-to-learn INRs by meta-learning.

We are the first to utilize the meta-learning framework for image-wise implicit neural representation models, resulting in increased convergence speed and enhanced generalizability of video implicit neural representation methods. As illustrated in Fig. 2, our method has shown significant performance in quantitative and qualitative experiments.

Methods

Problem Formulation

We aim to learn a prior over INR for video. As in (Chen et al. 2021; Li et al. 2022), the video can be viewed as a continuous function $f : T \to F$ defined in a bounded domain that $T \in \mathbb{R}, F \in \mathbb{R}^{H\times W\times 3}$. We define a video $V = \{F_n\}_{n=1}^{N}$, where F_n is the frame $\in \mathbb{R}^{H\times W\times 3}$ and N denotes totally N frames. The NeRV-based model fits the video via a deep neural network which is represented by a neural representation $f_\theta : \mathbb{R} \to \mathbb{R}^{H\times W\times 3}$, where the input is a frame index $t \in \mathbb{R}$ and the output is the corresponding RGB image $F_n \in \mathbb{R}^{H\times W\times 3}$. Therefore, video encoding is done by fitting a neural network f_θ to a given video. We present the details in Fig.1(a).

In NeRV-based models, the input usually consists of embedding vectors generated by frame index encoding. Some methods incorporate complementary information to generate these embedding vectors. We follow E-NeRV(Li et al. 2022) to integrate coordinate data to construct spatiotemporal embedding vectors. These embedding vectors are then fed into the generator, which is successively expanded by convolutional or anti-convolutional operations to predict the desired image size. The theoretical details concerning video fitting can be found in Appendix Section A.

MetaNeRV Framework

In this section, we provide a detailed introduction to our MetaNeRV framework, as illustrated in Fig. 3. Given a dataset of observations of videos $\{V\}$ from a specific distribution $\mathcal{D}$ (e.g., traditional videos or ultrasound videos), our objective is to find initial weights θ_0^* that minimize the final loss $L(\theta_m, V)$ when optimizing a network f_θ through m optimization steps to represent a new video from the same distribution. Our target function is formulated as follows:

$$\theta_0^* = \arg\min_{\theta_0} E_{V\sim\mathcal{D}}\left[L\left(\theta_m\left(\theta_0, V\right), V\right)\right]. \tag{1}$$

We utilize MAML (Finn, Abbeel, and Levine 2017; Liu et al. 2023a) to learn the initial weights of the network so that it can be a good starting point for gradient descent in the new tasks.

Given a video V, calculating the weight values $\theta_m(\theta_0, V)$ necessitates executing m optimization steps, collectively termed as the inner loop. We encapsulate this inner loop with an outer loop of meta-learning to ascertain the initial weights θ_0. In each iteration of the outer loop, we sample a video V_j from $\mathcal{D}$ and apply the update rule:

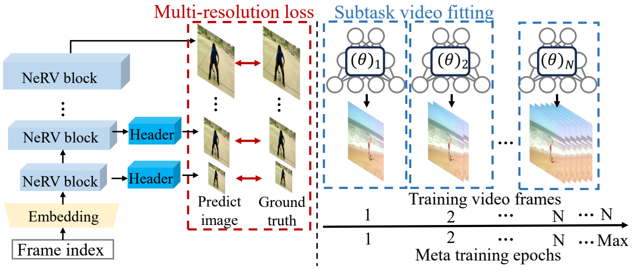
$$(\theta_0)_{j+1} = (\theta_0)_j - \eta\nabla_\theta L\left(\theta_m\left(\theta, V_j\right), V_j\right)\big|_{\theta=(\theta_0)_j}, \tag{2}$$

with meta-learning learning rate η. This update rule applies gradient descent to the loss on the weights $\theta_m\left(\theta, V_j\right)$ resulting from the inner loop optimization.

We adopt a combination of L1 and SSIM loss as our loss function for network optimization, which calculates the loss of overall pixel locations of the predicted image and the ground-truth image. Loss function details can be found in Appendix Section A.

Spatial Guidance

The spatial challenge arises because NeRV-based models progressively enlarge from a small vector by repeatedly passing through the same block. Directly applying the finest-grained supervision at the final resolution stages may result in insufficient supervisory signals for the preceding blocks, potentially making it difficult for the earlier resolution stages to converge to a better solution at that resolution. Therefore, we introduce multi-resolution supervision, which provides supervisory signals directly at different resolution stages, to encourage the output of all NeRV blocks to converge toward the unique ground truth. This spatial guidance enhances both the fitting accuracy and convergence speed.



(a) Spatial guidance (b) Temporal guidance

Figure 4: (a) NeRV network inputs a one-dimensional frame index, which expands through NeRV blocks to the image size, outputting corresponding frames. We propose adding a header block for spatial guidance at each NeRV block layer. (b) We propose a progressive training strategy for temporal guidance, gradually increasing video frame numbers in subtasks during meta-learning.

The NeRV-based model employs up-sample blocks to scale the encoding of the frame index into the image with an appropriate block-by-block size. These up-sample blocks comprise K feature layers, and for each of these layers, we append a convolutional header:

$$\{F_k'\}_K = \{\mathtt{header}_k(f_k)\}_K, \tag{3}$$

where the feature of each layer f_k is handled by the header layer $\mathtt{header}_k$ to generate a video frame F_k' corresponding to the size of the feature map as shown in Fig. 4(a).

This header transforms the multi-channel feature map into a three-channel feature map. We then downsample the ground truth image to align with the size of the feature map, ultimately computing the loss for gradient backpropagation:

$$L_{\text{multi}}(V_j, \{F_k'\}_K) = \sum_{k=1}^{K} L(F_k', \mathrm{Pooling(V_j)}), \tag{4}$$

where the final loss L_{multi} is calculated by computing a weighted sum based on the global average pooling of downsample frame X_j.

Temporal Guidance

In terms of time, a video typically consists of tens of frames, with minor changes between adjacent frames. For videos sharing similar backgrounds, utilizing spatial guidance during training can yield better results. However, when dealing with videos where background and foreground information exhibit significant differences, the training process encounters issues with low training efficiency.

To address this challenge and enhance the model's training efficiency, we introduce a progressive training strategy as temporal guidance. This method aims to assist the model in learning optimal initialization parameters from videos with distinct differences. As illustrated in Fig. 4(b), progressive training initiates the inner loop with a simple task: learning one frame per video. Subsequently, as iterations proceed, the number of frames in the videos is increased to elevate the task complexity gradually. The simple task enables the

Methods	PSNR ↑(Step1/Step3)					MSSSIM ↑(Step1/Step3)				
	MCL_JCV	HMDB-51	UCF101	EchoNet-LVH	EchoCP	MCL_JCV	HMDB-51	UCF101	EchoNet-LVH	EchoCP
NeRV	11.23/13.78	11.71/14.57	11.38/14.78	8.06/15.23	7.14/17.68	0.19/0.37	0.15/0.41	0.14/0.37	0.32/0.50	0.21/0.41
E-NeRV	11.13/15.78	11.13/15.78	10.04/15.04	6.36/16.44	7.79/18.53	0.24/0.61	0.24/0.61	0.25/0.59	0.38/0.71	0.21/0.74
FFNeRV	10.11/12.47	11.71/13.09	12.75/13.88	6.64/16.95	7.46/12.27	0.19/0.32	0.15/0.32	0.14/0.25	0.20/0.45	0.22/0.32
HNeRV	11.25/12.69	11.71/12.89	10.9/13.89	6.72/17.35	6.61/17.57	0.21/0.36	0.15/0.3	0.13/0.26	0.23/0.53	0.20/0.40
MetaNeRV	**17.60/22.02**	**18.43/21.43**	**18.69/22.46**	**24.05/26.94**	**23.34/25.44**	**0.67/0.83**	**0.73/0.88**	**0.72/0.86**	**0.88/0.94**	**0.89/0.93**

Table 1: The quantitative results of one-step and three-step inference for each method in five datasets.

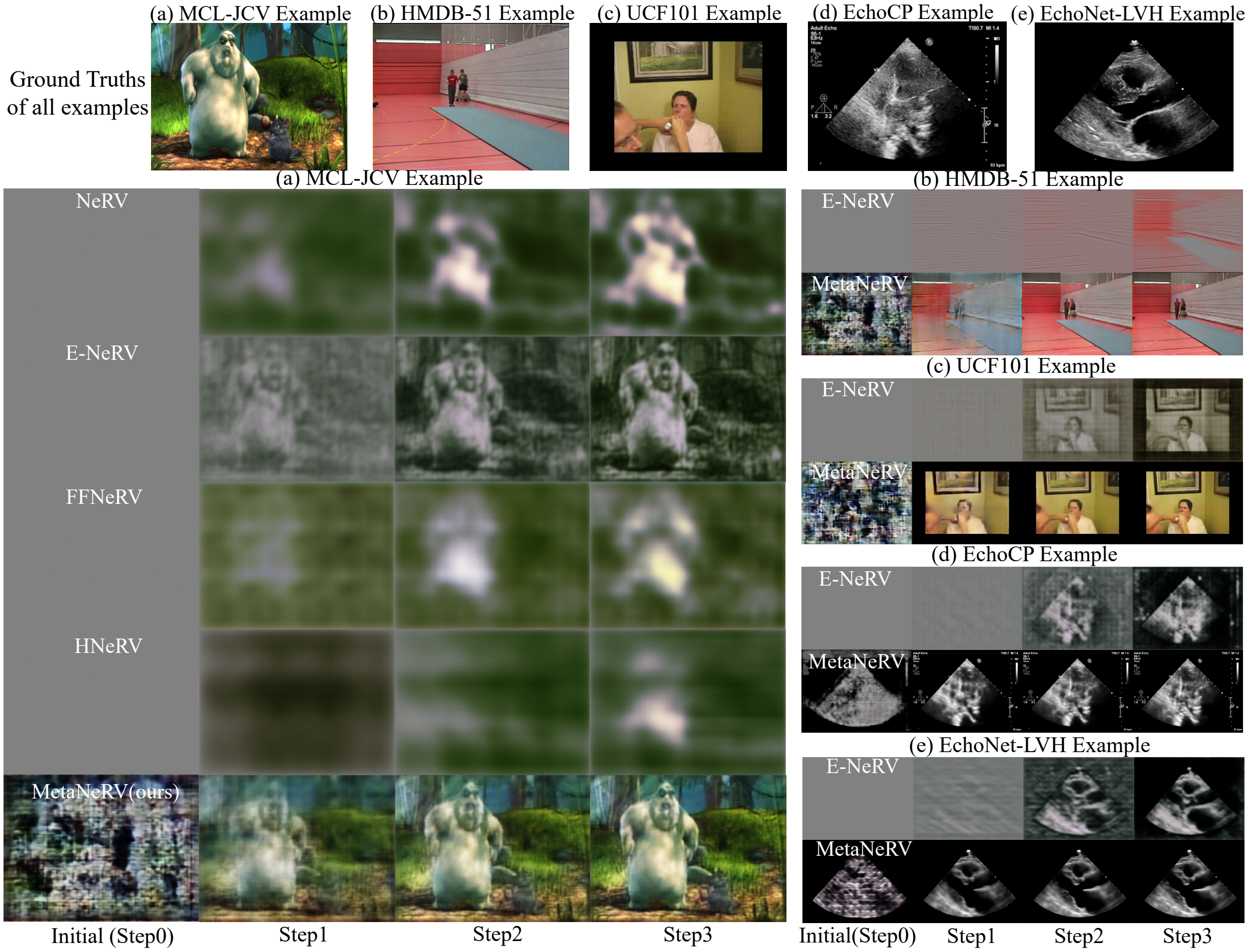


Figure 5: The visualization of NeRV, E-NeRV, FFNeRV, HNeRV, and MetaNeRV fitting the MCL_JCV, HMDB-51, UCF101, EchoCP, and EchoNet-LVH examples. Notably, our method produces remarkable results in merely 3 iteration steps. "step 0" represents inference results directly from the initialization weight without further training.

model to converge more rapidly, and by reducing the number of frames in the task, it also significantly decreases training time. Further details regarding the algorithm can be found in Appendix Section B.

Experiments

Datasets and Implementation Details

Dataset We conduct quantitative and qualitative comparison experiments on 8 different video datasets to evaluate our MetaNeRV against NeRV-based methods for video representation tasks. The datasets include multiple real-world datasets across various video types, such as UCF101(Soomro, Zamir, and Shah 2012), HMDB-51(Kuehne et al. 2011), and MCL_JCV (Wang et al. 2016), as well as ultrasound datasets like EchoCP (Wang et al. 2021), and EchoNet-LVH (Duffy et al. 2022). We selected 900 videos for each dataset, 800 for the training set, and 100 for the test set. Each video sequence contains 60 frames, processed to a resolution of 320×240.

Furthermore, we conduct inference experiments on HOLLYWOOD2 (Marcin 2009), SVW (Safdarnejad et al. 2015), and OOPS (Epstein, Chen, and Vondrick 2020), which are diverse datasets for action recognition, encompassing movie scenes, amateur sports, and unintentional human activities,

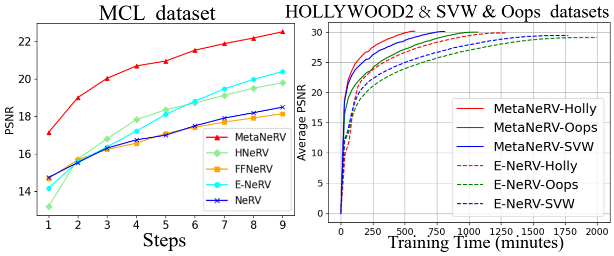


Figure 6: (left) Comparison of baselines, and our MetaNeRV on MCL_JCV dataset. Our method's performance on 1 step is better than FFNeRV's at 9 steps, which shows better performance and faster convergence. (right) Comparison of MetaNeRV and E-NeRV on HOLLYWOOD2, SVW, and Oops datasets. Given a training objective of achieving an average PSNR of 30 for the dataset, our model significantly reduces training time.

with 300 videos in each dataset. A description of more datasets can be found in Appendix Section C.

Implementation We set up-scale factors $5, 2, 2, 2, 2$ for each block of our MetaNeRV model to reconstruct a 320×240 image from the feature map of size 4×3. For a fair comparison, we follow the training schedule of the original E-NeRV implementation. We train the model using Adam optimizer (Kingma and Ba 2014) with a learning rate 1e-4 by Pytorch. We conduct all experiments with RTXA6000 GPU, while the number of inner loop steps is 3.

Metrics For evaluation metrics, we use PSNR and MS-SSIM (Wang, Simoncelli, and Bovik 2003) to evaluate reconstruction quality. Bits-per-pixel (BPP) is adopted to evaluate the image compression performance.

Main Results

Video representation. Initially, we compare our method with image-wise INR methods. Our model has trained optimal initialization weights separately on real-world datasets (HMDB-51, UCF101) and ultrasound datasets (EchoCP, EchoNet-LVH). Due to the limited number of videos in the MCL_JCV dataset, which hinders the optimization of better initialization weights, we directly infer on MCL_JCV using weights trained on HMDB-51.

Qualitative comparison. Our method can swiftly adapt to the content of new videos, even with just a few iteration steps in the video representation task, as shown in Fig. 5. We observe that visualizations of initial weights on real-world datasets appear more reasonable than visualizations from randomly initialized weights, while those on ultrasound datasets exhibit more dataset-specific characteristics, visually demonstrating that our method has learned an optimal initialization weight.

Quantitative comparison. Our method significantly all other outperforms image-wise methods under fewer iteration steps. as presented in Tab. 1. Notably, on ultrasound datasets, our method's PSNR and MS-SSIM metrics in one-step iteration exceed those of others by 300%. On real-world datasets, our method significantly outperforms other methods in both one-step and three-step iterations.

OOD results. To demonstrate the generalization and efficiency of our method, we conducted extensive experiments on out-of-distribution (OOD) datasets. All experiments in Fig. 6 used weights trained on HMDB-51 and directly inferred on four OOD datasets. The left side of Fig. 6 showcases our method's excellent performance on MCL_JCV, while the right side illustrates that, given a target PSNR value for training, our method can significantly reduce video representation time and improve efficiency on three datasets with a larger number of videos.

Video Compression and Denoising. We further evaluate MetaNeRV's versatility with two downstream tasks: 1) video denoising on the Bunny data, and 2) video compression on the UVG dataset. Adhered to NeRV's setting, we apply videos with noise as training data, and compare the prediction results with the real videos. We also apply an additional neural network parameter pruning with various prune ratios for different NeRV-based methods to evaluate the video compression performance. In addition, we compare the compression ability of our methods with lots of popular methods, including H.264(Wiegand et al. 2003), HEVC(Sullivan et al. 2012), HLVC(Yang et al. 2020a), Scale-space(Agustsson et al. 2020), Wu et al.(Wu, Singhal, and Krahenbuhl 2018), NeRV(Chen et al. 2021), and PS-NeRV(Bai et al. 2023).

Stronger denoising ability. As shown in Fig. 7(a), we observe that the denoising result from MetaNeRV achieves better visualization performance and higher PSNR than the denoising result from NeRV, which demonstrates the strong denoising ability of MetaNeRV.

Better performance for network pruning. As depicted in Fig. 7(b), MetaNeRV achieves better reconstruction PSNR at all different sparsity (pruning with different parameter ratios) than NeRV and E-NeRV, highlighting its robust network pruning ability for video compression.

More powerful compression ability. We present the rate-distortion curves in Fig.7(c). We find that MetaNeRV surpasses all image-wise NeRV-based approaches. Furthermore, MetaNeRV outperforms traditional video compression technologies and other learning-based video compression methods at most BPPs.

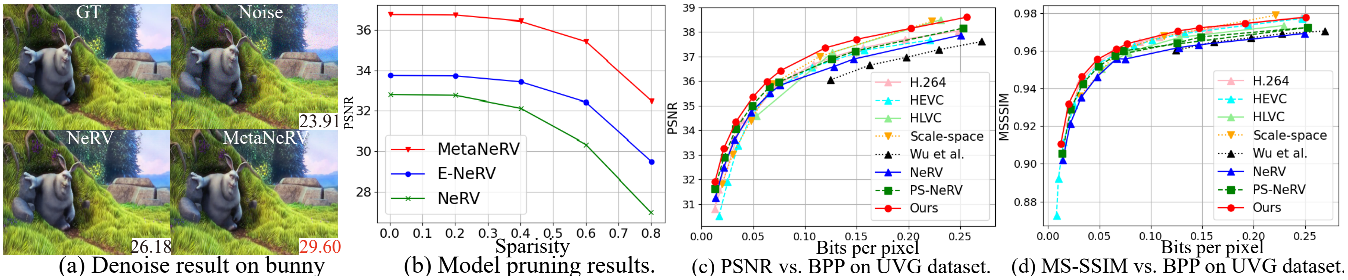
Ablation Studies

We also conducted ablation studies on four datasets to verify the effect of different guidance. More qualitative and quantitative results can be found in Appendix Section D.

Meta-learning. The meta-learning framework accelerates the video representation of four datasets by gradually adding both guidances across all experiment video datasets, as presented in Tab.2.

Spatial guidance. The spatial guidance significantly enhances the model's performance as shown in Fig.8 (a) of training curves. Furthermore, during the inference stage in Fig.8 (c), the method with added spatial guidance achieves a higher PSNR under the same number of iterations, indicating its effectiveness in improving the model's performance.

Temporal guidance. The temporal guidance is effective in reducing the model's training time while also achieving a



(a) Denoise result on bunny (b) Model pruning results. (c) PSNR vs. BPP on UVG dataset. (d) MS-SSIM vs. BPP on UVG dataset.

Figure 7: (a) Better denoise result on Bunny data. “Noise” denotes the noisy frames before any denoising process. (b) MetaNeRV outperforms other methods in pruning. (c)(d) MetaNeRV shows better video compression results on the UVG dataset.

Methods	Meta-learning	Temporal Guidance	Spatial Guidance	PSNR ↑(Step1/Step3)				MSSSIM ↑(Step1/Step3)			
				HMDB-51	UCF101	EchoNet-LVH	EchoCP	HMDB-51	UCF101	EchoNet-LVH	EchoCP
E-NeRV	×	×	×	11.13/15.78	10.04/15.04	6.36/16.44	7.79/18.53	0.24/0.61	0.25/0.59	0.38/0.71	0.21/0.74
MetaNeRV-NoG	✓	×	×	16.89/19.43	16.4/19.08	21.66/23.72	19.77/21.78	0.57/0.71	0.55/0.7	0.81/0.87	0.79/0.87
MetaNeRV-TG	✓	✓	×	17.3/19.31	16.83/19.53	22.16/24.31	20.63/23.15	0.63/0.76	0.59/0.73	0.83/0.88	0.81/0.89
MetaNeRV-SG	✓	×	✓	17.41/22.06	18.04/21.96	23.19/25.63	22.03/23.77	0.69/0.85	0.68/0.83	0.85/0.91	0.88/0.92
MetaNeRV-STG	✓	✓	✓	**18.43/21.43**	**18.69/22.46**	**24.05/26.94**	**23.34/25.44**	**0.73/0.88**	**0.72/0.86**	**0.88/0.94**	**0.89/0.93**

Table 2: The ablation quantitative results of one-step and three-step inference for each method in four datasets.

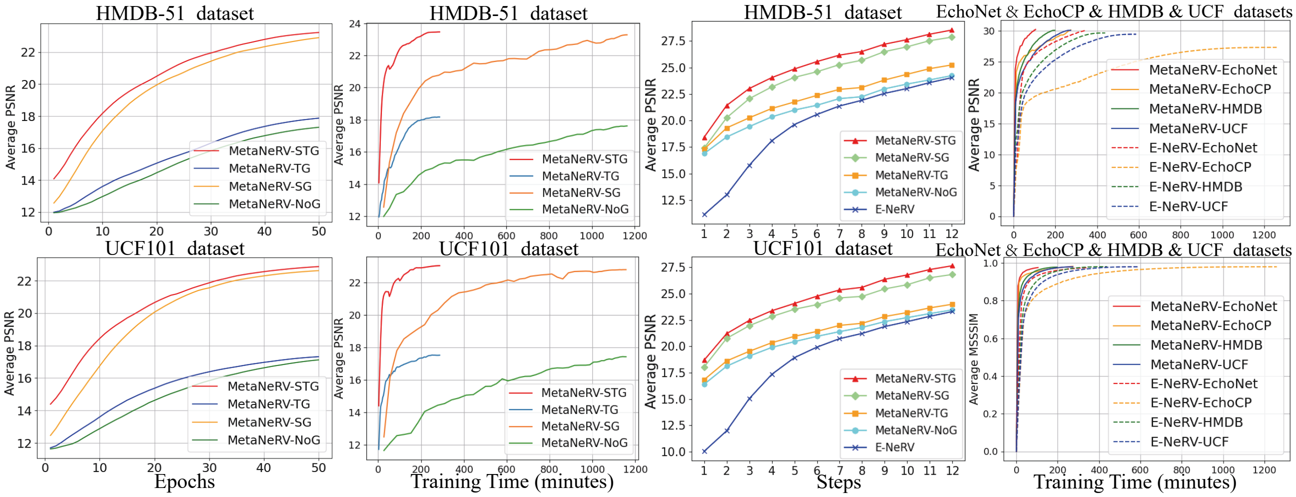


Figure 8: (a) Training curves of Epochs vs PSNR with different guidance, demonstrating improved model performance with spatial guidance. (b) Training curves of Time vs PSNR with different guidance, proving reduced training time and enhanced efficiency with temporal guidance. (c) Inference performance of our method variants, all surpassing the baseline. (d) Our method significantly reduces representation time given a 30 PSNR target.

slight performance improvement, as shown in Fig.8 of training curves. During the inference stage, the model exhibits good performance, indicating that our proposed temporal guidance enhances training efficiency without compromising the model’s overall performance.

Significant results. Remarkably, our proposed method shows outstanding results after one iteration, surpassing the baseline by over 16 PSNR and 3x MS-SSIM on EchoNet-LVH, and consistently exceeding by at least 4 PSNR across other datasets. Fig.8 (d) illustrates that our proposed method significantly enhances the efficiency of video representation under the given training objectives.

Conclusion

In conclusion, we present MetaNeRV, a sophisticated meta-learning framework designed to optimize the initialization process of NeRV models. By learning optimal initialization parameters, we have achieved significant improvements in both the performance and efficiency of video reconstruction tasks. We introduce spatial guidance for precise training supervision and a temporal guidance regimen to enhance training efficiency while maintaining stability. The model has limitations, it cannot represent video for higher resolutions without retraining and may face convergence challenges with limited video training data. Future work will focus on addressing these constraints.

Acknowledgements

This work is supported by the National Key R&D Program of China (Grant No. 2022ZD0160703), the National Natural Science Foundation of China (Grant Nos. 62202422, 62406279, and 62071330), and the Shanghai Artificial Intelligence Laboratory.

References

Agustsson, E.; Minnen, D.; Johnston, N.; Balle, J.; Hwang, S. J.; and Toderici, G. 2020. Scale-space flow for end-to-end optimized video compression. In *CVPR*, 8503–8512.

Antoniou, A.; Edwards, H.; and Storkey, A. 2018. How to train your MAML. *arXiv preprint arXiv:1810.09502*.

Bai, Y.; Dong, C.; Wang, C.; and Yuan, C. 2023. Ps-nerv: Patch-wise stylized neural representations for videos. In *ICIP*, 41–45. IEEE.

Chen, H.; Gwilliam, M.; He, B.; Lim, S.-N.; and Shrivastava, A. 2022. CNeRV: Content-adaptive neural representation for visual data. *arXiv preprint arXiv:2211.10421*.

Chen, H.; Gwilliam, M.; Lim, S.-N.; and Shrivastava, A. 2023. Hnerv: A hybrid neural representation for videos. In *CVPR*, 10270–10279.

Chen, H.; He, B.; Wang, H.; Ren, Y.; Lim, S. N.; and Shrivastava, A. 2021. Nerv: Neural representations for videos. *NeurIPS*, 34: 21557–21568.

Chen, X.; Ding, M.; Wang, X.; Xin, Y.; Mo, S.; Wang, Y.; Han, S.; Luo, P.; Zeng, G.; and Wang, J. 2024. Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision*, 132(1): 208–223.

Chen, Y.; Liu, S.; and Wang, X. 2021. Learning continuous image representation with local implicit image function. In *CVPR*, 8628–8638.

Duffy, G.; Cheng, P. P.; Yuan, N.; He, B.; Kwan, A. C.; Shun-Shin, M. J.; Alexander, K. M.; Ebinger, J.; Lungren, M. P.; Rader, F.; et al. 2022. High-throughput precision phenotyping of left ventricular hypertrophy with cardiovascular deep learning. *JAMA cardiology*, 7(4): 386–395.

Dupont, E.; Goliński, A.; Alizadeh, M.; Teh, Y. W.; and Doucet, A. 2021. Coin: Compression with implicit neural representations. *arXiv preprint arXiv:2103.03123*.

Epstein, D.; Chen, B.; and Vondrick, C. 2020. Oops! predicting unintentional action in video. In *CVPR*, 919–929.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Flennerhag, S.; Rusu, A. A.; Pascanu, R.; Visin, F.; Yin, H.; and Hadsell, R. 2019. Meta-learning with warped gradient descent. *arXiv preprint arXiv:1909.00025*.

Guo, H.; Ba, Y.; Hu, J.; Si, L.; Qiang, W.; and Shi, L. 2024. Self-Supervised Representation Learning with Meta Comprehensive Regularization. In *AAAI*, volume 38, 1959–1967.

He, B.; Yang, X.; Wang, H.; Wu, Z.; Chen, H.; Huang, S.; Ren, Y.; Lim, S.-N.; and Shrivastava, A. 2023. Towards scalable neural representation for diverse videos. In *CVPR*, 6132–6142.

He, K. 2022. Masked autoencoders are scalable vision learners. In *CVPR*, 16000–16009.

Hinton, G. E.; Osindero, S.; and Teh, Y.-W. 2006. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7): 1527–1554.

Hospedales, T.; Antoniou, A.; Micaelli, P.; and Storkey, A. 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169.

Hsu, Y.-C.; Lv, Z.; Schlosser, J.; Odom, P.; and Kira, Z. 2019. Multi-class classification without multi-class labels. *arXiv preprint arXiv:1901.00544*.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *2011 ICCV*, 2556–2563. IEEE.

Kwan, H. M.; Gao, G.; Zhang, F.; Gower, A.; and Bull, D. 2024. Hinerv: Video compression with hierarchical encoding-based neural representation. *NeurIPS*, 36.

Lai, Z. 2021. Video autoencoder: self-supervised disentanglement of static 3d structure and motion. In *ICCV*, 9730–9740.

Lee, J. C.; Rho, D.; Ko, J. H.; and Park, E. 2023. Ffnerv: Flow-guided frame-wise neural representations for videos. In *Proceedings of the 31st ACM International Conference on Multimedia*, 7859–7870.

Li, M.; Liu, K.; Chen, H.; Bu, J.; Wang, H.; and Wang, H. 2024. TSINR: Capturing Temporal Continuity via Implicit Neural Representations for Time Series Anomaly Detection. *arXiv preprint arXiv:2411.11641*.

Li, Z.; Wang, M.; Pi, H.; Xu, K.; Mei, J.; and Liu, Y. 2022. E-nerv: Expedite neural video representation with disentangled spatial-temporal context. In *ECCV*, 267–284. Springer.

Li, Z.; Zhou, F.; Chen, F.; and Li, H. 2017. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*.

Liu, K.; Liu, F.; Wang, H.; Ma, N.; Bu, J.; and Han, B. 2023a. Partition Speeds Up Learning Implicit Neural Representations Based on Exponential-Increase Hypothesis. In *ICCV*, 5474–5483.

Liu, K.; Ma, N.; Wang, Z.; Gu, J.; Bu, J.; and Wang, H. 2023b. Implicit Neural Distance Optimization for Mesh Neural Subdivision. In *2023 IEEE International Conference on Multimedia and Expo*, 2039–2044. IEEE.

Marcin. 2009. Actions in Context. In *CVPR*.

Mescheder, L.; Oechsle, M.; Niemeyer, M.; Nowozin, S.; and Geiger, A. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 4460–4470.

Mildenhall, B.; Srinivasan, P. P.; Tancik, M.; Barron, J. T.; Ramamoorthi, R.; and Ng, R. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1): 99–106.

Mishra, N.; Rohaninejad, M.; Chen, X.; and Abbeel, P. 2017. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*.

Park, J. J.; Florence, P.; Straub, J.; Newcombe, R.; and Lovegrove, S. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 165–174.

Patravali, J. 2021. Unsupervised few-shot action recognition via action-appearance aligned meta-adaptation. In *ICCV*, 8484–8494.

Peng, R.; Zeng, Y.; and Zhao, J. 2022a. Distill The Image to Nowhere: Inversion Knowledge Distillation for Multimodal Machine Translation. In *EMNLP*, 2379–2390.

Peng, R.; Zeng, Y.; and Zhao, J. 2022b. HybridVocab: Towards Multi-Modal Machine Translation via Multi-Aspect Alignment. In *ICMR*, 380–388.

Pinkus, A. 1999. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8: 143–195.

Rajeswaran, A.; Finn, C.; Kakade, S. M.; and Levine, S. 2019. Meta-learning with implicit gradients. *NeurIPS*, 32.

Ravi, S.; and Larochelle, H. 2016. Optimization as a model for few-shot learning. In *International conference on learning representations*.

Rho, D.; Cho, J.; Ko, J. H.; and Park, E. 2022. Neural residual flow fields for efficient video representations. In *Proceedings of the Asian Conference on Computer Vision*, 3447–3463.

Safdarnejad, S. M.; Liu, X.; Udpa, L.; Andrus, B.; Wood, J.; and Craven, D. 2015. Sports videos in the wild (svw): A video dataset for sports analysis. In *2015 11th IEEE international conference and workshops on automatic face and gesture recognition (FG)*, volume 1, 1–7. IEEE.

Sitzmann, V.; Chan, E.; Tucker, R.; Snavely, N.; and Wetzstein, G. 2020a. Metasdf: Meta-learning signed distance functions. *NeurIPS*, 33: 10136–10147.

Sitzmann, V.; Martel, J.; Bergman, A.; Lindell, D.; and Wetzstein, G. 2020b. Implicit neural representations with periodic activation functions. *NeurIPS*, 33: 7462–7473.

Sitzmann, V.; Zollhöfer, M.; and Wetzstein, G. 2019. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *NeurIPS*, 32.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sullivan, G. J.; Ohm, J.-R.; Han, W.-J.; and Wiegand, T. 2012. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12): 1649–1668.

Tancik, M.; Mildenhall, B.; Wang, T.; Schmidt, D.; Srinivasan, P. P.; Barron, J. T.; and Ng, R. 2021. Learned initializations for optimizing coordinate-based neural representations. In *CVPR*, 2846–2855.

Tancik, M.; Srinivasan, P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.; and Ng, R. 2020. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 33: 7537–7547.

Tong, Z. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NeurIPS*, 35: 10078–10093.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *NeurIPS*, 30.

Wang, H.; Gan, W.; Hu, S.; Lin, J. Y.; Jin, L.; Song, L.; Wang, P.; Katsavounidis, I.; Aaron, A.; and Kuo, C.-C. J. 2016. MCL-JCV: a JND-based H. 264/AVC video quality assessment dataset. In *ICIP*, 1509–1513. IEEE.

Wang, T.; Li, Z.; Huang, M.; Zhuang, J.; Bi, S.; Zhang, J.; Shi, Y.; Fei, H.; and Xu, X. 2021. EchoCP: An Echocardiography Dataset in Contrast Transthoracic Echocardiography for Patent Foramen Ovale Diagnosis. In *MICCAI*, 506–515. Springer.

Wang, Z.; Simoncelli, E. P.; and Bovik, A. C. 2003. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, 1398–1402. Ieee.

Wiegand, T.; Sullivan, G. J.; Bjontegaard, G.; and Luthra, A. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7): 560–576.

Wu, C.-Y.; Singhal, N.; and Krahenbuhl, P. 2018. Video compression through image interpolation. In *ECCV*, 416–431.

Yang, R.; Mentzer, F.; Gool, L. V.; and Timofte, R. 2020a. Learning for video compression with hierarchical quality and recurrent enhancement. In *CVPR*, 6628–6637.

Yang, R.; Yang, Y.; Marino, J.; and Mandt, S. 2020b. Hierarchical autoregressive modeling for neural video compression. *arXiv preprint arXiv:2010.10258*.

Yu, A.; Ye, V.; Tancik, M.; and Kanazawa, A. 2021. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 4578–4587.

Zhang, S.; Liu, K.; Gu, J.; Cai, X.; Wang, Z.; Bu, J.; and Wang, H. 2025. Attention Beats Linear for Fast Implicit Neural Representation Generation. In *ECCV 2024*, 1–18.