JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

NIRVANA: Neural Implicit Representations of Videos with Adaptive Networks and Autoregressive Patch-wise Modeling

Shishira R Maiya* , Sharath Girish*, Max Ehrlich , Hanyu Wang , Kwot Sin Lee , Patrick Poirson , Pengxiang Wu , Chen Wang , Abhinav Shrivastava

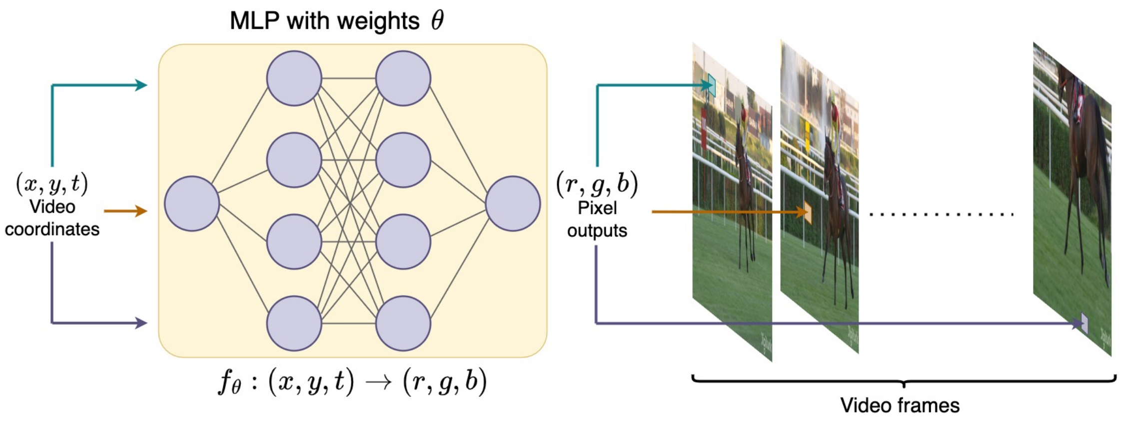
University of Maryland, College Park

Snap Inc

Poster: WED-PM-194

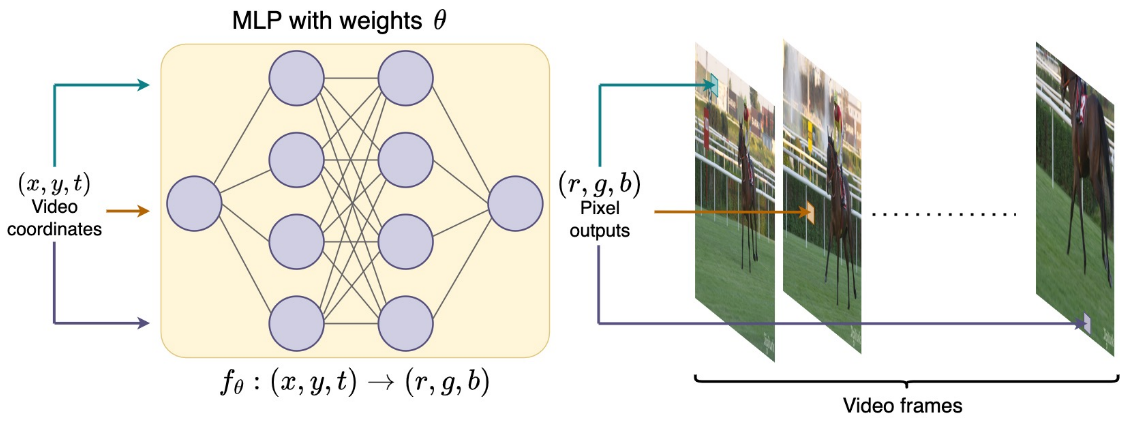
INRs for videos

SIRENs: MLP networks with sinusoidal activations



INRs for videos

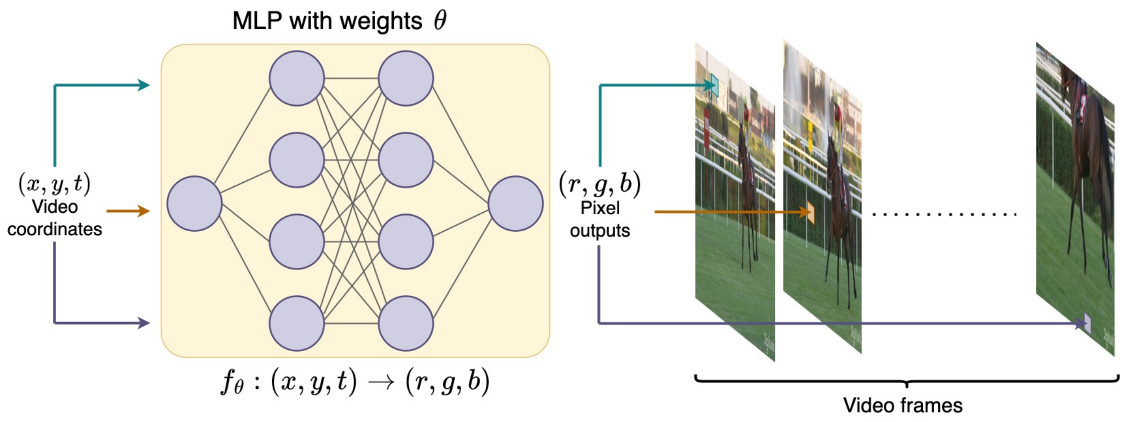
SIRENs: MLP networks with sinusoidal activations



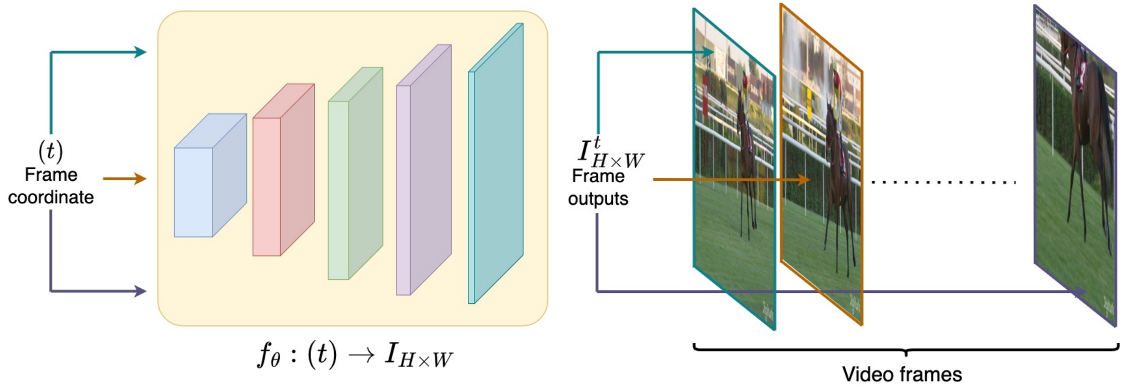
Pixelwise coordinate input

INRs for videos

SIRENs: MLP networks with sinusoidal activations

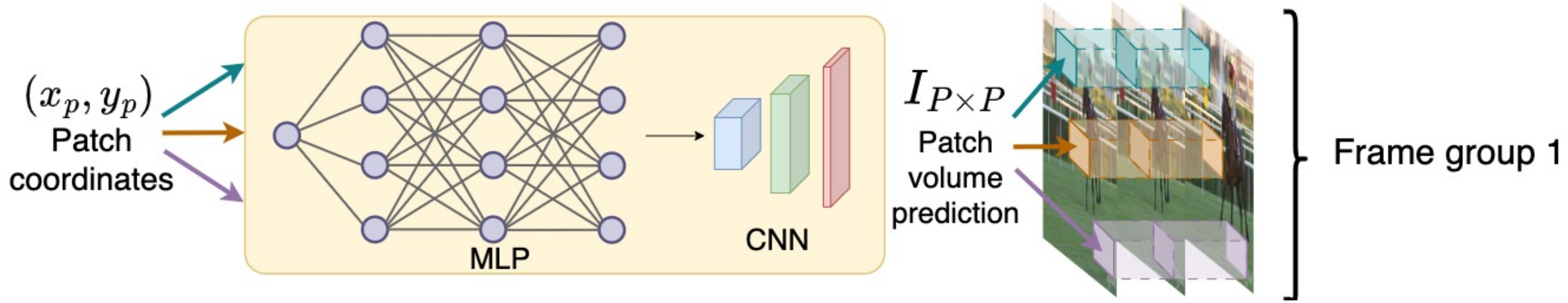


NeRV: Convolutional upsampling networks NERV

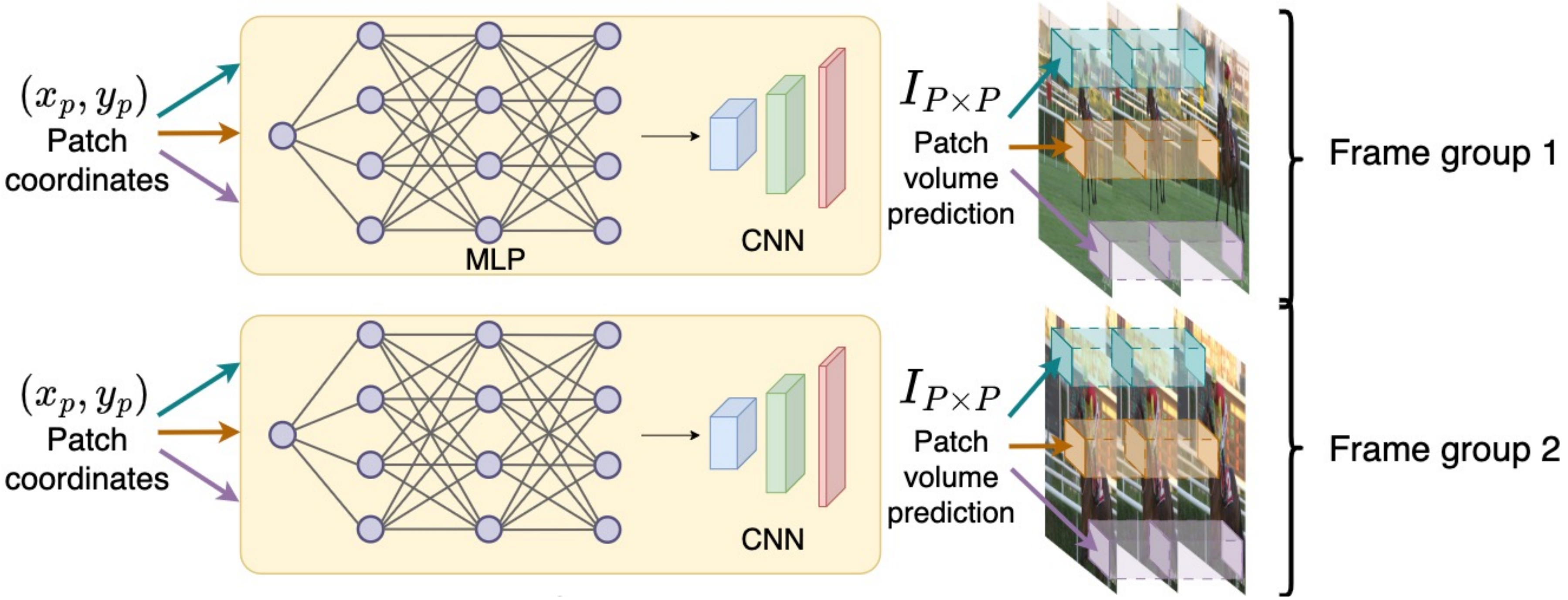


Framewise coordinate input

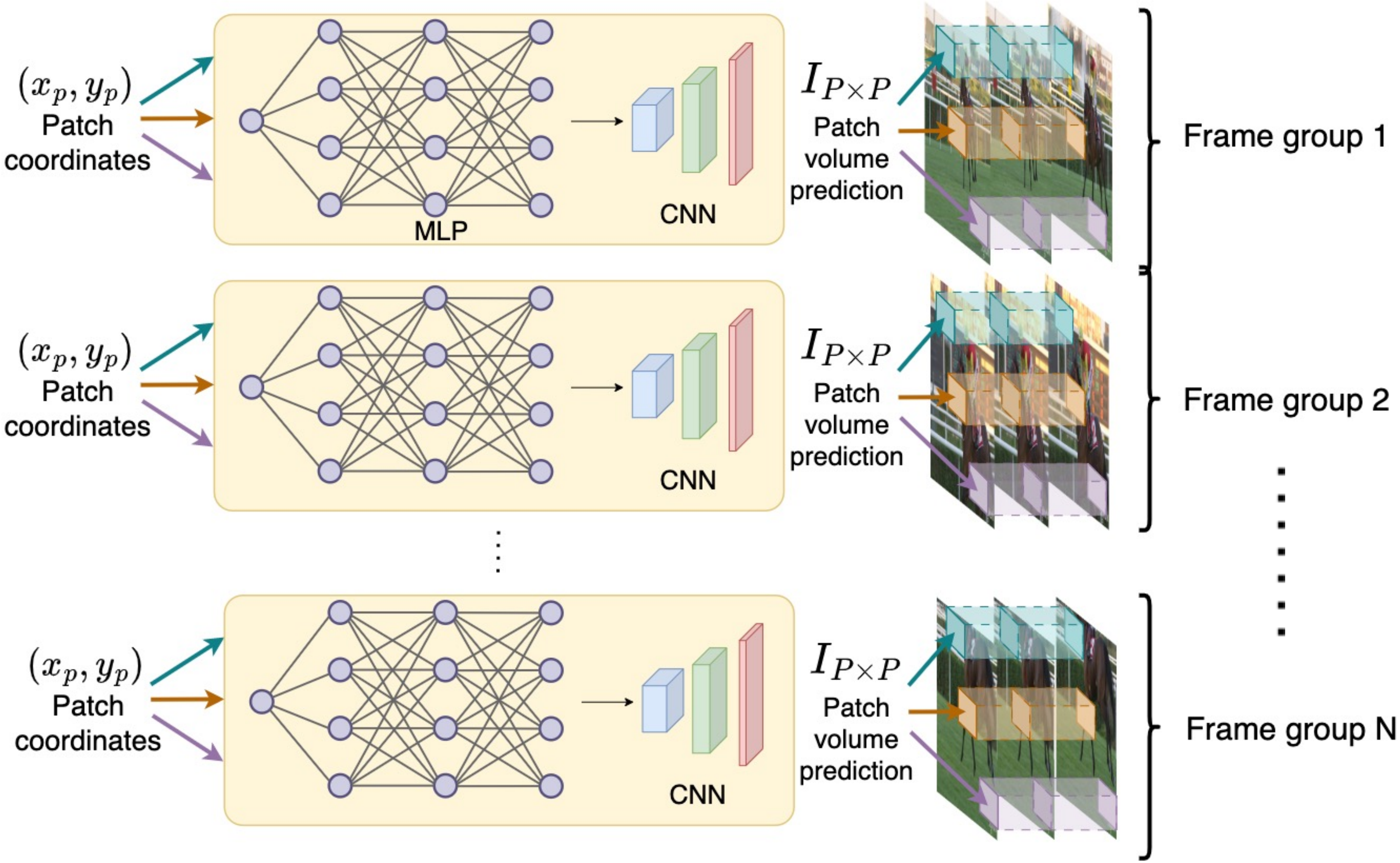
NIRVANA: Video INRs with Patchwise Prediction



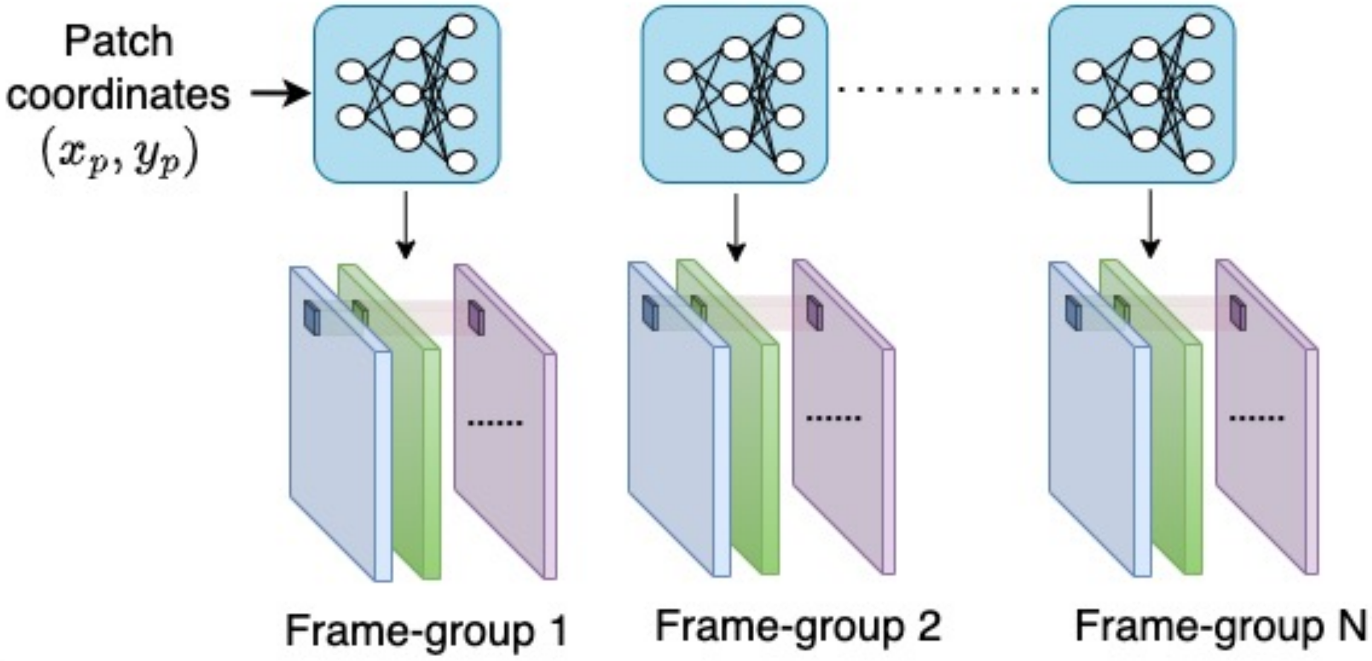
NIRVANA: Video INRs with Patchwise Prediction



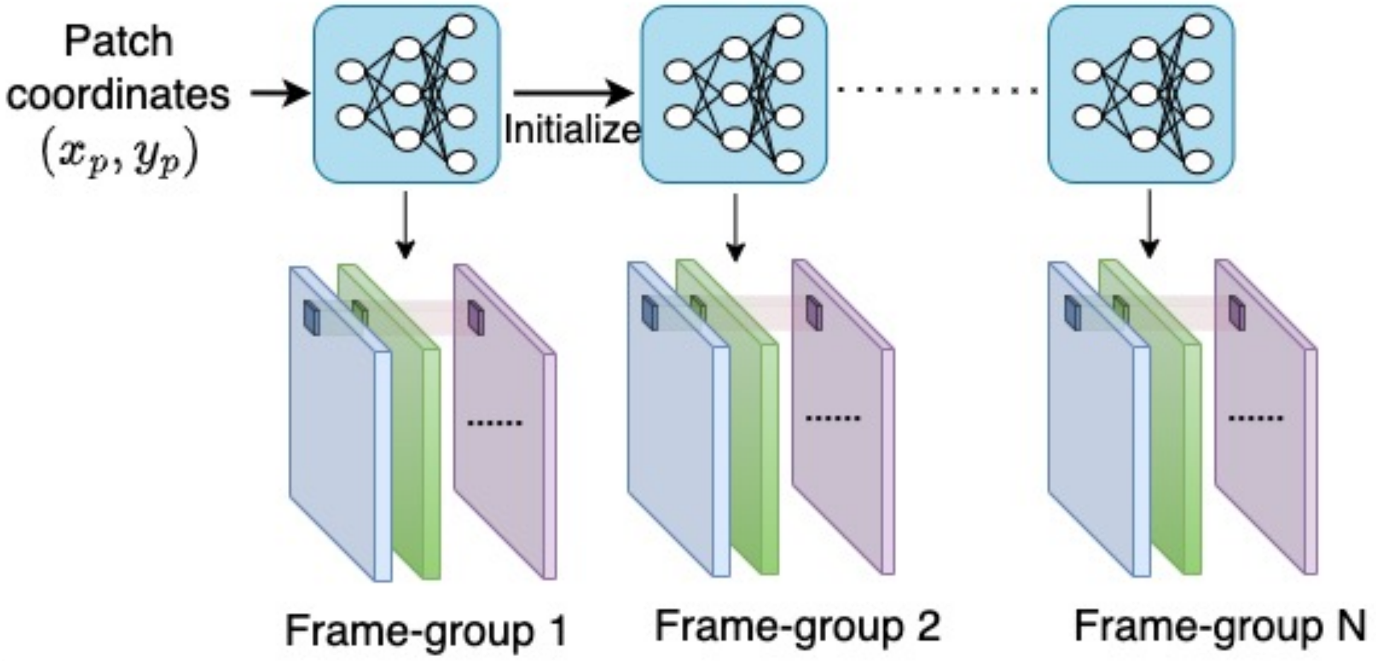
NIRVANA: Video INRs with Patchwise Prediction



Autoregressive modeling

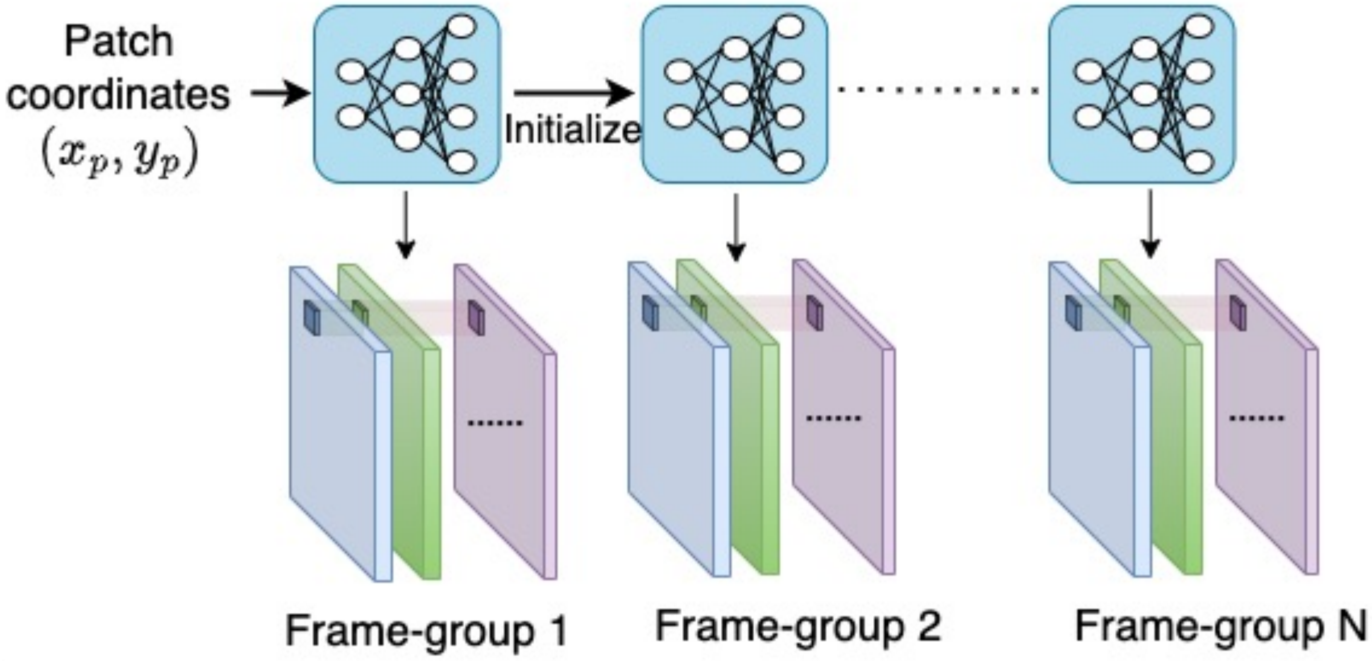


Autoregressive modeling

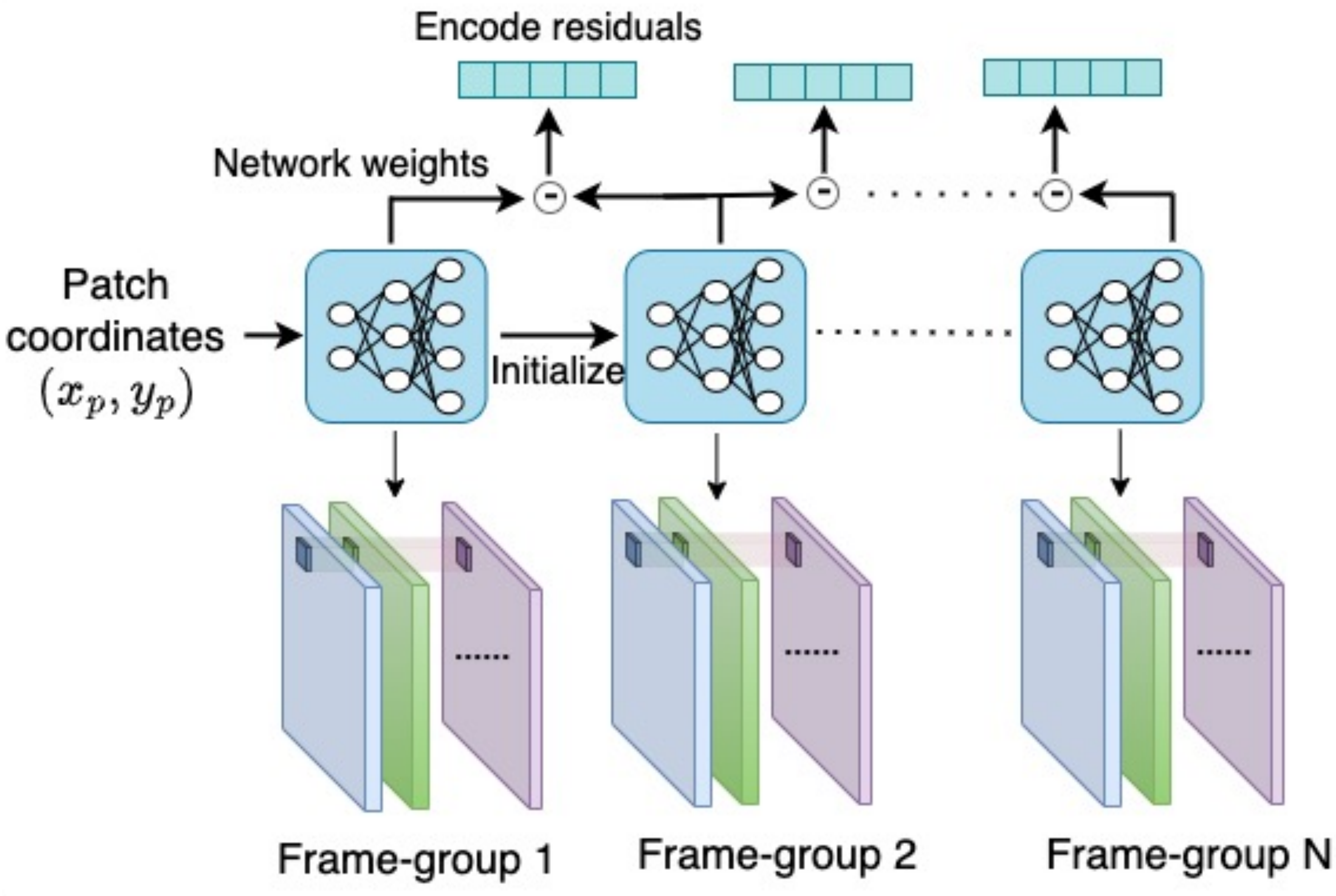


Autoregressive modeling

Faster training/convergence using previous weight initialization.

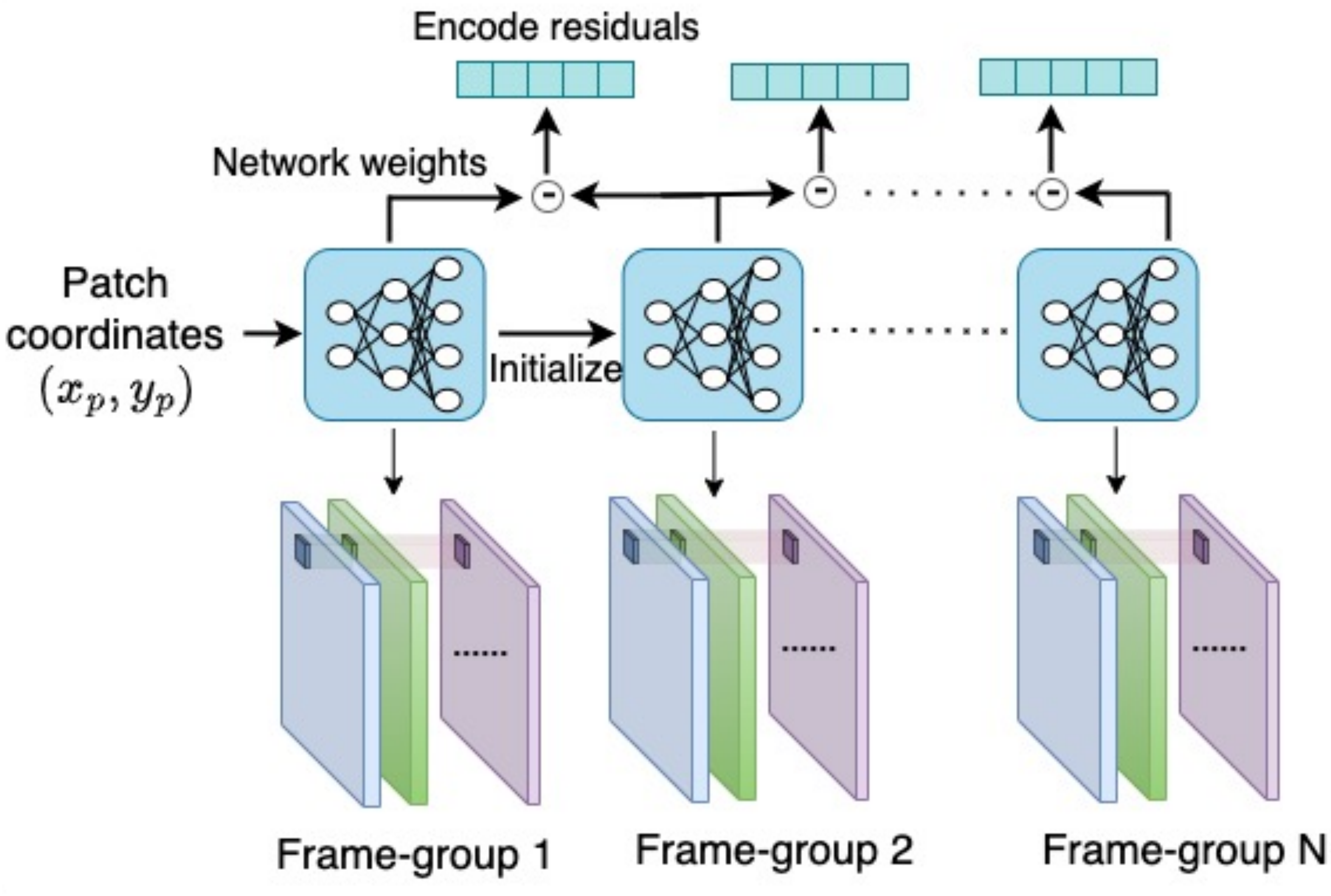


Autoregressive modeling



Faster training/convergence using previous weight initialization.

Autoregressive modeling



Faster training/convergence using previous weight initialization.

Lower network size by storing only sparse weight residuals.

Faster encoding/decoding speed

Dataset	Method	Encoding Time (Hours) ↓	Decoding Speed (FPS) ↑	PSNR ↑	BPP ↓
UVG-HD	SIREN	$\sim$30	15.62	27.20	**0.28**
	NIRVANA (Ours)	**5.44**	**87.91**	**34.71**	0.32
	NeRV	$\sim$80	11.01	37.36	0.92
	NIRVANA (Ours)	**6.71**	**65.42**	**37.70**	**0.86**
UVG-4K	NeRV	$\sim$134	8.27	**35.24**	0.28
	NIRVANA (Ours)	**20.89**	**50.83**	35.18	**0.27**

12x lower encoding time and 6x faster decoding than NeRV at similar PSNR and BPP

Faster encoding/decoding speed

Dataset	Method	Encoding Time (Hours) ↓	Decoding Speed (FPS) ↑	PSNR ↑	BPP ↓
UVG-HD	SIREN	∼30	15.62	27.20	**0.28**
	NIRVANA (Ours)	**5.44**	**87.91**	**34.71**	0.32
	NeRV	∼80	11.01	37.36	0.92
	NIRVANA (Ours)	**6.71**	$\boxed{\mathbf{65.42}}$	**37.70**	**0.86**
UVG-4K	NeRV	∼134	8.27	**35.24**	0.28
	NIRVANA (Ours)	**20.89**	**50.83**	35.18	**0.27**

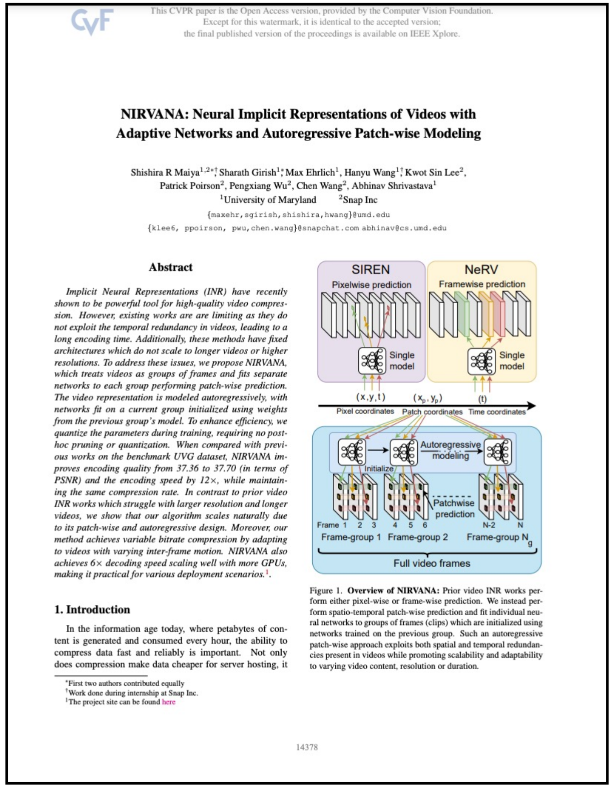
Real time decoding of videos!

Additional details

Paper and full presentation include:

- Scalability of our approach to longer and larger resolution videos.
- Additional qualitative and quantitative results.

Visit us at poster #194 on Wednesday evening session (4:30 PM – 6:30 PM) at CVPR 2023!



NIRVANA: Neural Implicit Representations of Videos with Adaptive Networks and Autoregressive Patch-wise Modeling

Shishira R Maiya[1,2*†], Sharath Girish[1*], Max Ehrlich[1], Hanyu Wang[1†], Kwot Sin Lee[2], Patrick Poirson[2], Pengxiang Wu[2], Chen Wang[2], Abhinav Shrivastava[1]
[1]University of Maryland [2]Snap Inc
{maxehr,sgirish,shishira,hwang}@umd.edu
{klee6, ppoirson, pwu,chen.wang}@snapchat.com abhinav@cs.umd.edu

Abstract

Implicit Neural Representations (INR) have recently shown to be powerful tool for high-quality video compression. However, existing works are are limiting as they do not exploit the temporal redundancy in videos, leading to a long encoding time. Additionally, these methods have fixed architectures which do not scale to longer videos or higher resolutions. To address these issues, we propose NIRVANA, which treats videos as groups of frames and fits separate networks to each group performing patch-wise prediction. The video representation is modeled autoregressively, with networks fit on a current group initialized using weights from the previous group's model. To enhance efficiency, we quantize the parameters during training, requiring no post-hoc pruning or quantization. When compared with previous works on the benchmark UVG dataset, NIRVANA improves encoding quality from 37.36 to 37.70 (in terms of PSNR) and the encoding speed by 12×, while maintaining the same compression rate. In contrast to prior video INR works which struggle with larger resolution and longer videos, we show that our algorithm scales naturally due to its patch-wise and autoregressive design. Moreover, our method achieves variable bitrate compression by adapting to videos with varying inter-frame motion. NIRVANA also achieves 6× decoding speed scaling well with more GPUs, making it practical for various deployment scenarios.[1].

Figure 1. **Overview of NIRVANA:** Prior video INR works perform either pixel-wise or frame-wise prediction. We instead perform spatio-temporal patch-wise prediction and fit individual neural networks to groups of frames (clips) which are initialized using networks trained on the previous group. Such an autoregressive patch-wise approach exploits both spatial and temporal redundancies present in videos while promoting scalability and adaptability to varying video content, resolution or duration.

1. Introduction

In the information age today, where petabytes of content is generated and consumed every hour, the ability to compress data fast and reliably is important. Not only does compression make data cheaper for server hosting, it

*First two authors contributed equally
†Work done during internship at Snap Inc.
[1]The project site can be found here

14378

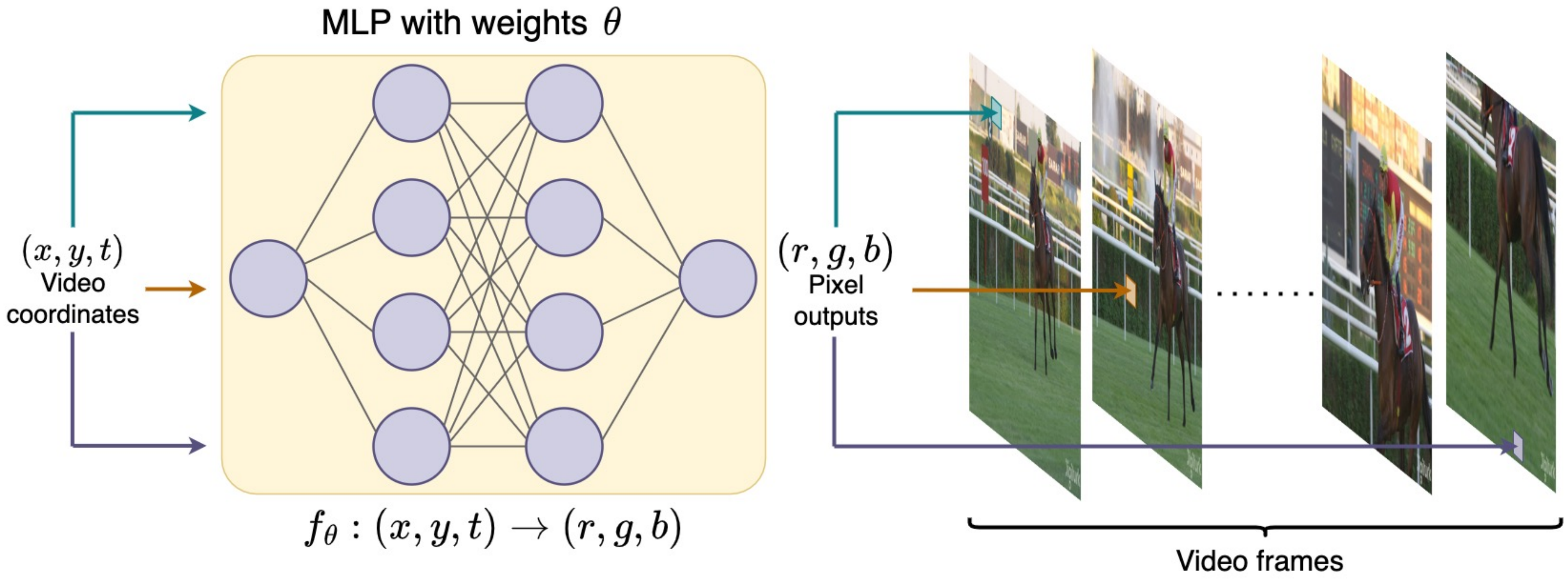
Project page:

Prior works for video INRs

SIRENs: MLP networks with sinusoidal activations

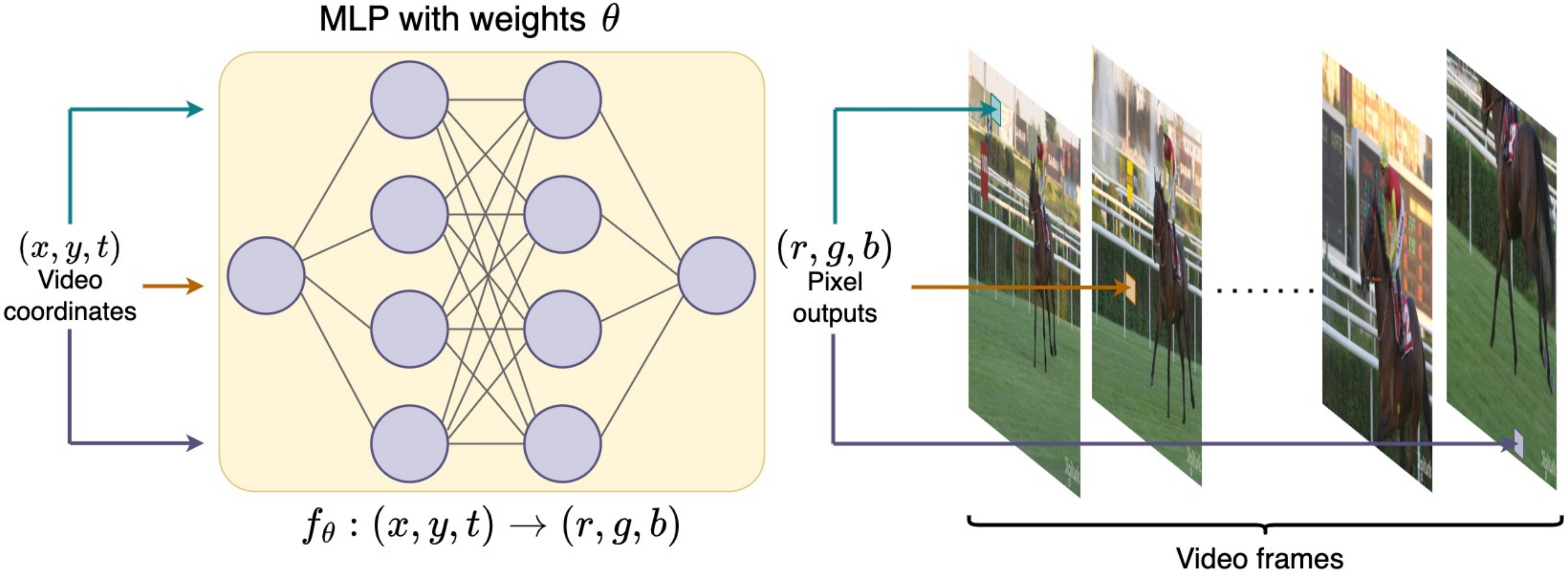
INRs for videos

SIRENs: MLP networks with sinusoidal activations



INRs for videos

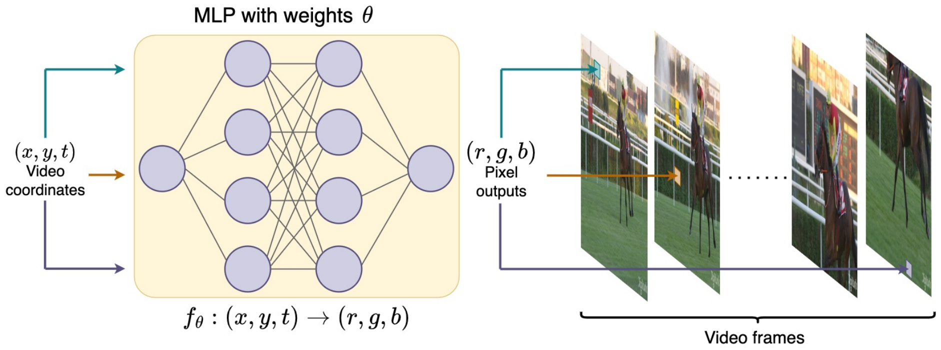
SIRENs: MLP networks with sinusoidal activations



Optimize θ with loss function: $min_\theta \| f_\theta(x, y, t) - I(x, y, t) \|_2^2$

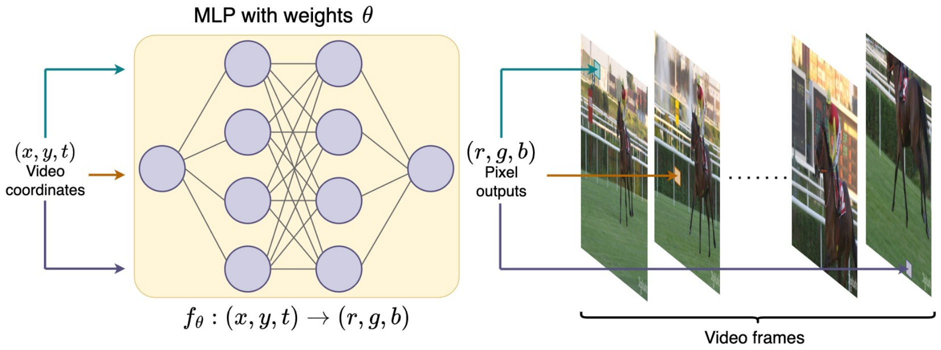
INRs for videos

SIRENs: MLP networks with sinusoidal activations



INRs for videos

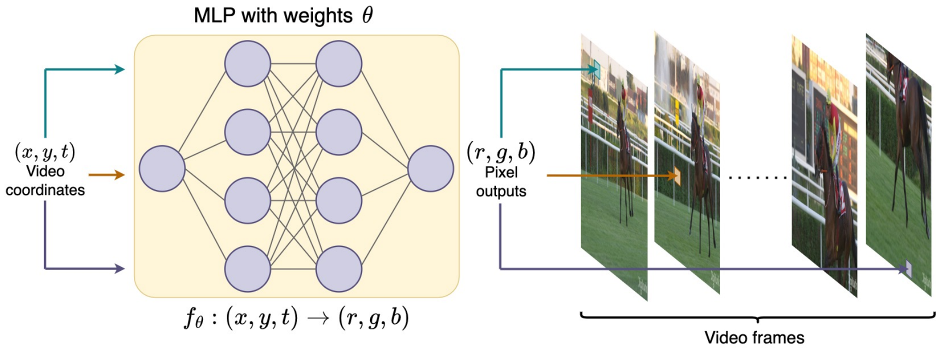
SIRENs: MLP networks with sinusoidal activations



	Sharp reconstruction
SIREN	❌

INRs for videos

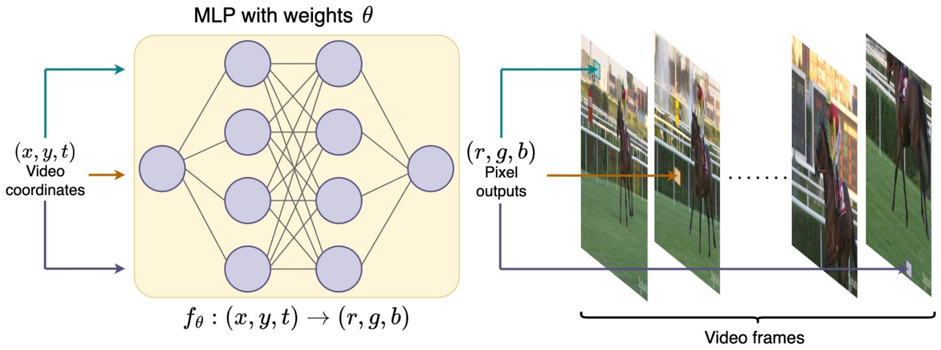
SIRENs: MLP networks with sinusoidal activations



	Sharp reconstruction	Spatial redundancies	Temporal redundancies
SIREN	❌	❌	❌

INRs for videos

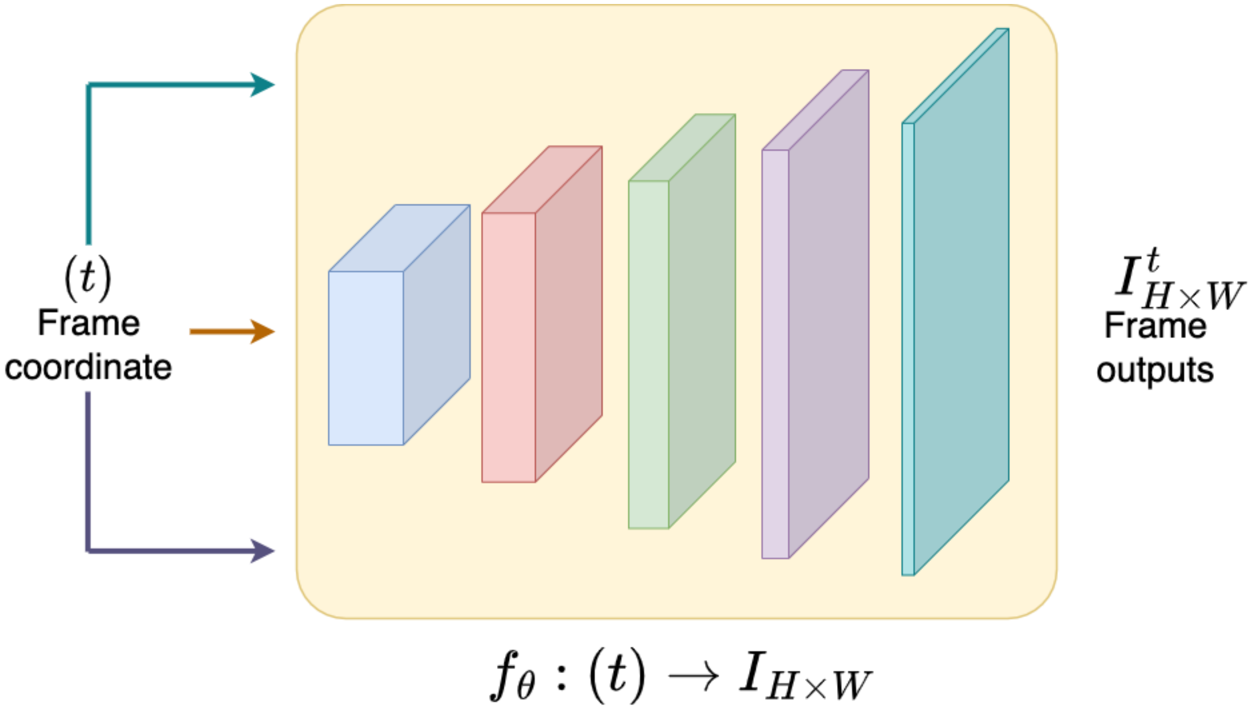
SIRENs: MLP networks with sinusoidal activations



	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	❌	❌	❌	❌

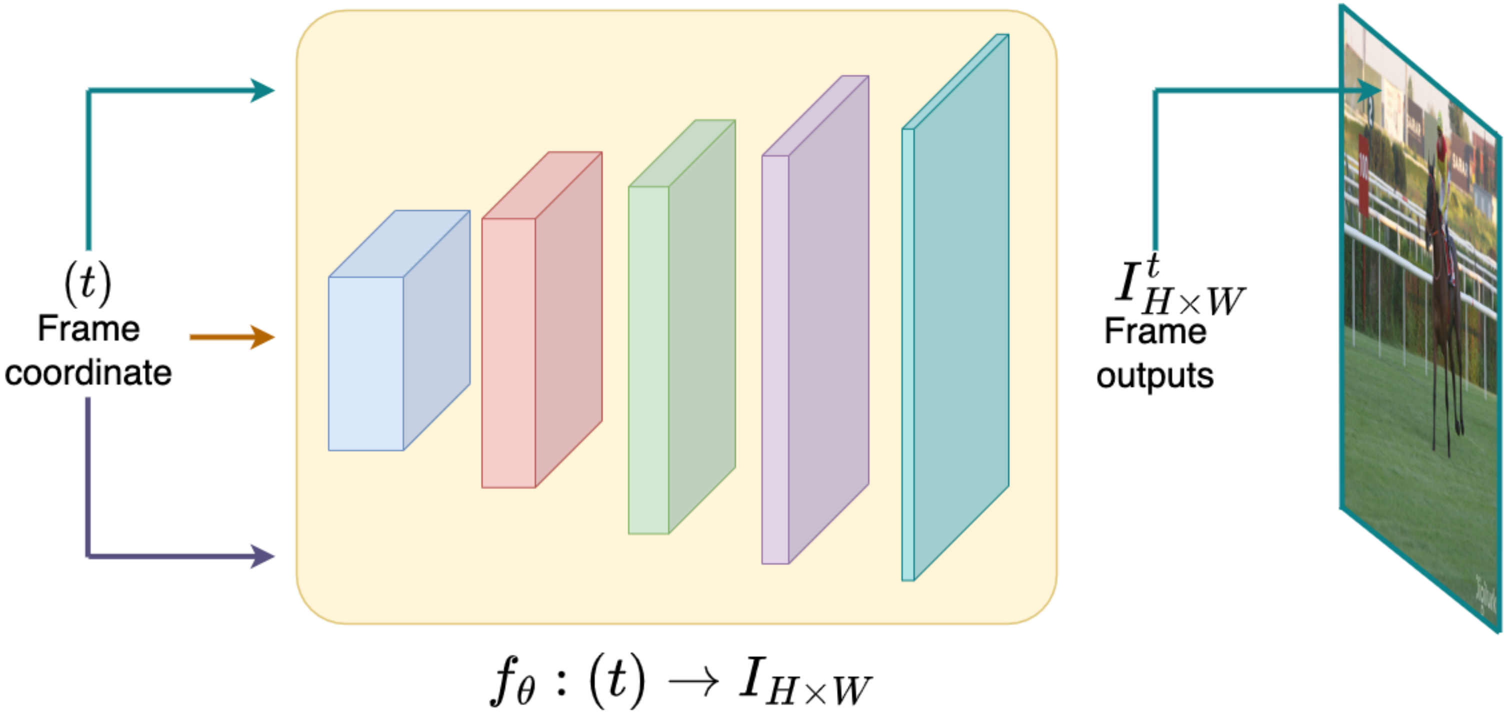
INRs for videos

NeRV: Convolutional upsampling networks



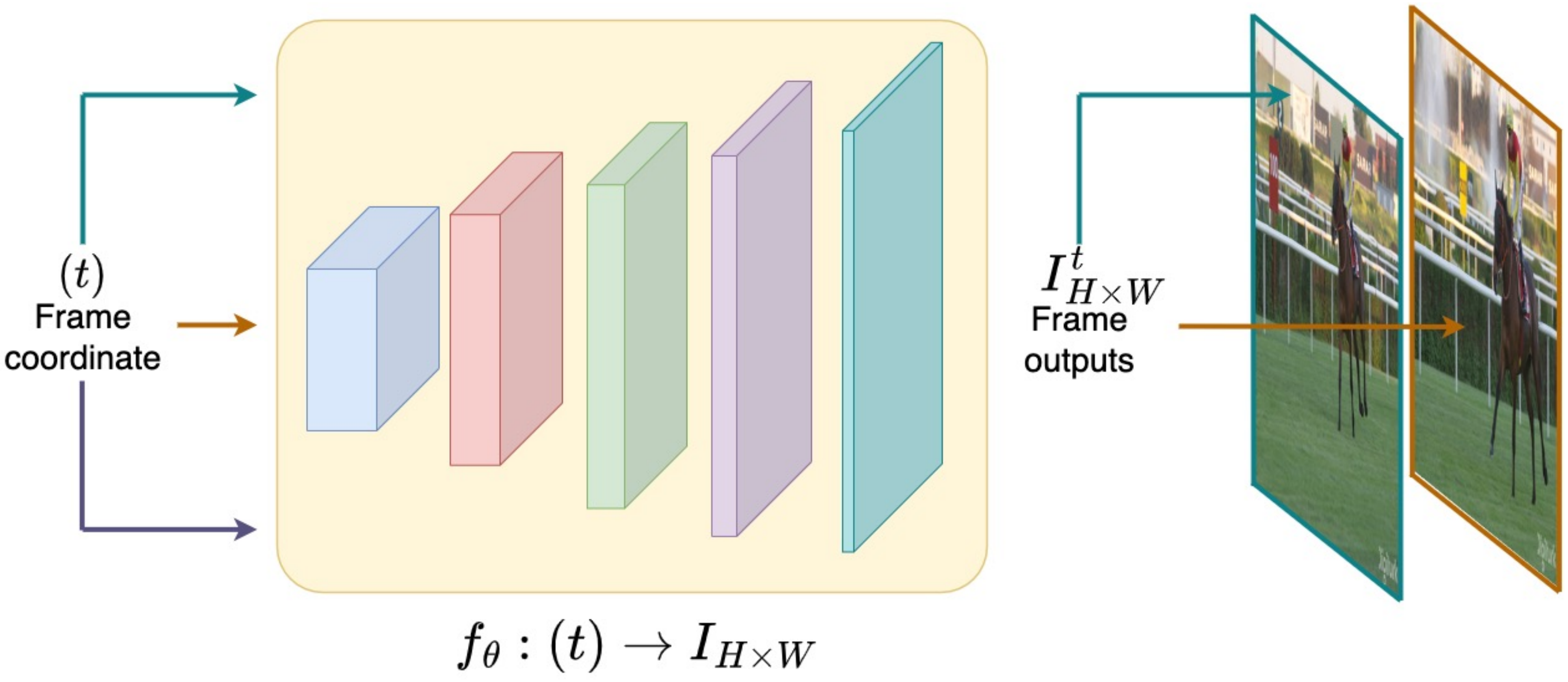
INRs for videos

NeRV: Convolutional upsampling networks



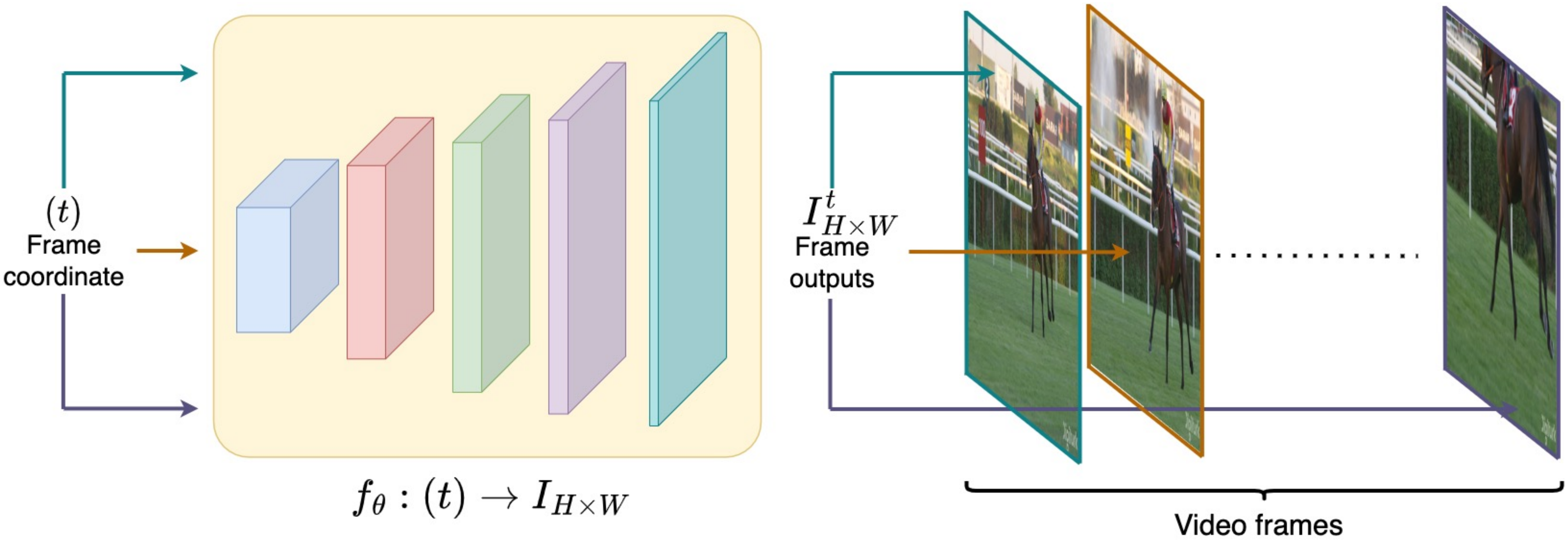
INRs for videos

NeRV: Convolutional upsampling networks



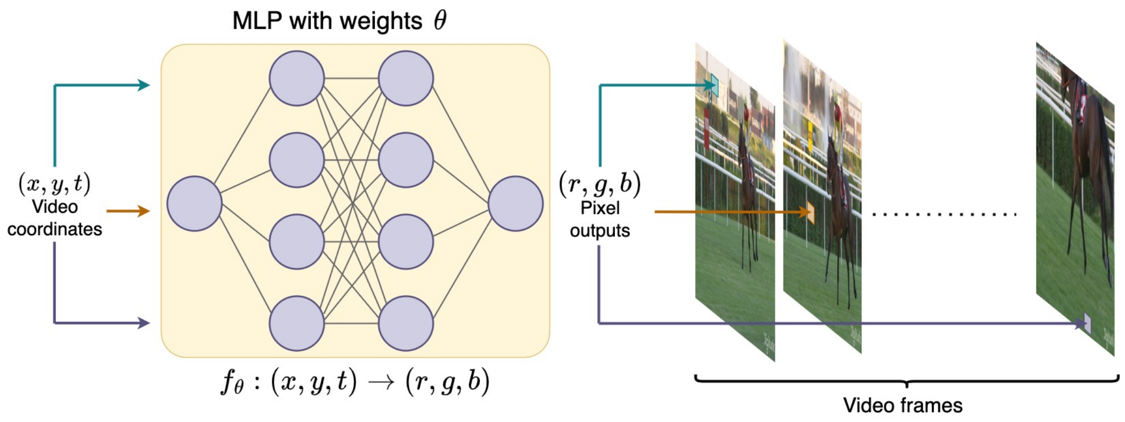
INRs for videos

NeRV: Convolutional upsampling networks

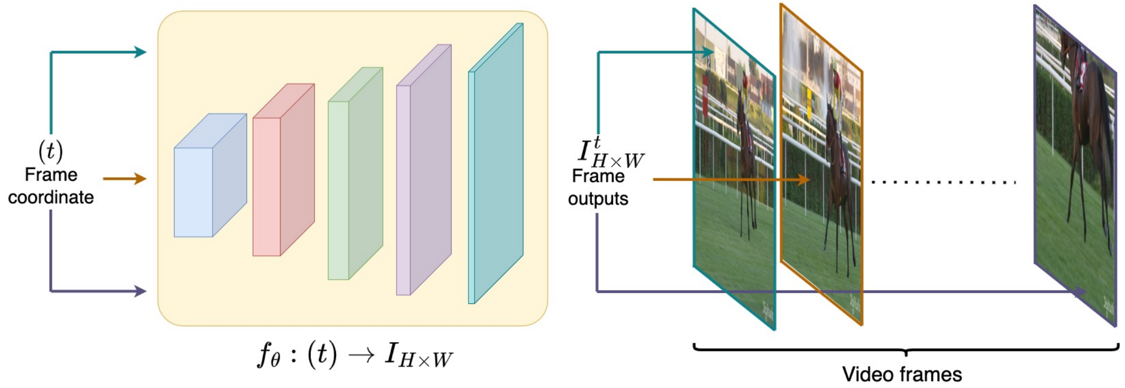


INRs for videos

SIRENs: MLP networks with sinusoidal activations



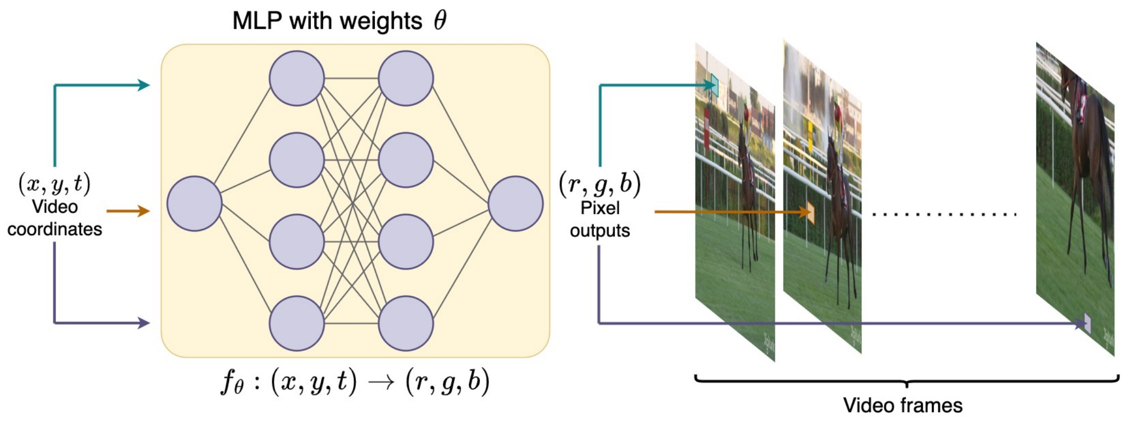
NeRV: Convolutional upsampling networks



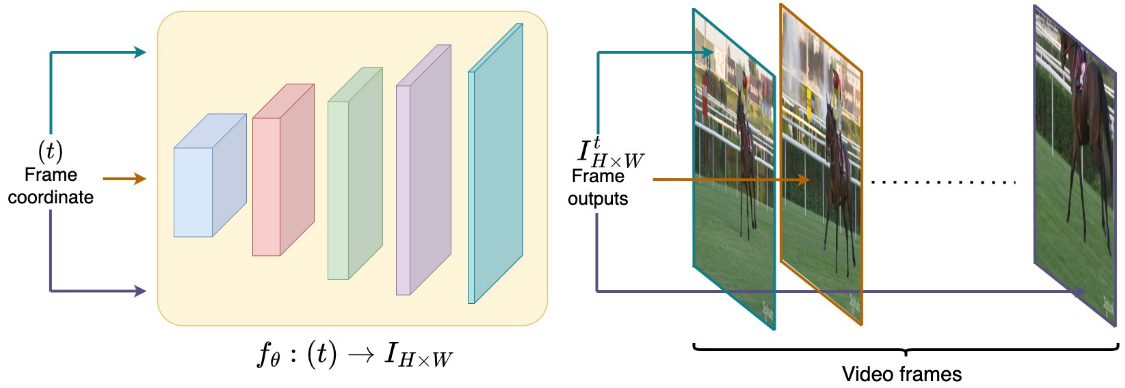
	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	✗	✗	✗	✗
NERV				

INRs for videos

SIRENs: MLP networks with sinusoidal activations



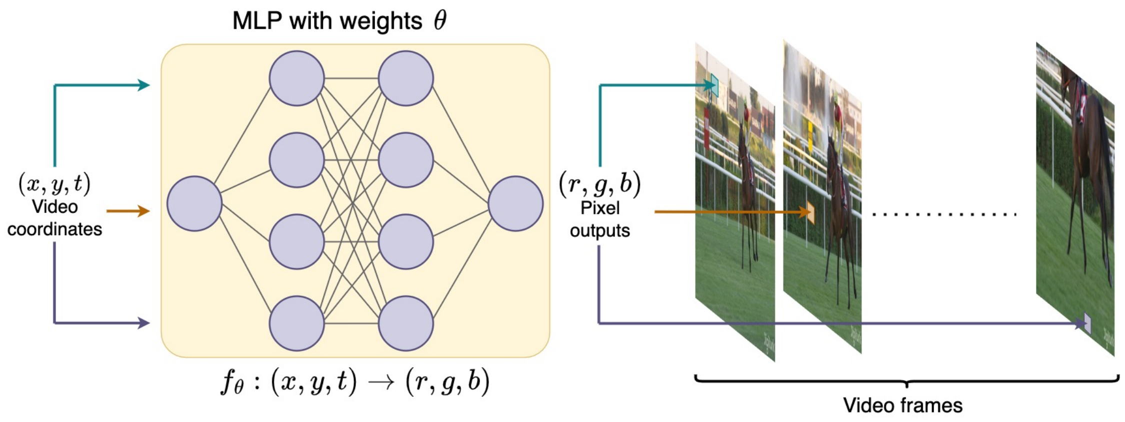
NeRV: Convolutional upsampling networks



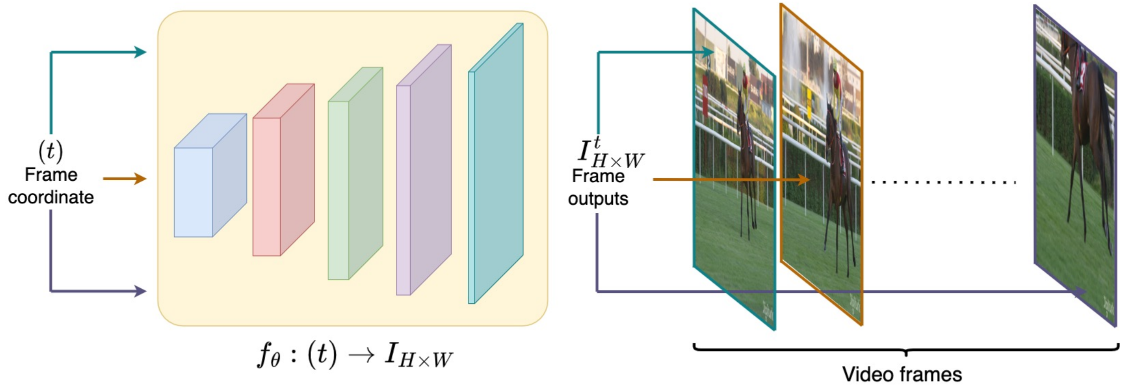
	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	✗	✗	✗	✗
NERV	✓			

INRs for videos

SIRENs: MLP networks with sinusoidal activations



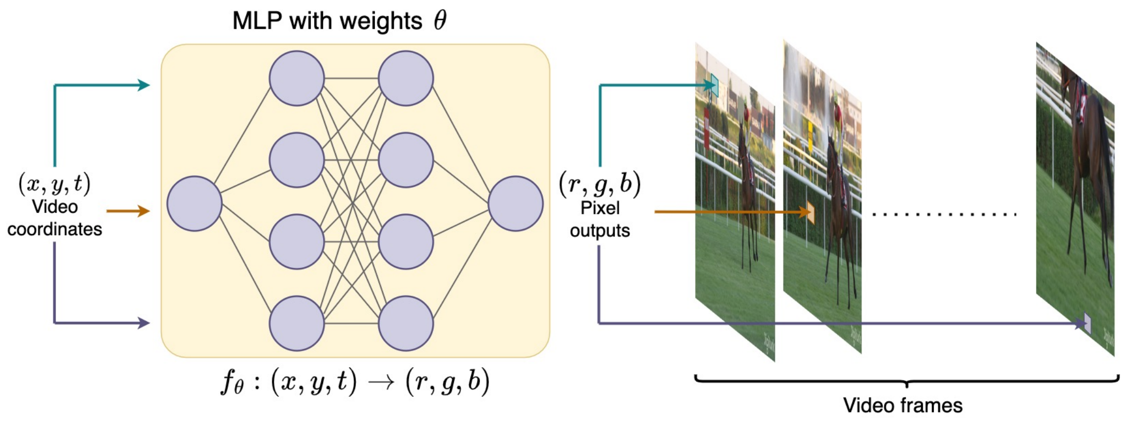
NeRV: Convolutional upsampling networks



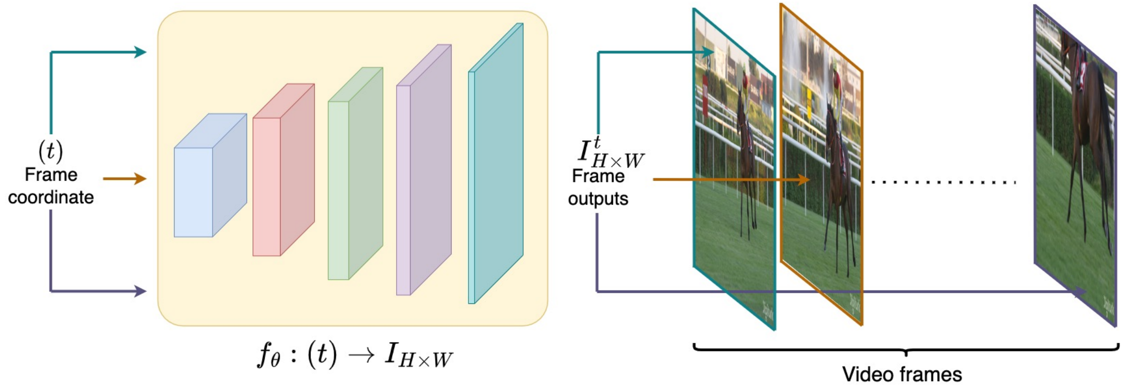
	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	❌	❌	❌	❌
NERV	✅	✅		

INRs for videos

SIRENs: MLP networks with sinusoidal activations



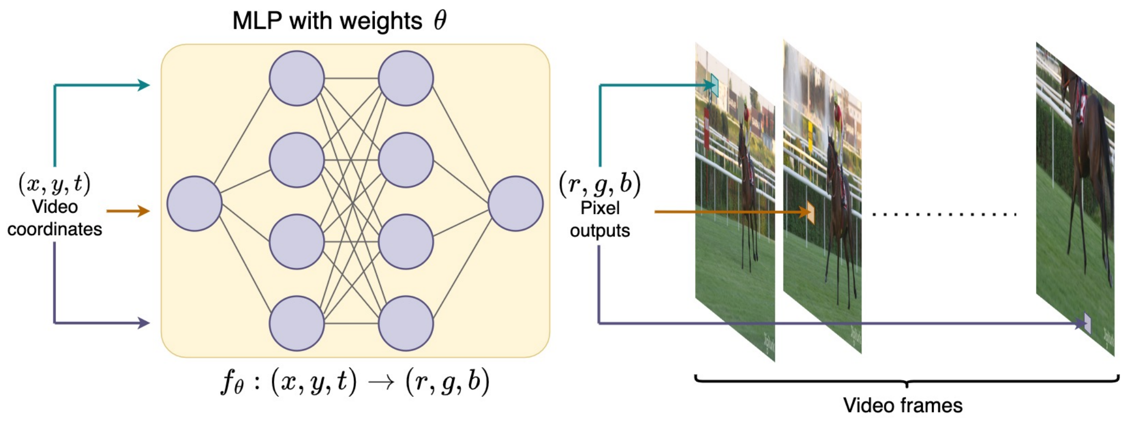
NeRV: Convolutional upsampling networks



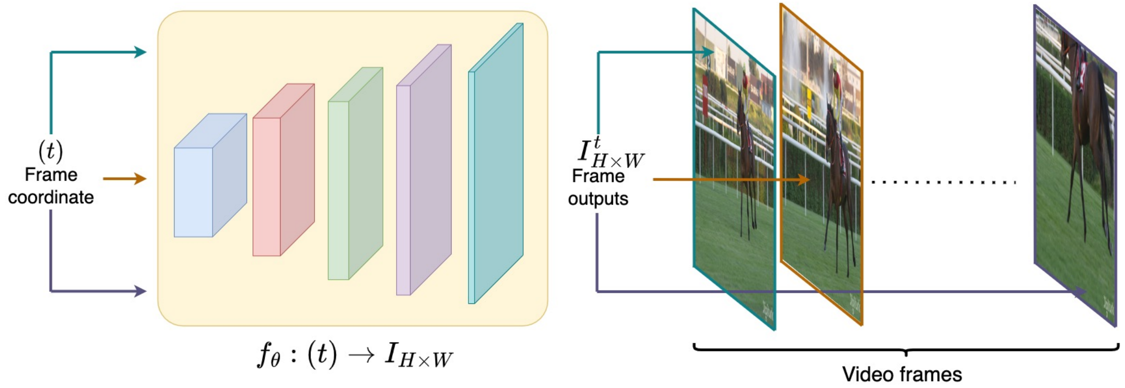
	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	✗	✗	✗	✗
NERV	✓	✓	✗	

INRs for videos

SIRENs: MLP networks with sinusoidal activations

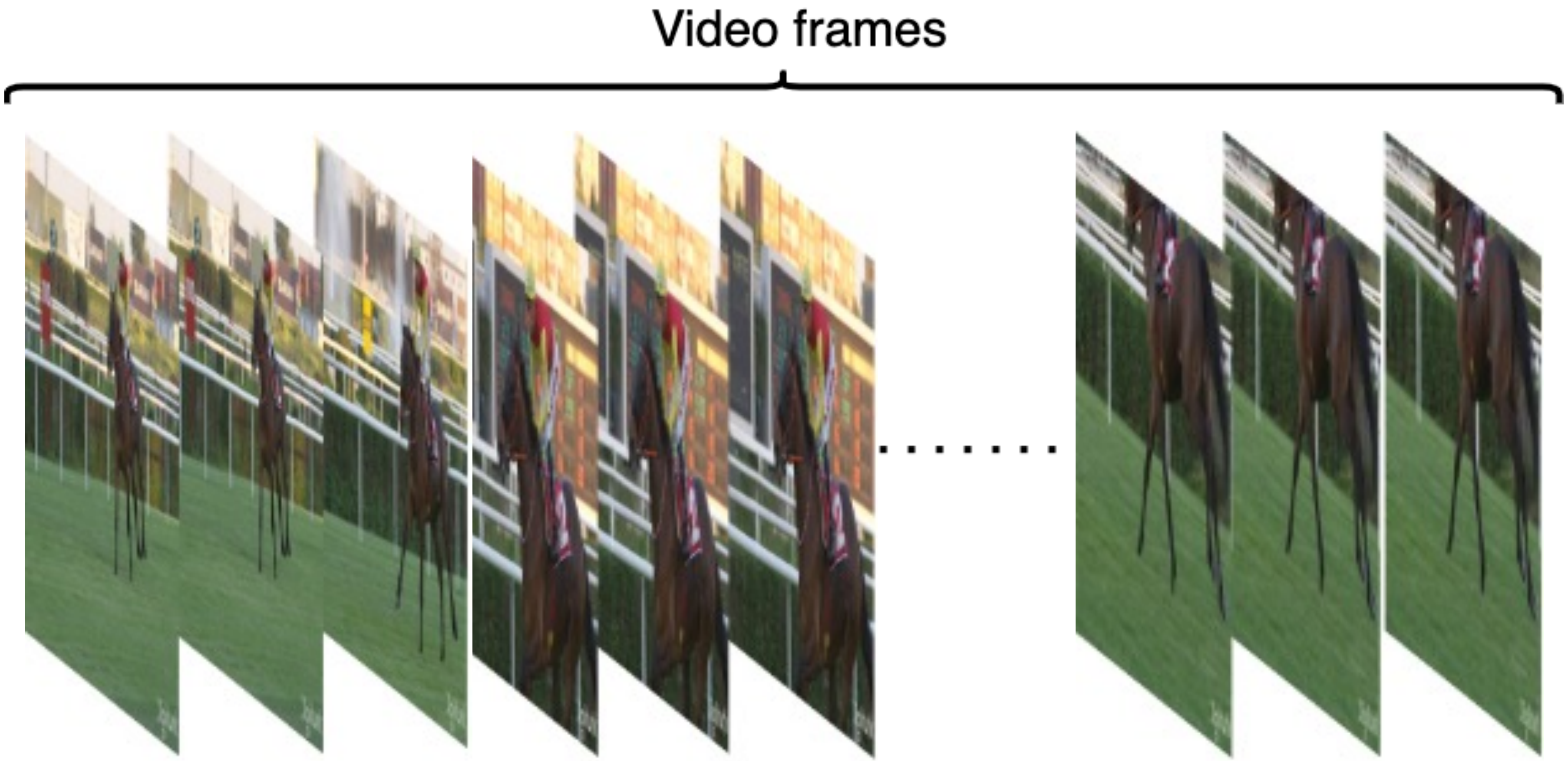


NeRV: Convolutional upsampling networks

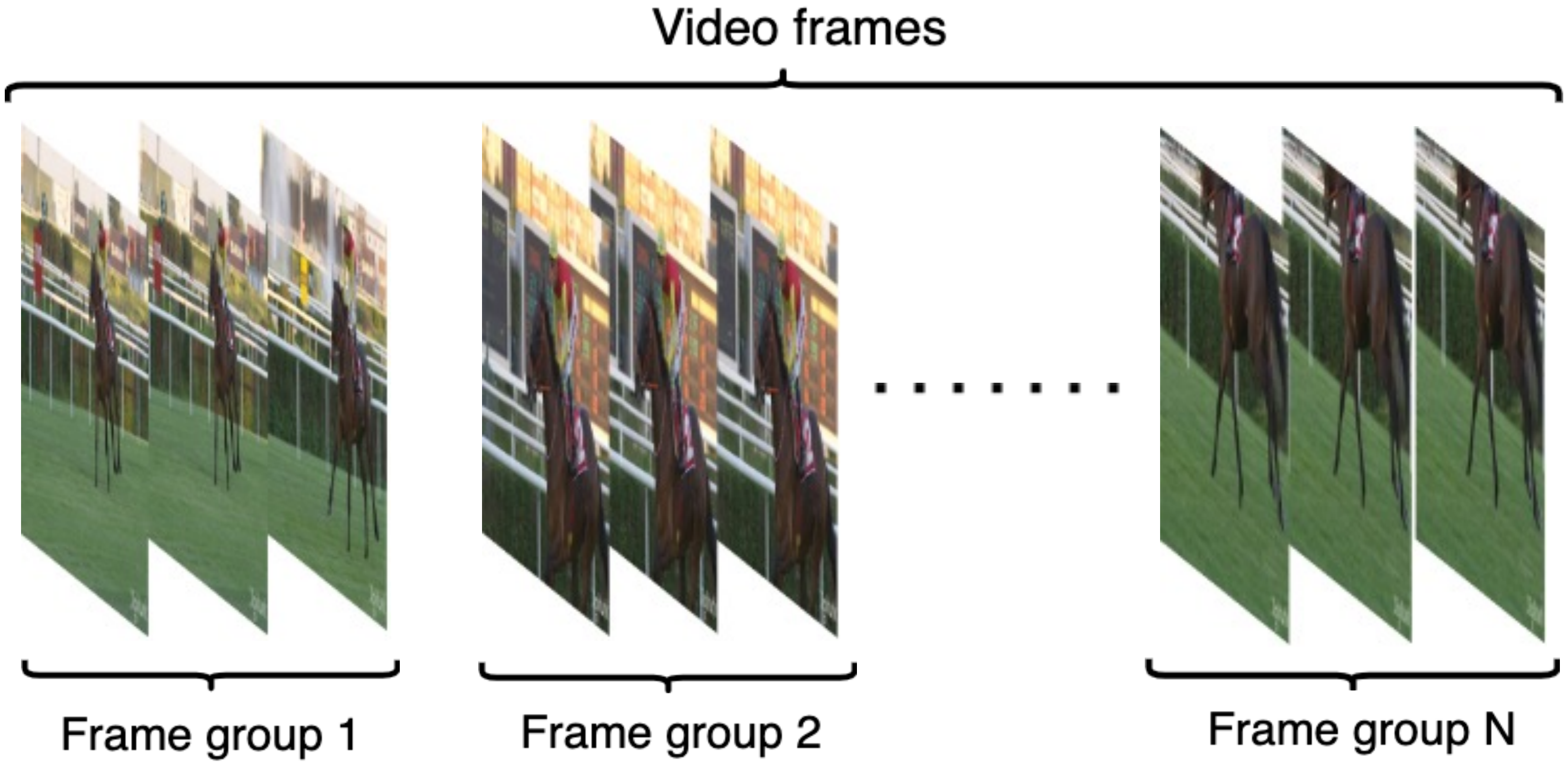


	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	✗	✗	✗	✗
NERV	✓	✓	✗	✗

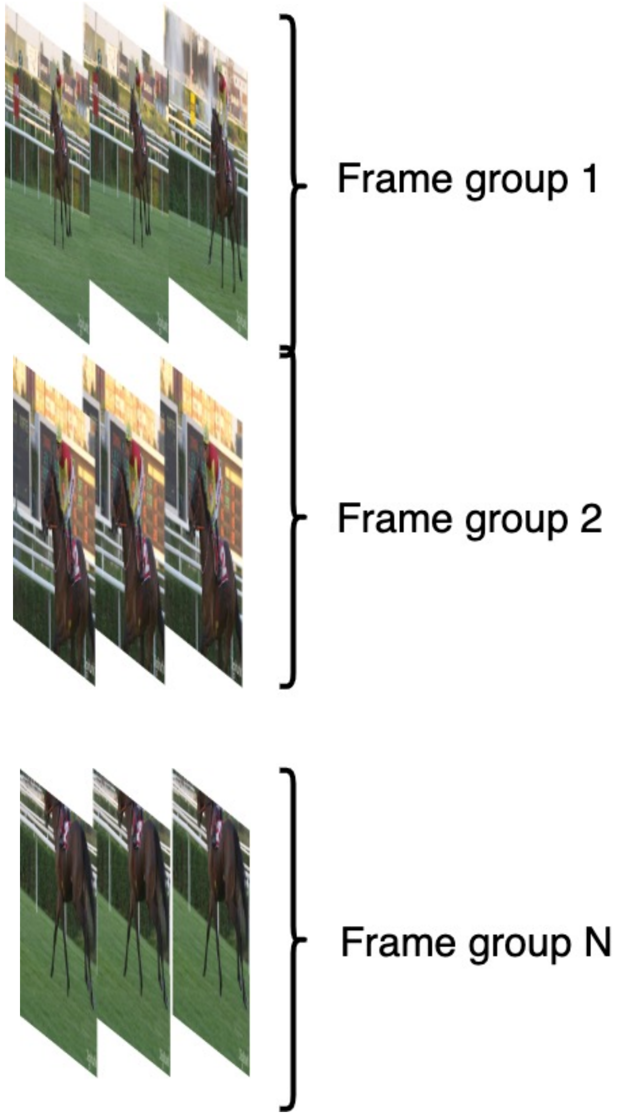
NIRVANA: Video INRs with Patchwise Prediction



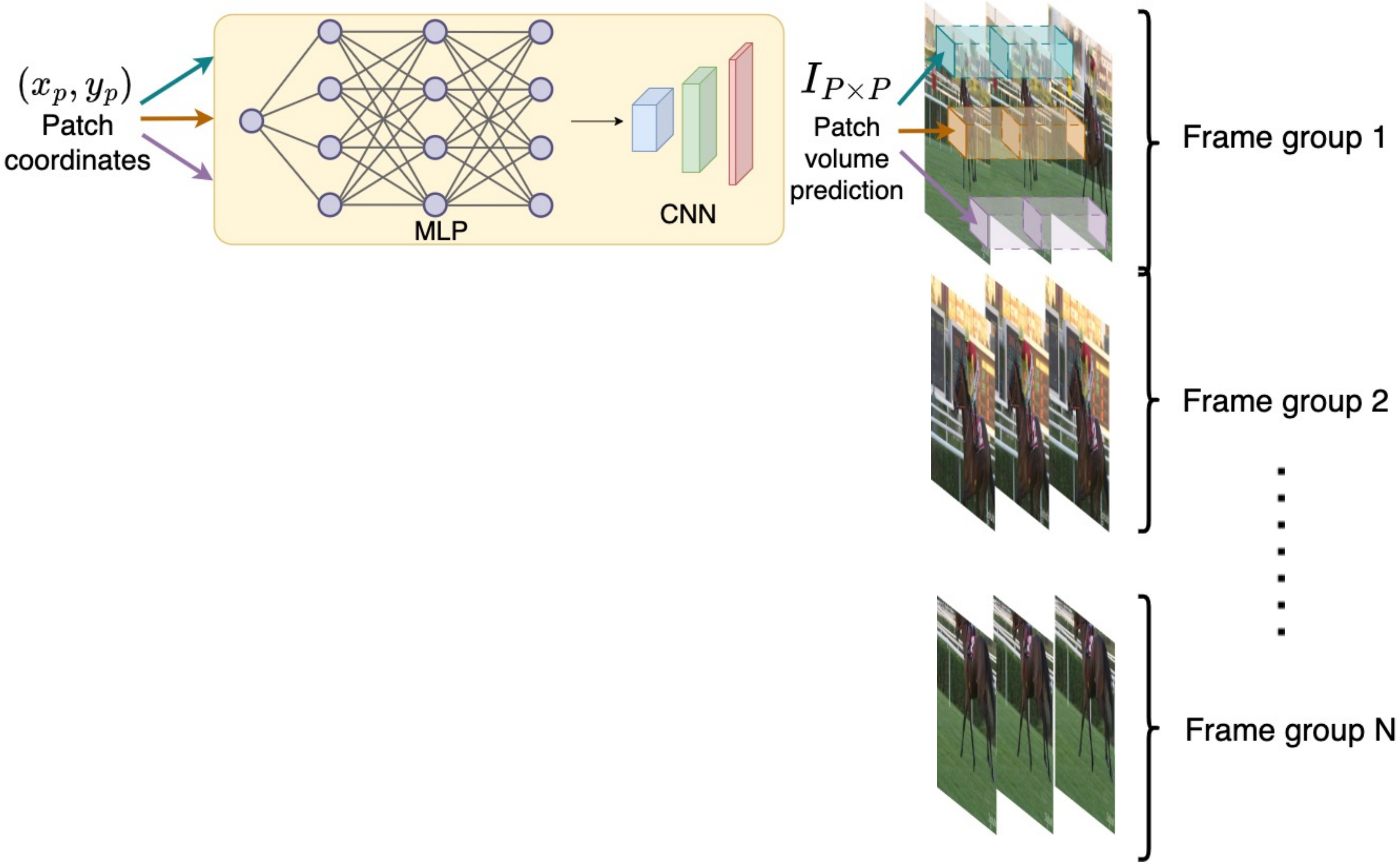
NIRVANA: Video INRs with Patchwise Prediction



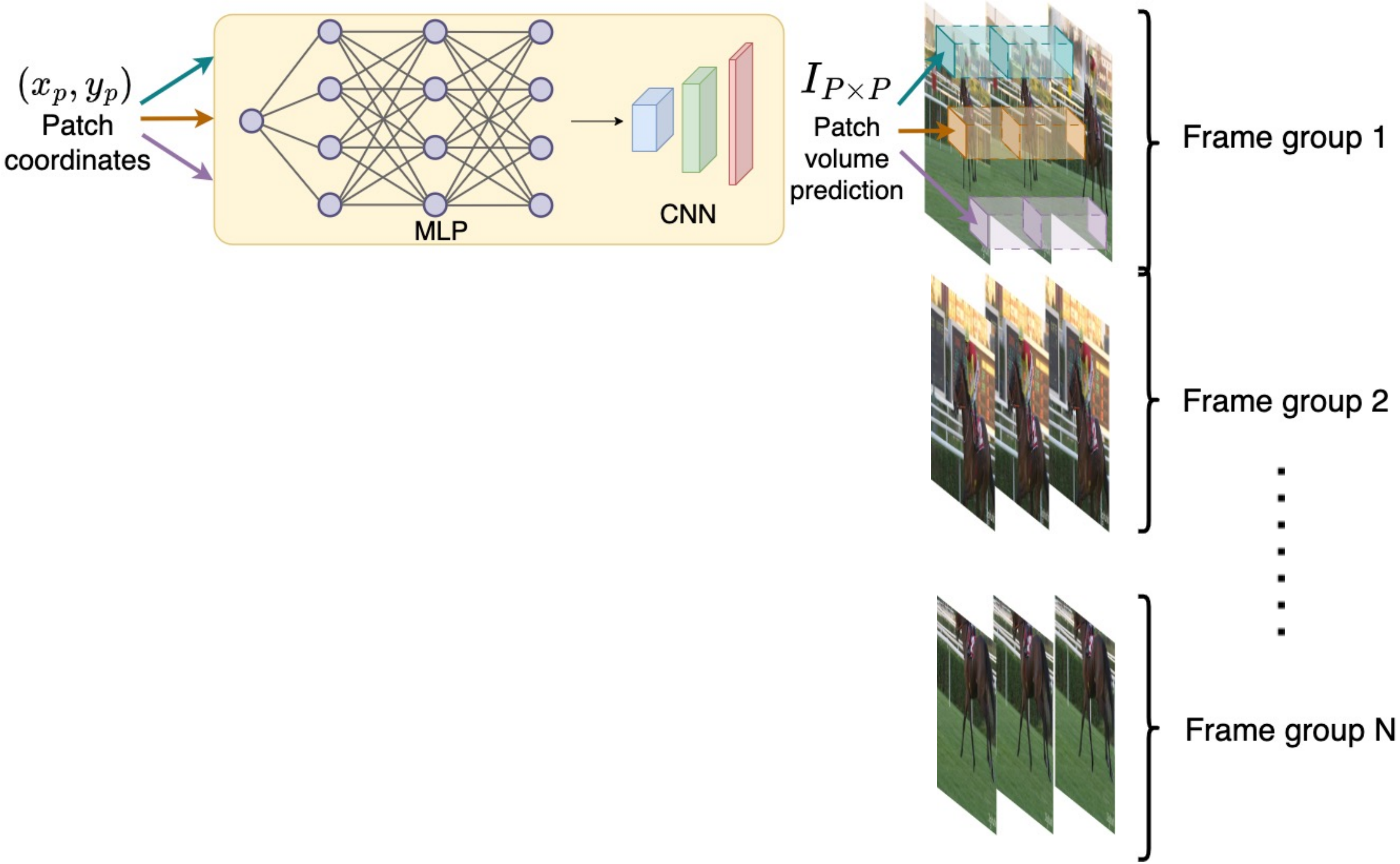
NIRVANA: Video INRs with Patchwise Prediction



NIRVANA: Video INRs with Patchwise Prediction



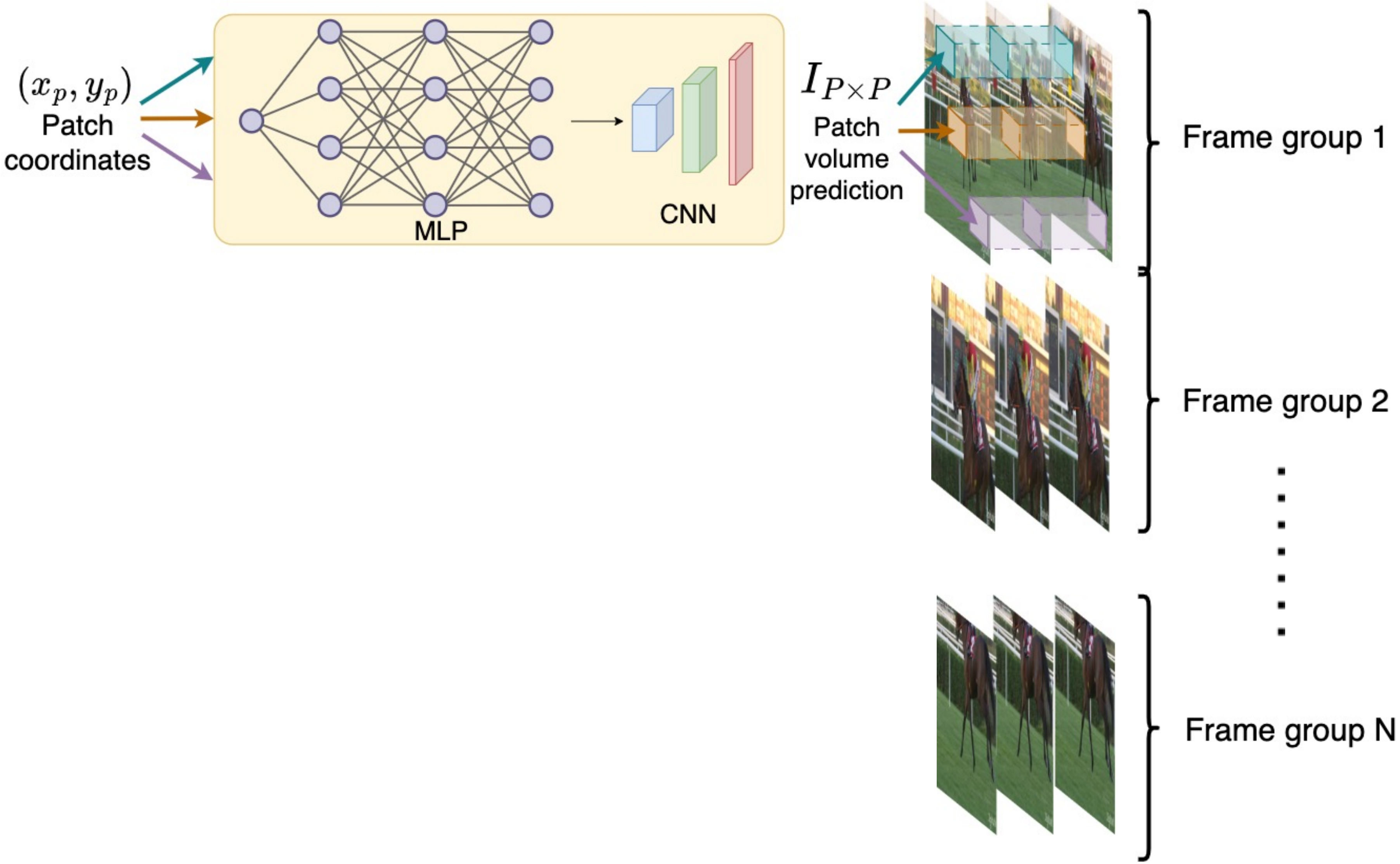
NIRVANA: Video INRs with Patchwise Prediction



Patch volume prediction

- Exploits spatial redundancy via patches.

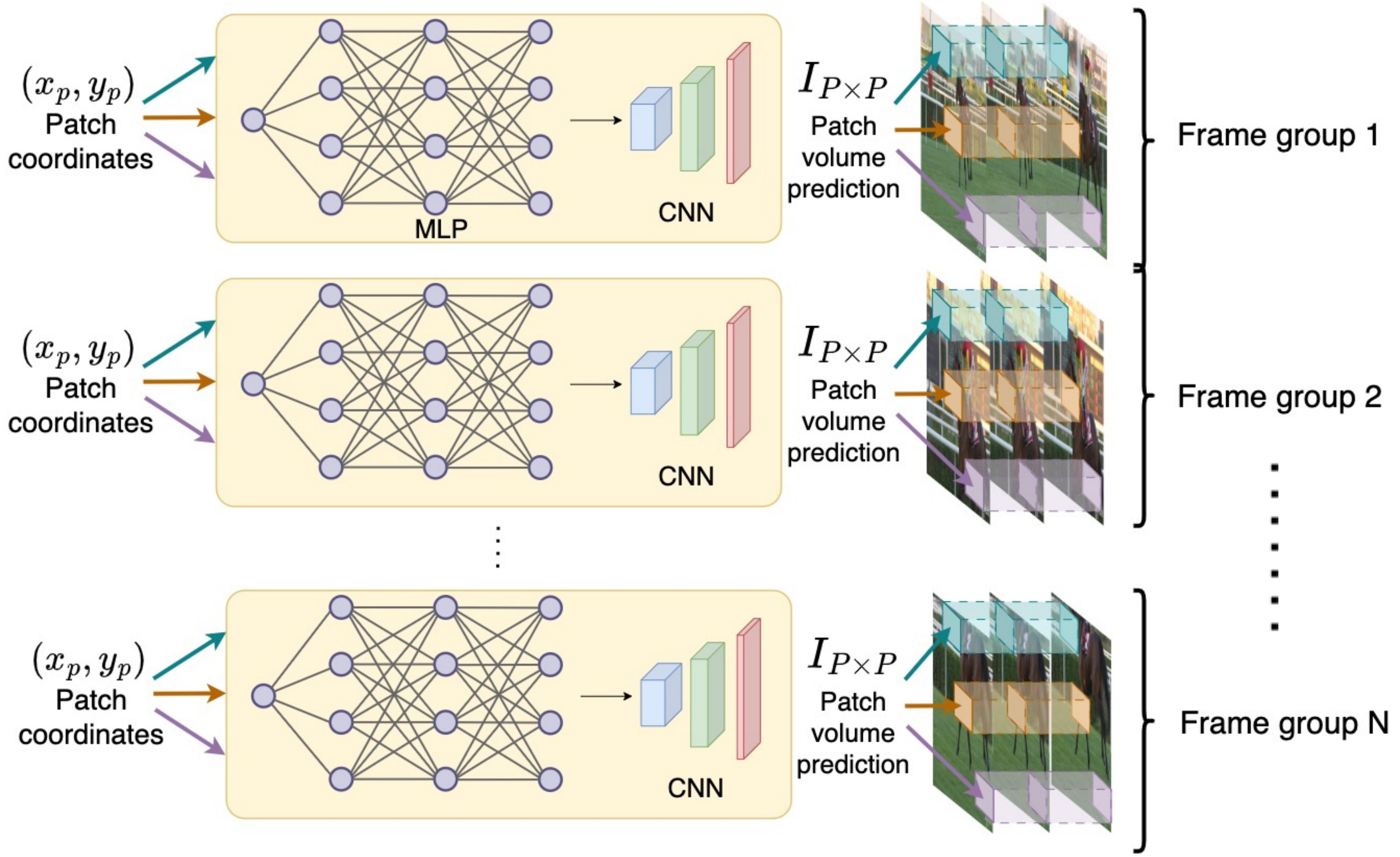
NIRVANA: Video INRs with Patchwise Prediction



Patch volume prediction

- Exploits spatial redundancy via patches.
- Scales well with increasing frame size/resolution.

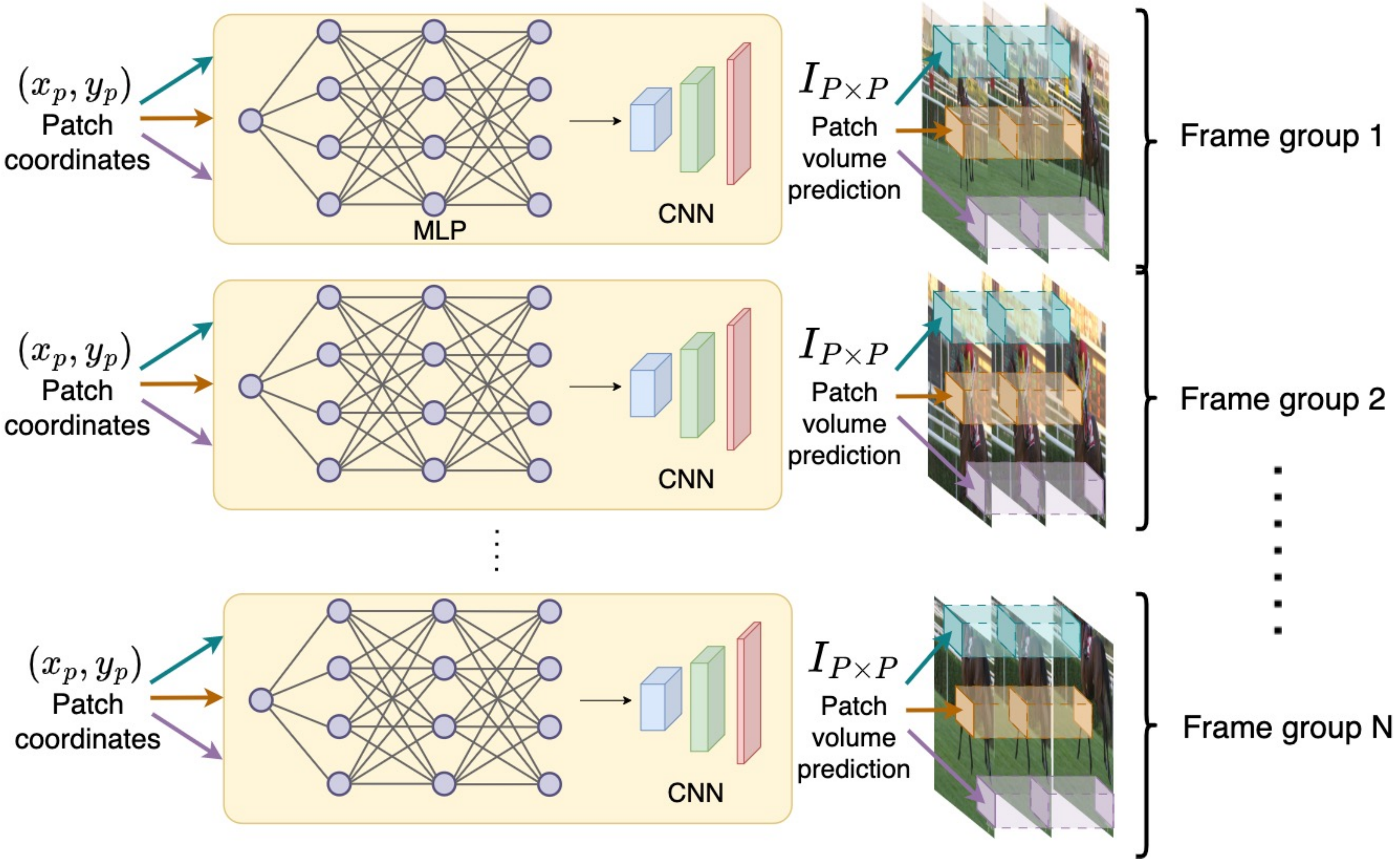
NIRVANA: Video INRs with Patchwise Prediction



Patch volume prediction

- Exploits spatial redundancy via patches.
- Scales well with increasing frame size/resolution.

NIRVANA: Video INRs with Patchwise Prediction



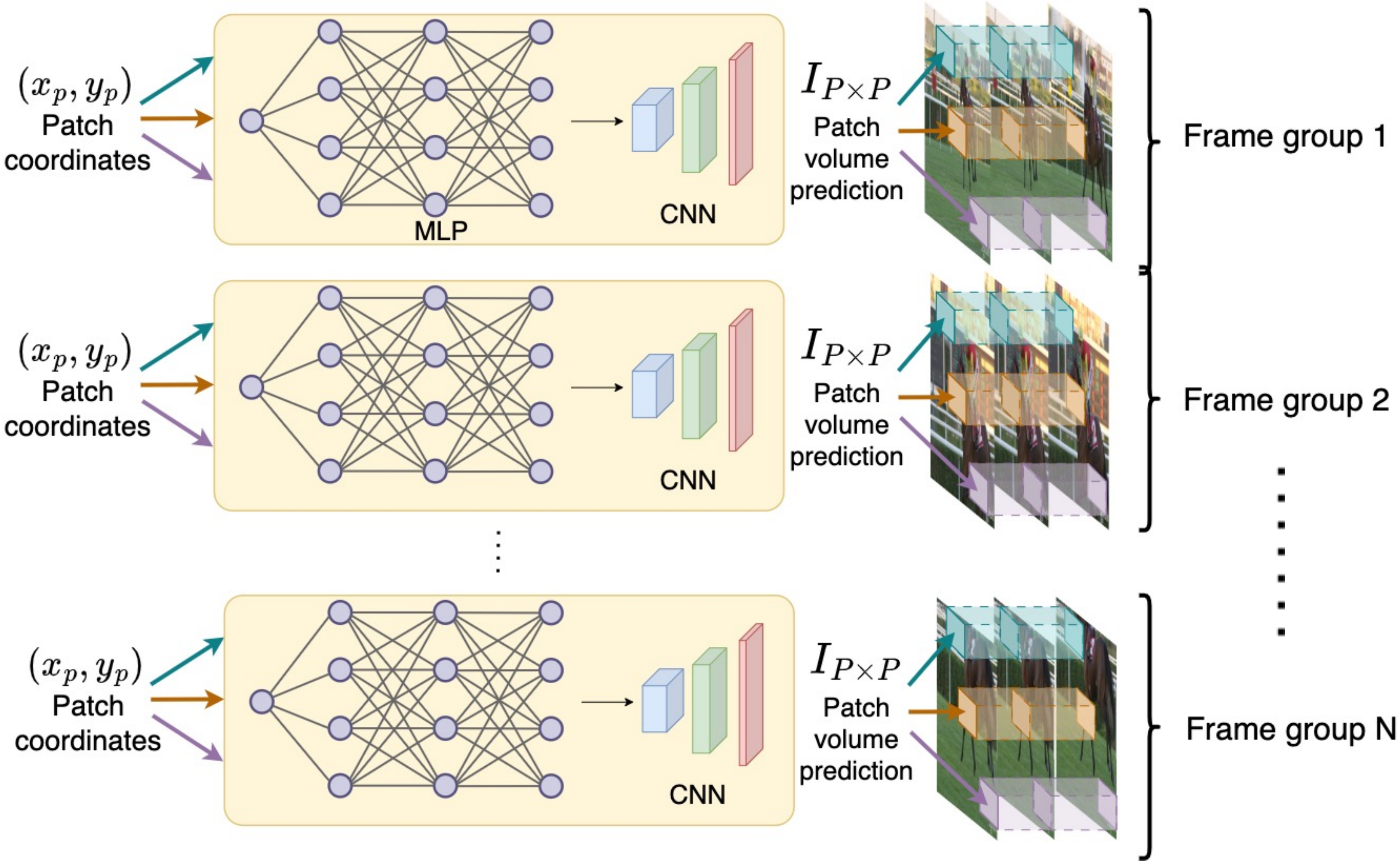
Patch volume prediction

- Exploits spatial redundancy via patches.
- Scales well with increasing frame size/resolution.

Groupwise prediction

- Exploits temporal redundancy via frame groups.

NIRVANA: Video INRs with Patchwise Prediction



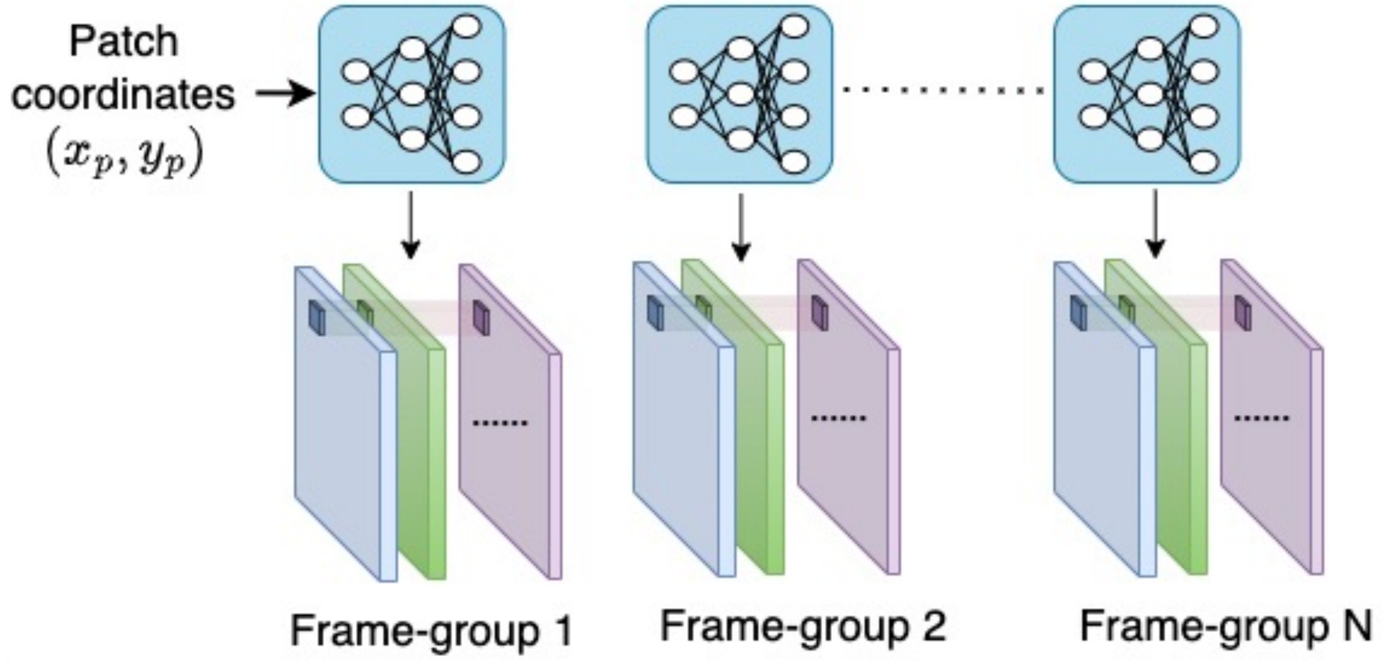
Patch volume prediction

- Exploits spatial redundancy via patches.
- Scales well with increasing frame size/resolution.

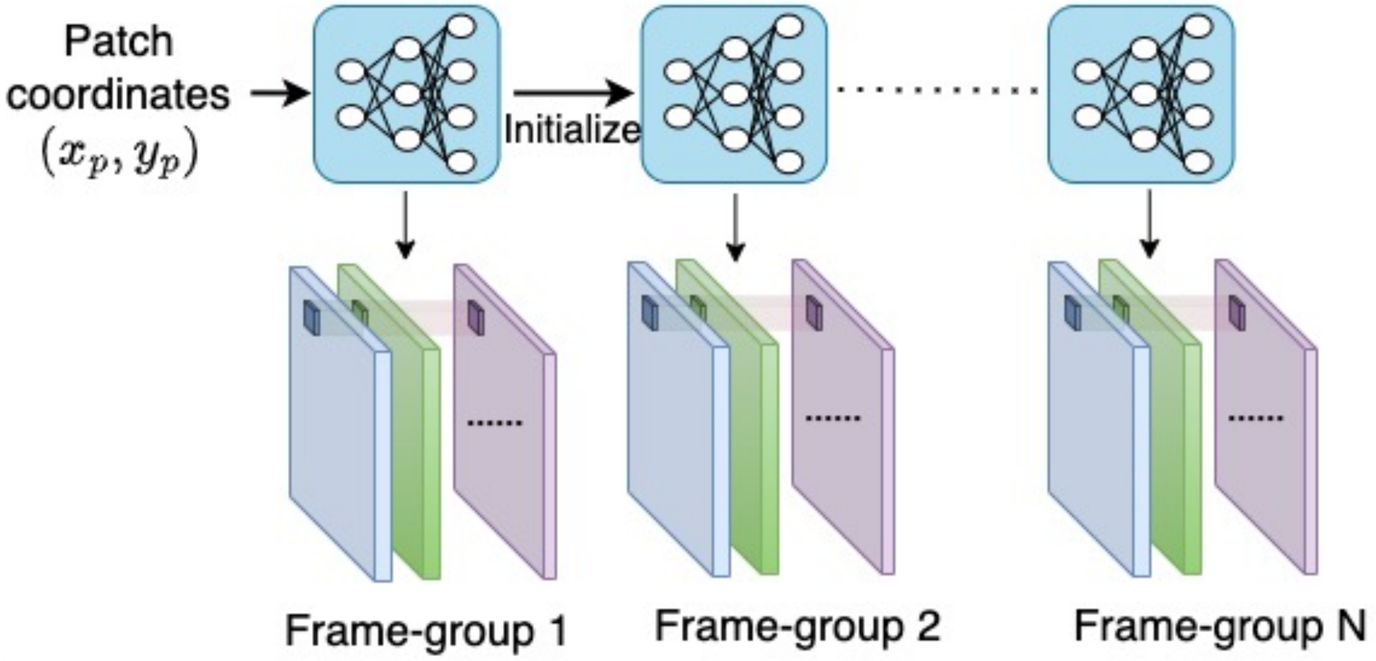
Groupwise prediction

- Exploits temporal redundancy via frame groups.
- Directly extends to arbitrarily long videos.

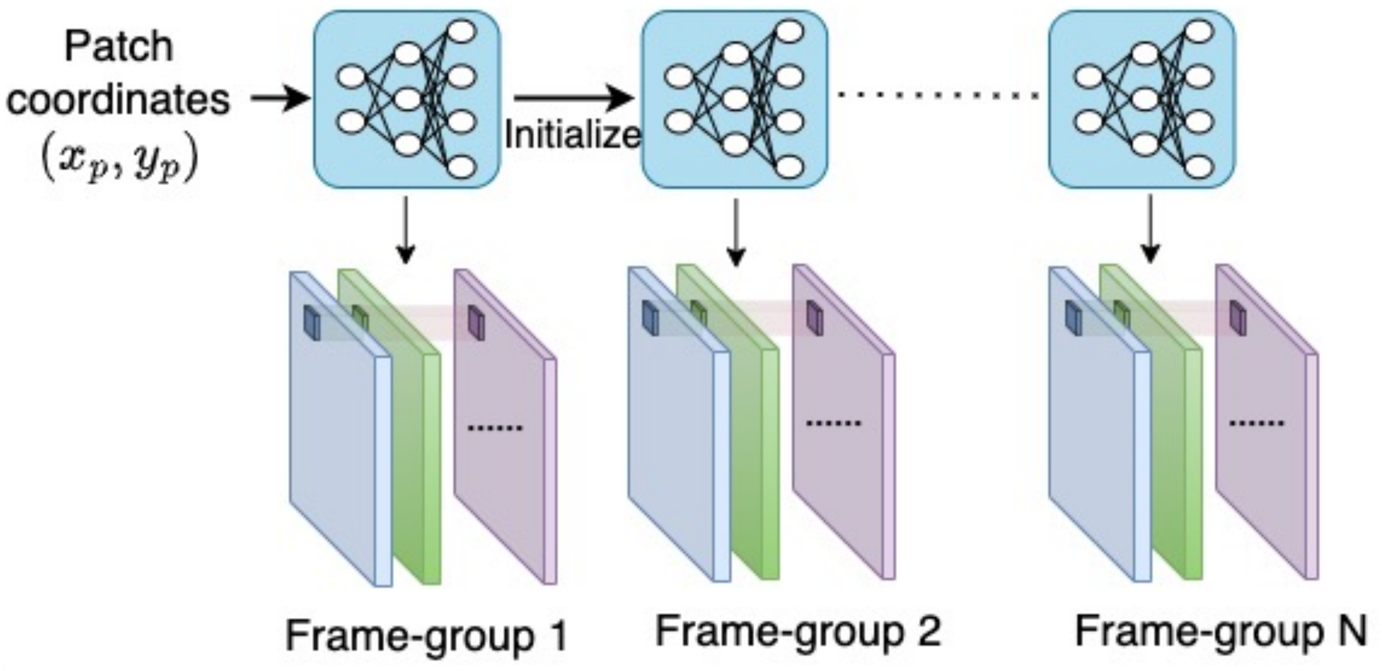
Autoregressive modeling



Autoregressive modeling

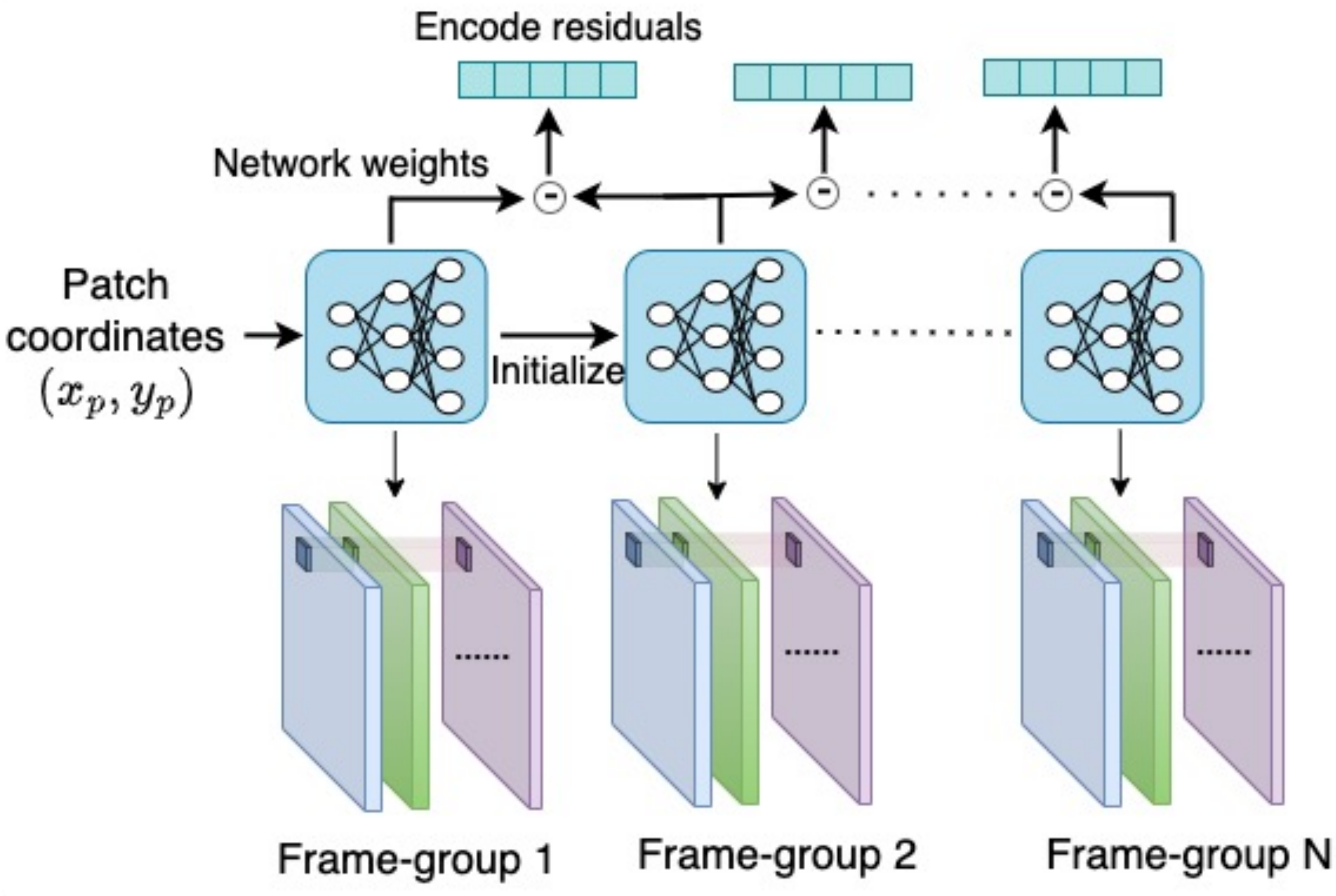


Autoregressive modeling



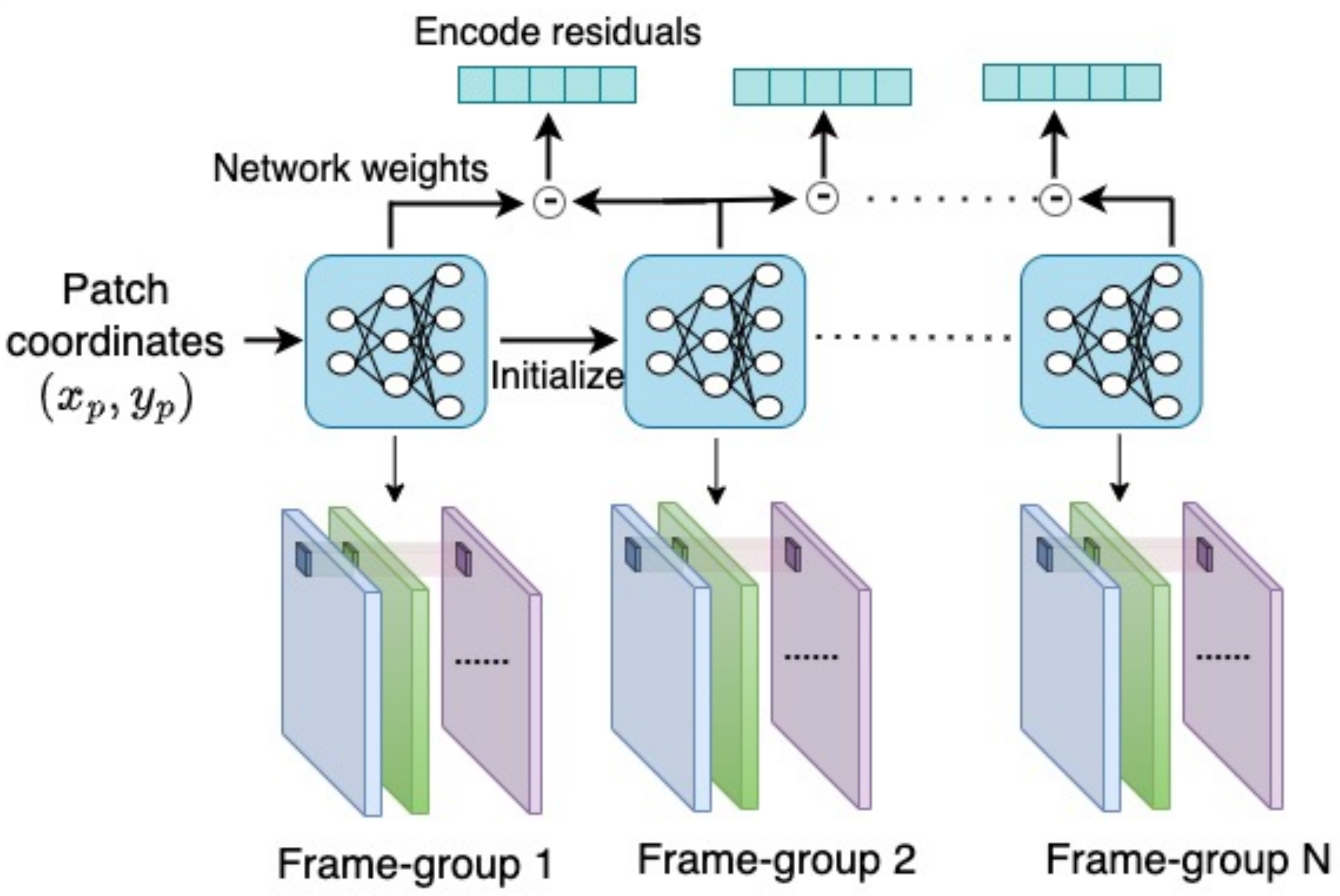
Faster training/convergence using previous weight initialization.

Autoregressive modeling



Faster training/convergence using previous weight initialization.

Autoregressive modeling

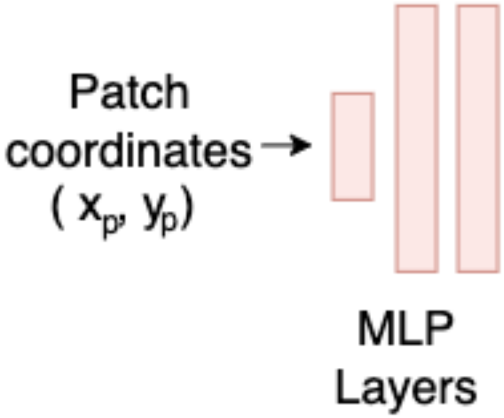


Faster training/convergence using previous weight initialization.

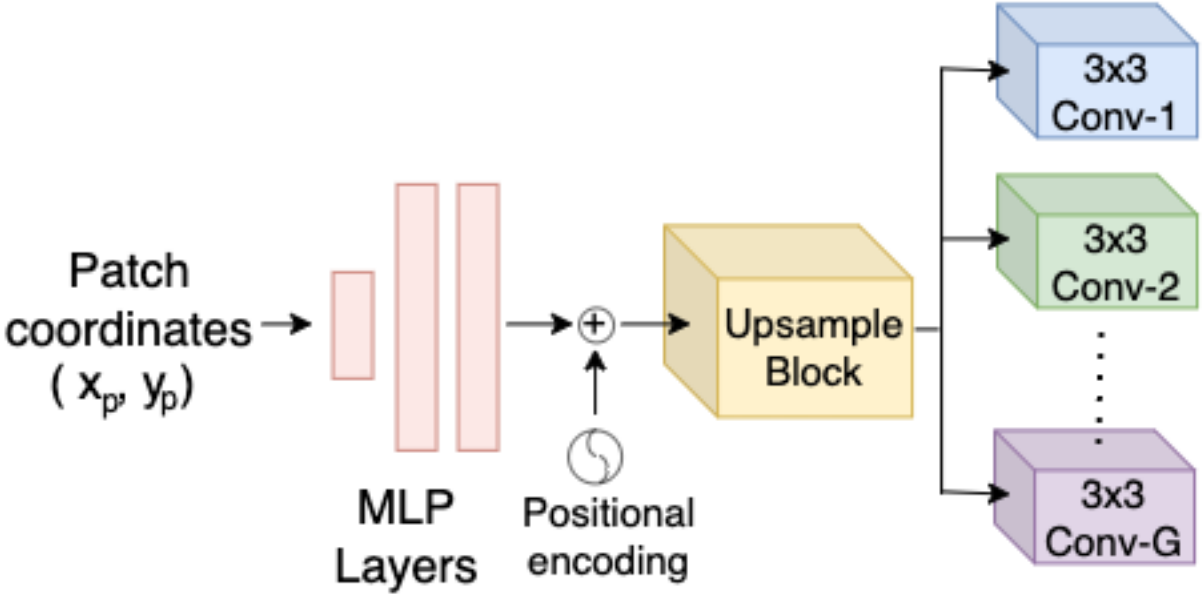
Lower network size by storing only sparse weight residuals.

Network architecture

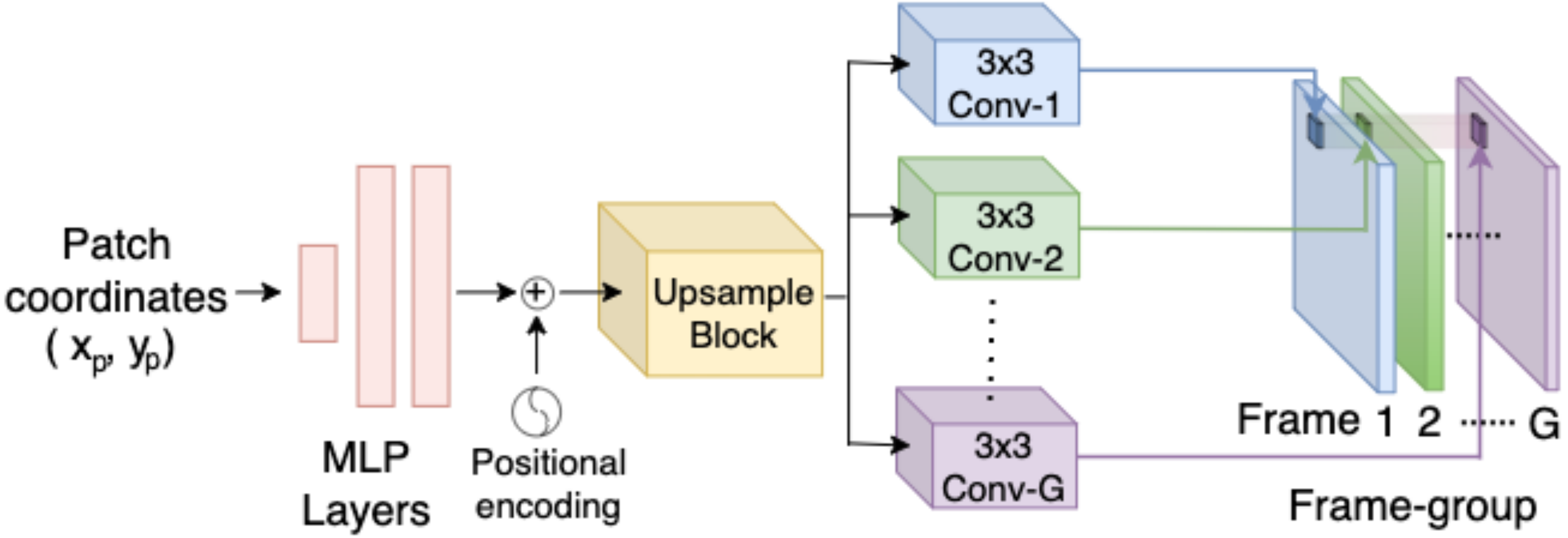
Network architecture



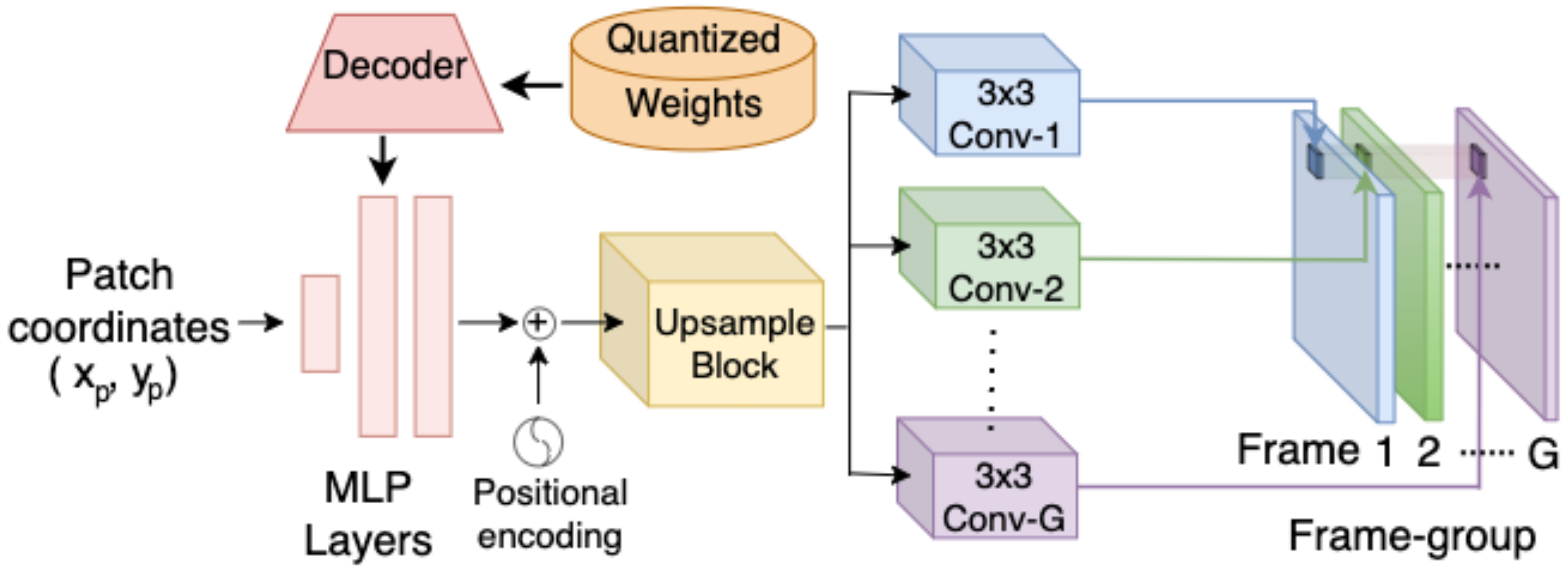
Network architecture



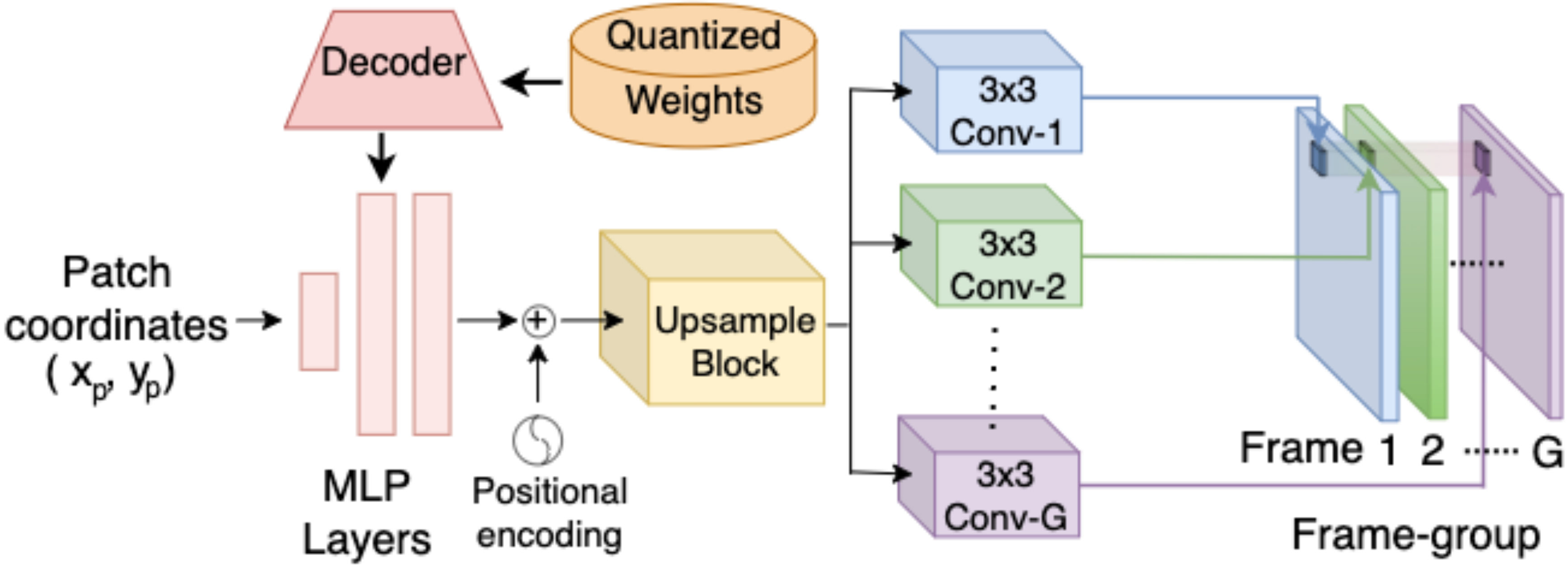
Network architecture



Network architecture

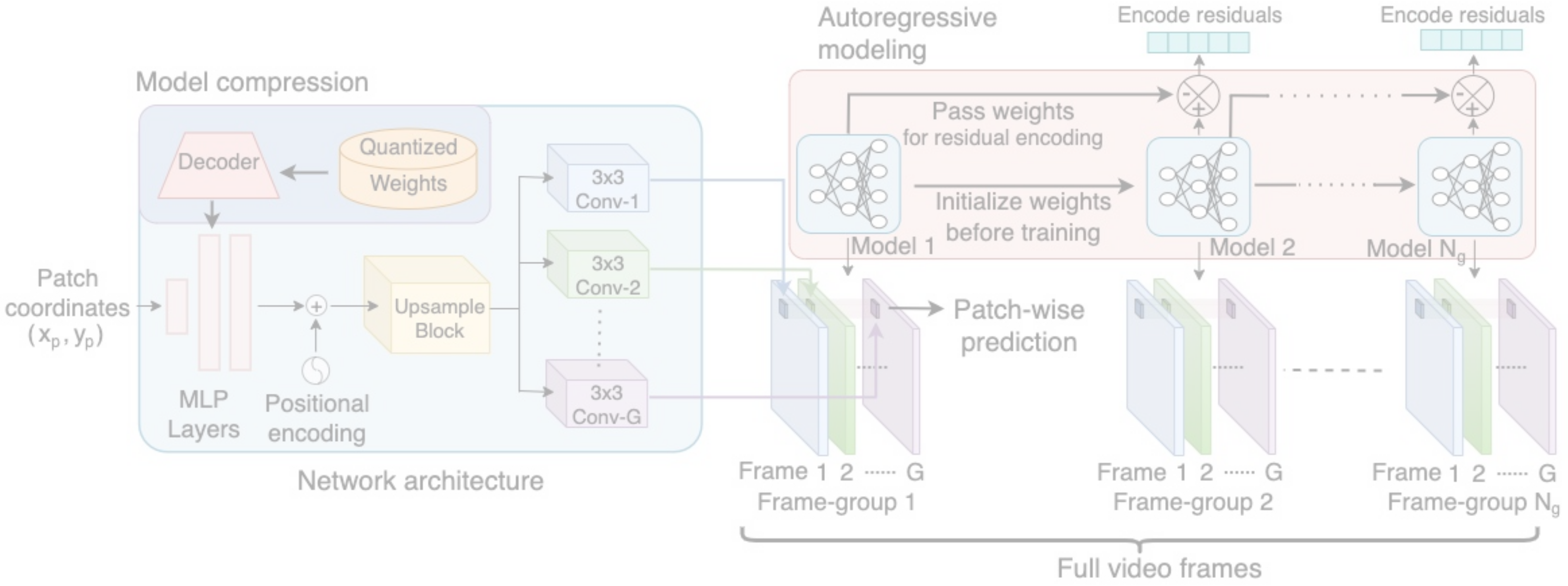


Network architecture

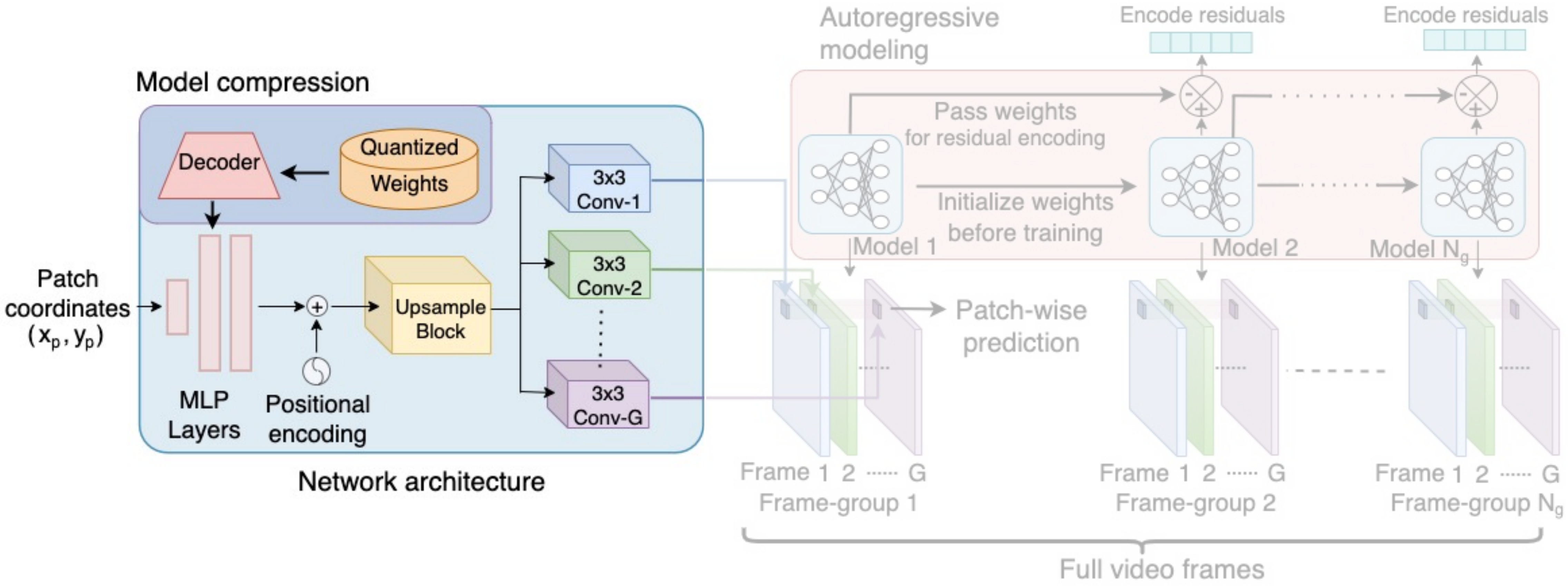


Optimize network weights θ_g with loss function: $min_{\theta_g}\|f_{\theta_g}(x_p, y_p) - I_g(x_p, y_p)\|_2^2$

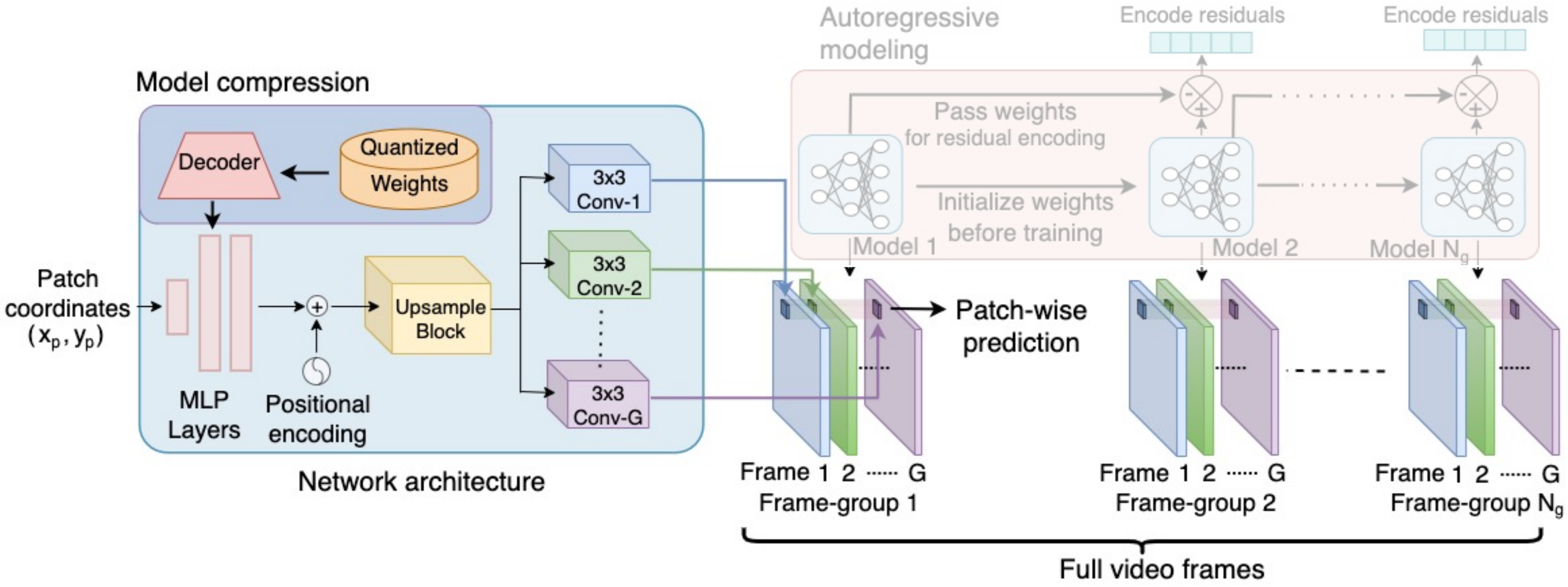
Patchwise autoregressive framework



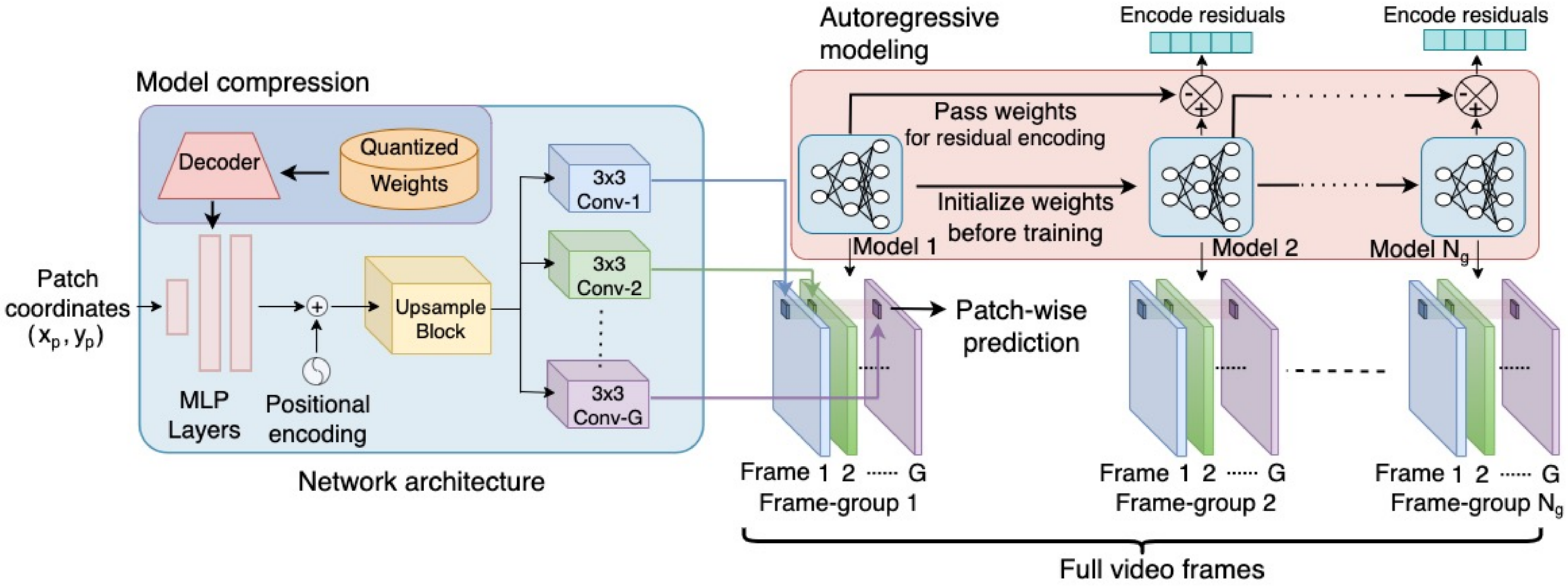
Patchwise autoregressive framework



Patchwise autoregressive framework

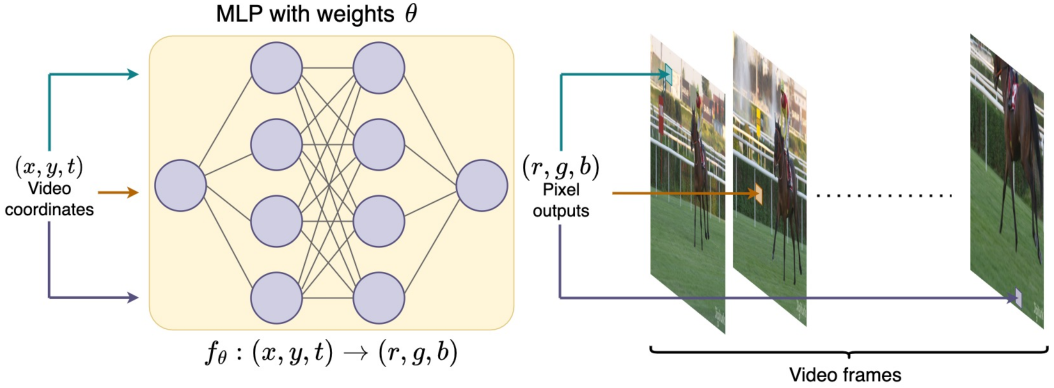


Patchwise autoregressive framework

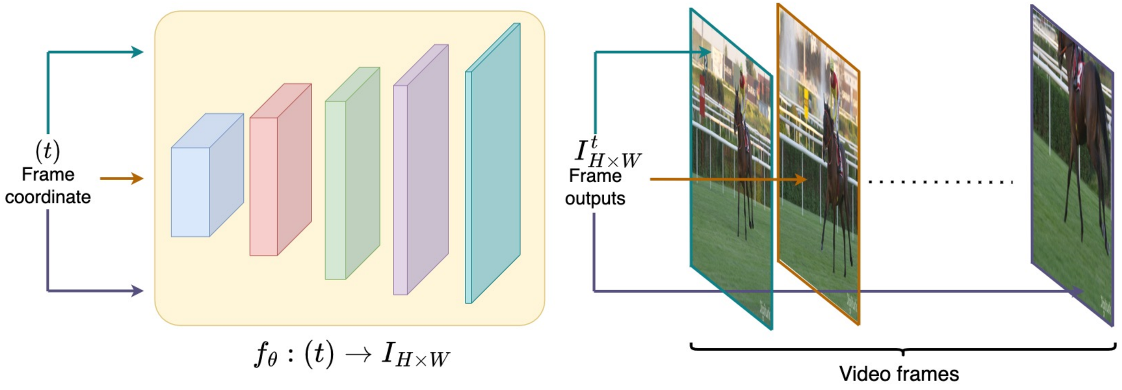


INRs for videos

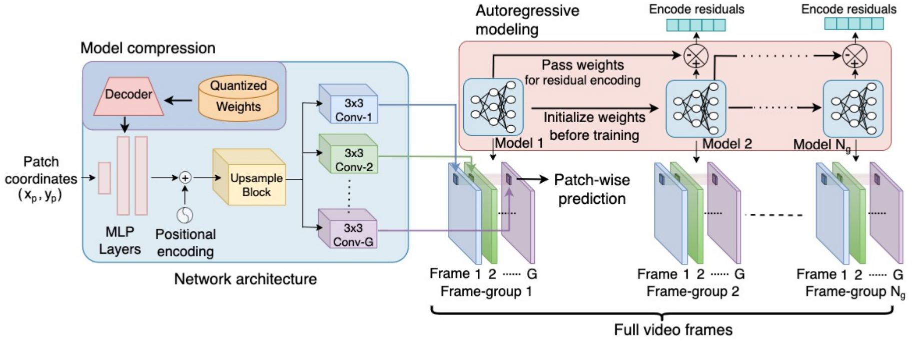
SIREN (pixelwise)



NeRV (Framewise)



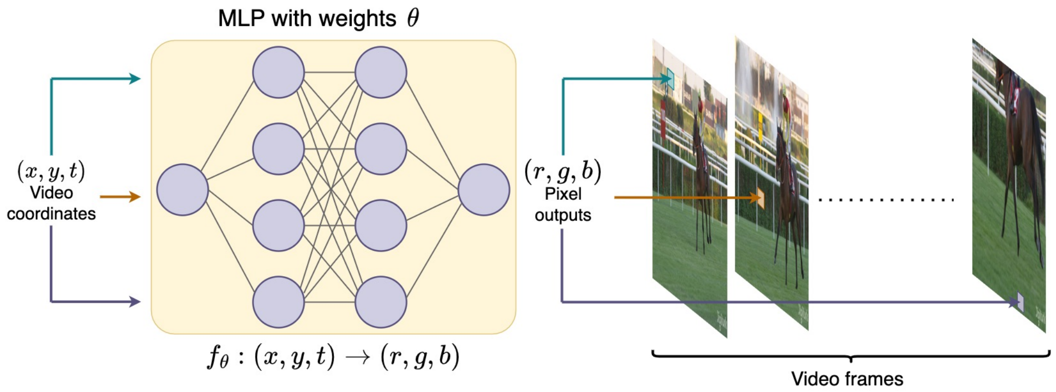
NIRVANA (patch groupwise)



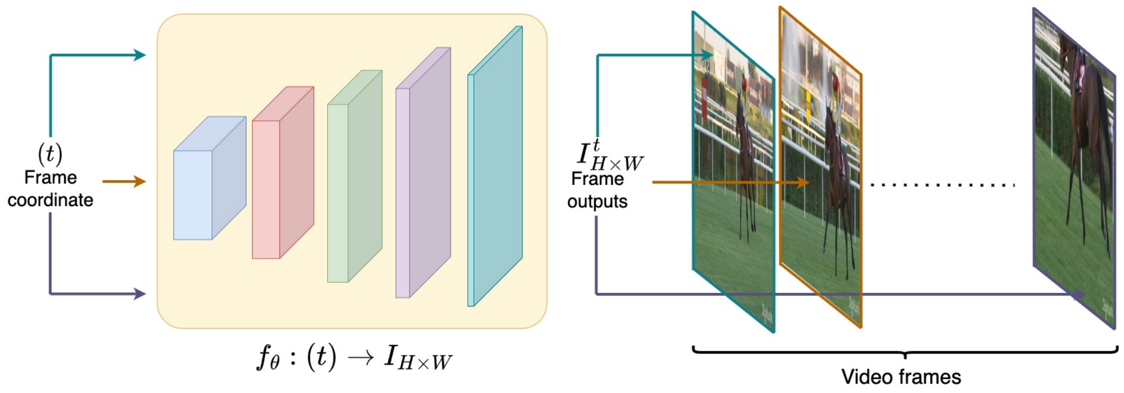
	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	❌	❌	❌	❌
NERV	✅	✅	❌	❌
NIRVANA				

INRs for videos

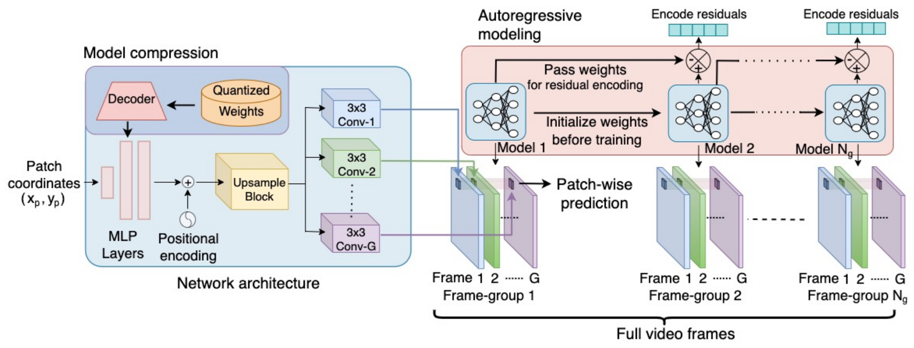
SIREN (pixelwise)



NeRV (Framewise)



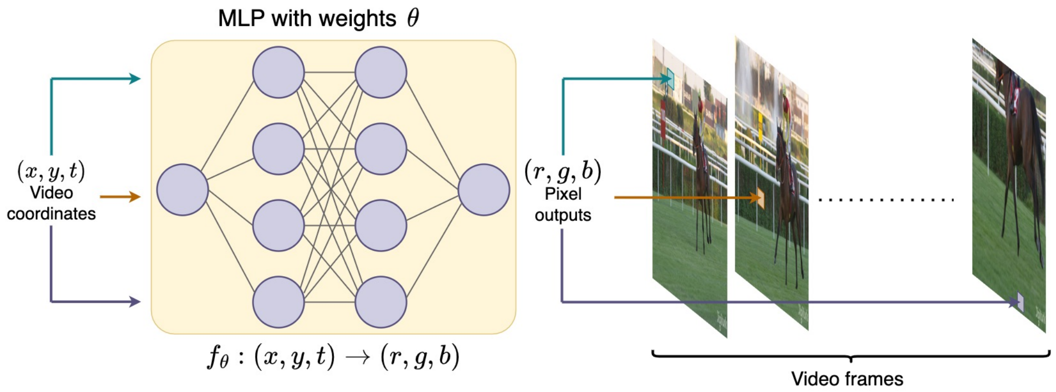
NIRVANA (patch groupwise)



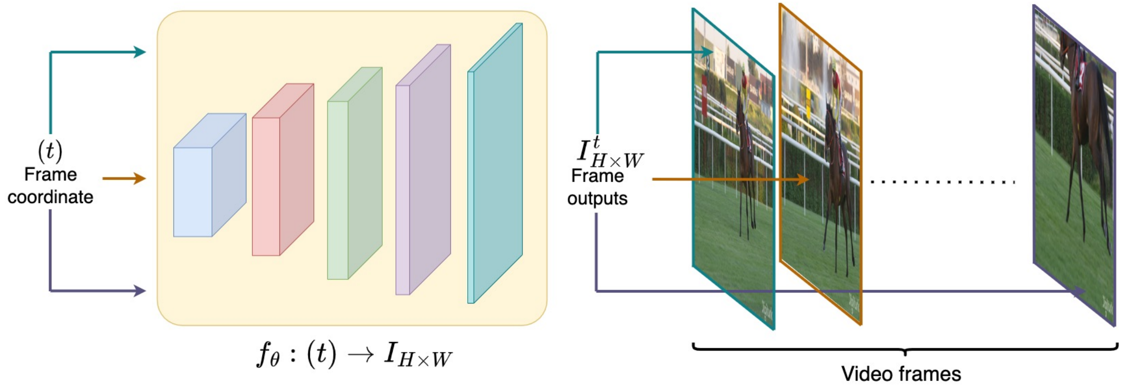
	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	✗	✗	✗	✗
NERV	✓	✓	✗	✗
NIRVANA	✓	✓		

INRs for videos

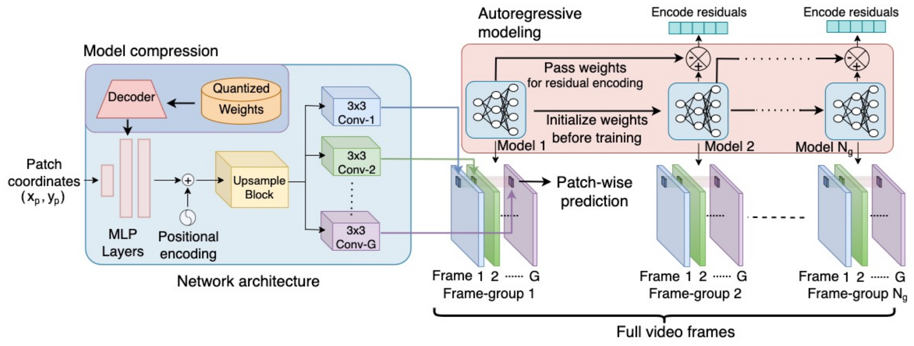
SIREN (pixelwise)



NeRV (Framewise)



NIRVANA (patch groupwise)

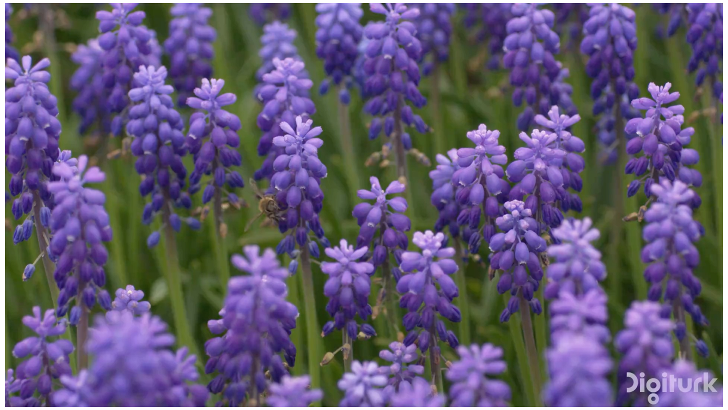


	Sharp reconstruction	Spatial redundancies	Temporal redundancies	Fast training
SIREN	✗	✗	✗	✗
NERV	✓	✓	✗	✗
NIRVANA	✓	✓	✓	✓

Datasets

UVG-HD: 7 HD videos (1920x1080) at 120 FPS and mostly 600 frames

UVG-4K: 2x upsampling of UVG-HD at 4K resolution (3840x2160)






Comparison with related work

Dataset	Method	Encoding Time (Hours) ↓	Decoding Speed (FPS) ↑	PSNR ↑	BPP ↓
UVG-HD	SIREN	∼30	15.62	27.20	**0.28**
	NIRVANA (Ours)	**5.44**	**87.91**	**34.71**	0.32
	NeRV	∼80	11.01	37.36	0.92
	NIRVANA (Ours)	**6.71**	**65.42**	**37.70**	**0.86**
UVG-4K	NeRV	∼134	8.27	**35.24**	0.28
	NIRVANA (Ours)	**20.89**	**50.83**	35.18	**0.27**

Comparison with related work

Dataset	Method	Encoding Time (Hours) ↓	Decoding Speed (FPS) ↑	PSNR ↑	BPP ↓
UVG-HD	SIREN	$\sim$30	15.62	27.20	**0.28**
	NIRVANA (Ours)	**5.44**	**87.91**	**34.71**	0.32
	NeRV	$\sim$80	11.01	37.36	0.92
	NIRVANA (Ours)	**6.71**	**65.42**	**37.70**	**0.86**
UVG-4K	NeRV	$\sim$134	8.27	**35.24**	0.28
	NIRVANA (Ours)	**20.89**	**50.83**	35.18	**0.27**

+7dB improvement in PSNR over SIREN at similar bits per pixel (BPP)

Faster encoding/decoding speed

Dataset	Method	Encoding Time (Hours) ↓	Decoding Speed (FPS) ↑	PSNR ↑	BPP ↓
UVG-HD	SIREN	~30	15.62	27.20	**0.28**
	NIRVANA (Ours)	**5.44**	**87.91**	**34.71**	0.32
	NeRV	~80	11.01	37.36	0.92
	NIRVANA (Ours)	**6.71**	**65.42**	**37.70**	**0.86**
UVG-4K	NeRV	~134	8.27	**35.24**	0.28
	NIRVANA (Ours)	**20.89**	**50.83**	35.18	**0.27**

+7dB improvement in PSNR over SIREN at similar bits per pixel (BPP)

5x faster encoding and 6x faster decoding

Faster encoding/decoding speed

Dataset	Method	Encoding Time (Hours) ↓	Decoding Speed (FPS) ↑	PSNR ↑	BPP ↓
UVG-HD	SIREN	∼30	15.62	27.20	**0.28**
	NIRVANA (Ours)	**5.44**	**87.91**	**34.71**	0.32
	NeRV	∼80	11.01	37.36	0.92
	NIRVANA (Ours)	**6.71**	**65.42**	**37.70**	**0.86**
UVG-4K	NeRV	∼134	8.27	**35.24**	0.28
	NIRVANA (Ours)	**20.89**	**50.83**	35.18	**0.27**

12x lower encoding time and 6x faster decoding than NeRV at similar PSNR and BPP

Scalability to 4K videos

Dataset	Method	Encoding Time (Hours) ↓	Decoding Speed (FPS) ↑	PSNR ↑	BPP ↓
UVG-HD	SIREN	∼30	15.62	27.20	**0.28**
	NIRVANA (Ours)	**5.44**	**87.91**	**34.71**	0.32
	NeRV	∼80	11.01	37.36	0.92
	NIRVANA (Ours)	**6.71**	**65.42**	**37.70**	**0.86**
UVG-4K	NeRV	∼134	8.27	**35.24**	0.28
	NIRVANA (Ours)	**20.89**	**50.83**	35.18	**0.27**

Scales better to 4K videos (2160x3840) with 6x faster encoding and decoding at similar PSNR, BPP

Scalability to longer videos

Num Frames	Method	Encoding Time (Hours) $\downarrow$	PSNR $\uparrow$	BPP $\downarrow$
2000	NeRV	84.44	33.38	0.22
	NIRVANA (Ours)	20.85	35.43	0.62
3000	NeRV	134.58	31.6	0.16
	NIRVANA (Ours)	31.37	35.21	0.64
4000	NeRV	190.30	30.53	0.12
	NIRVANA (Ours)	41.84	35.15	0.65

Scalability to longer videos

Num Frames	Method	Encoding Time (Hours) ↓	PSNR ↑	BPP ↓
2000	NeRV	84.44	33.38	0.22
	NIRVANA (Ours)	20.85	35.43	0.62
3000	NeRV	134.58	31.6	0.16
	NIRVANA (Ours)	31.37	35.21	0.64
4000	NeRV	190.30	30.53	0.12
	NIRVANA (Ours)	41.84	35.15	0.65

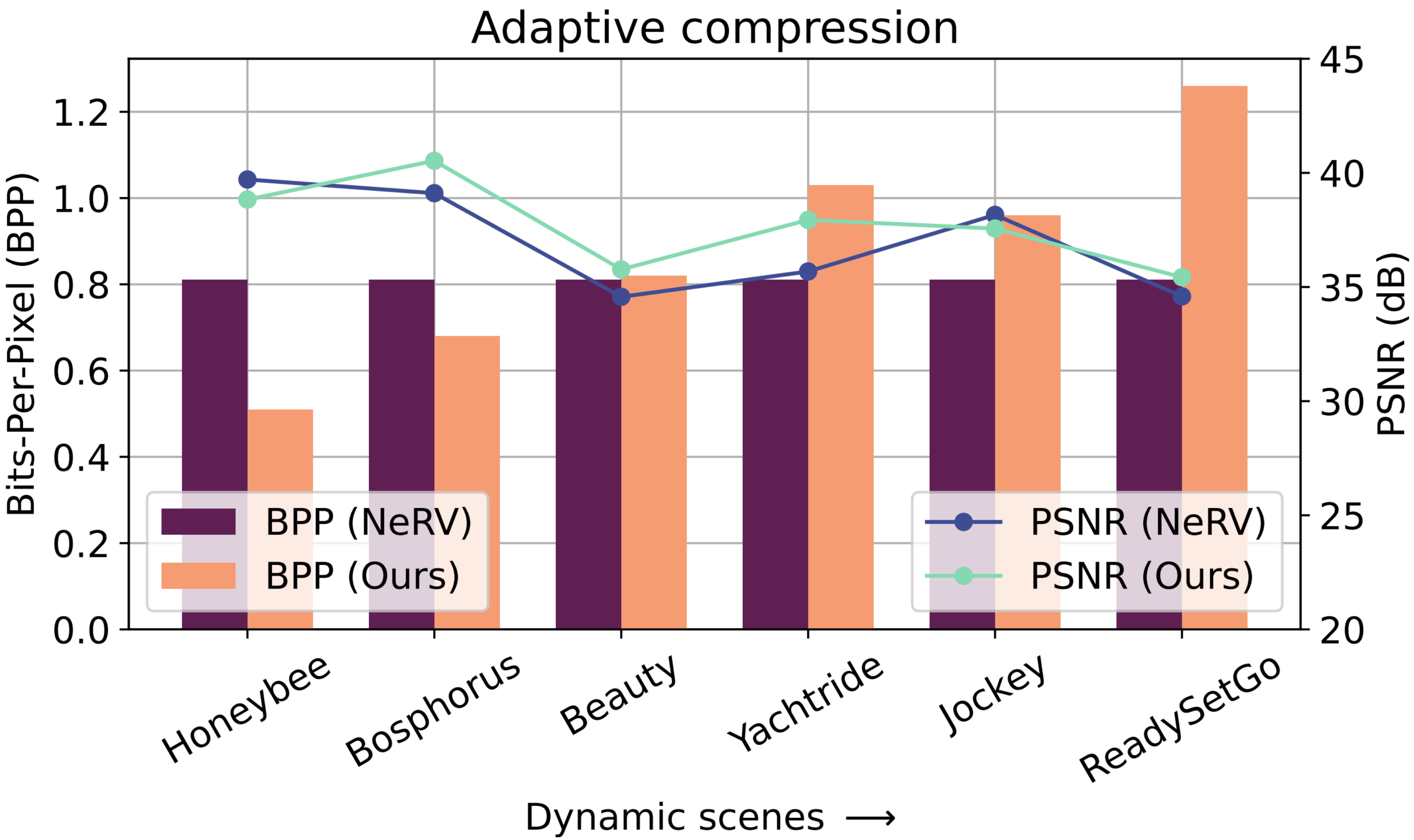
NeRV degrades in performance for longer videos due to fixed size model

Scalability to longer videos

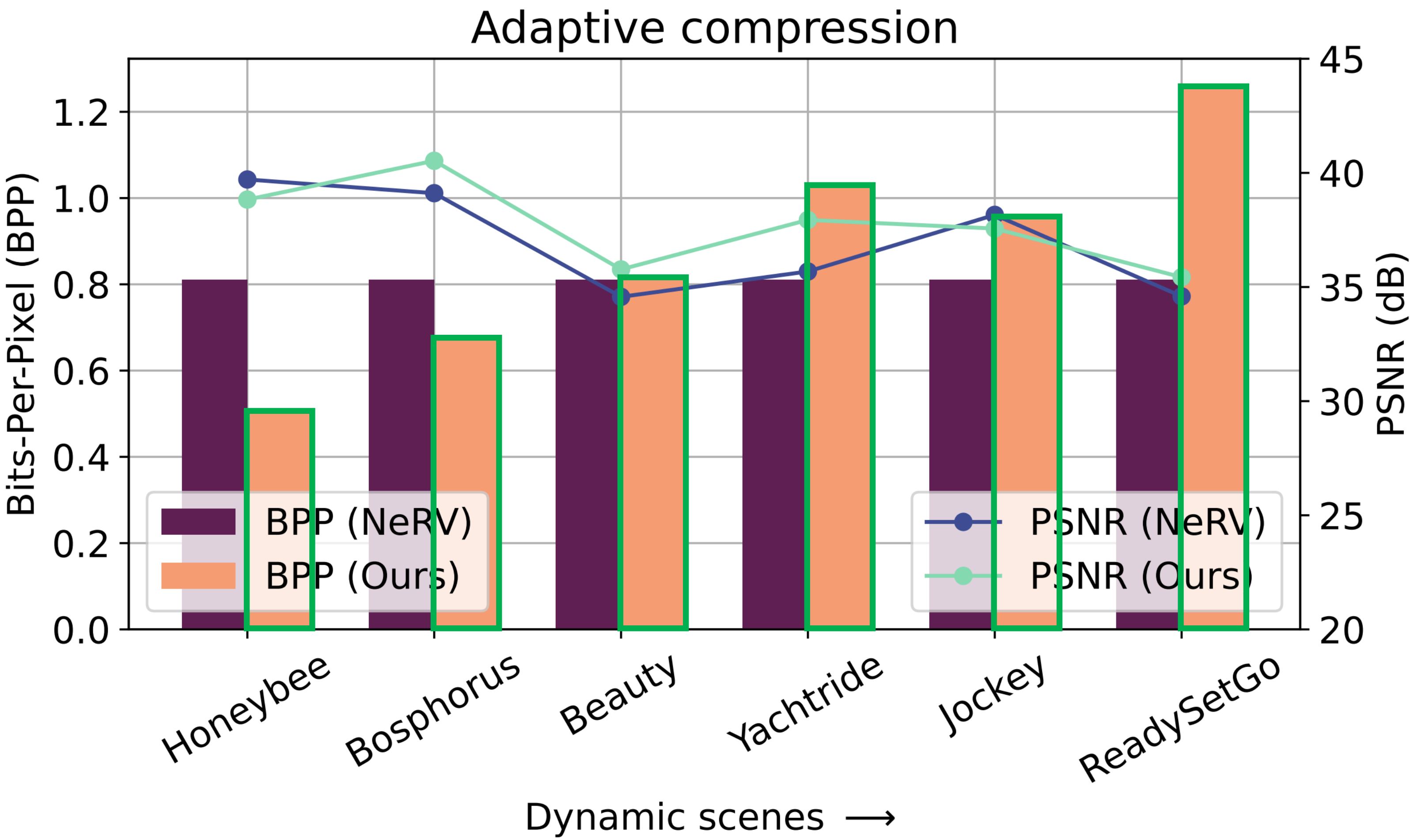
Num Frames	Method	Encoding Time (Hours) ↓	PSNR ↑	BPP ↓
2000	NeRV	84.44	33.38	0.22
	NIRVANA (Ours)	20.85	35.43	0.62
3000	NeRV	134.58	31.6	0.16
	NIRVANA (Ours)	31.37	35.21	0.64
4000	NeRV	190.30	30.53	0.12
	NIRVANA (Ours)	41.84	35.15	0.65

NIRVANA maintains performance with longer videos due to autoregressive modeling

Video content adaptability



Video content adaptability

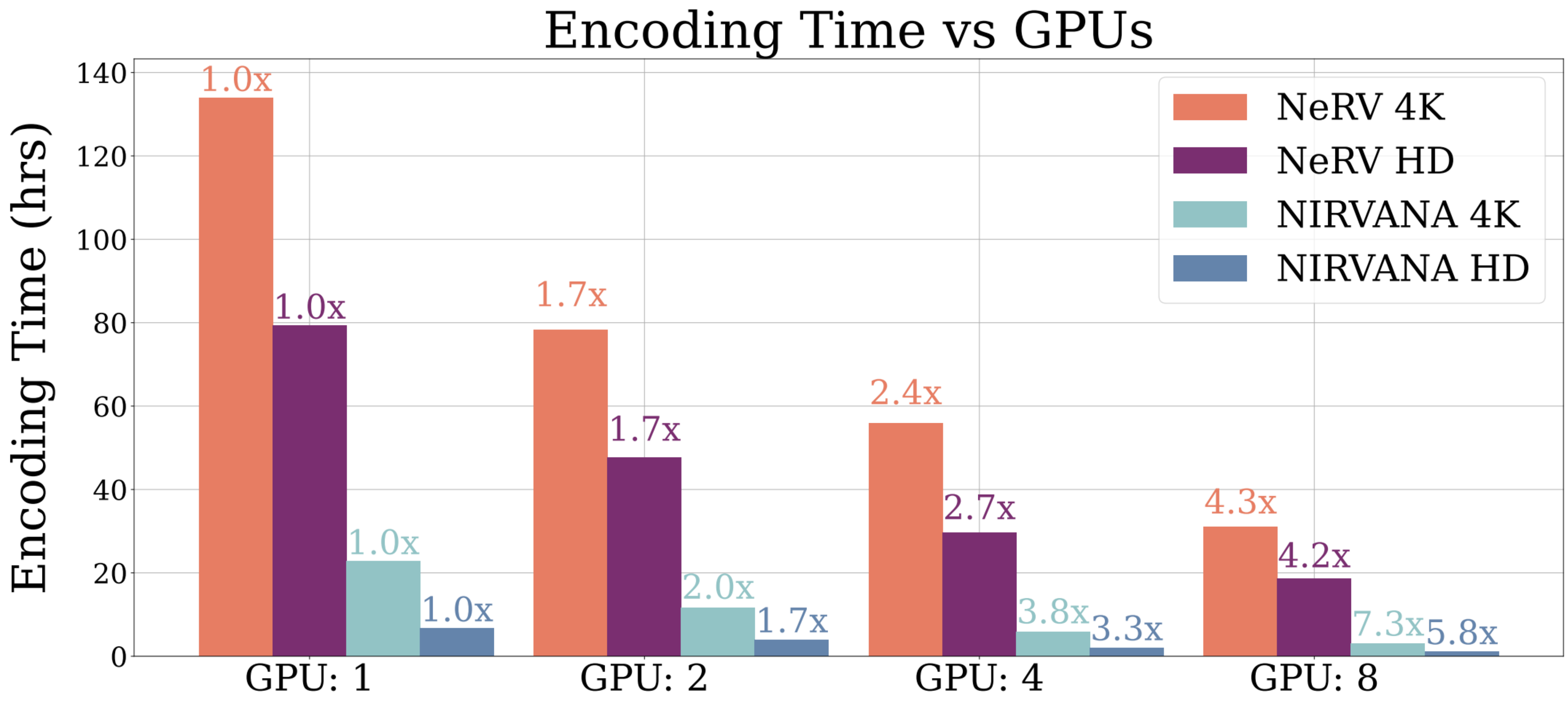


NIRVANA adapts to video content with static scenes requiring lesser BPP

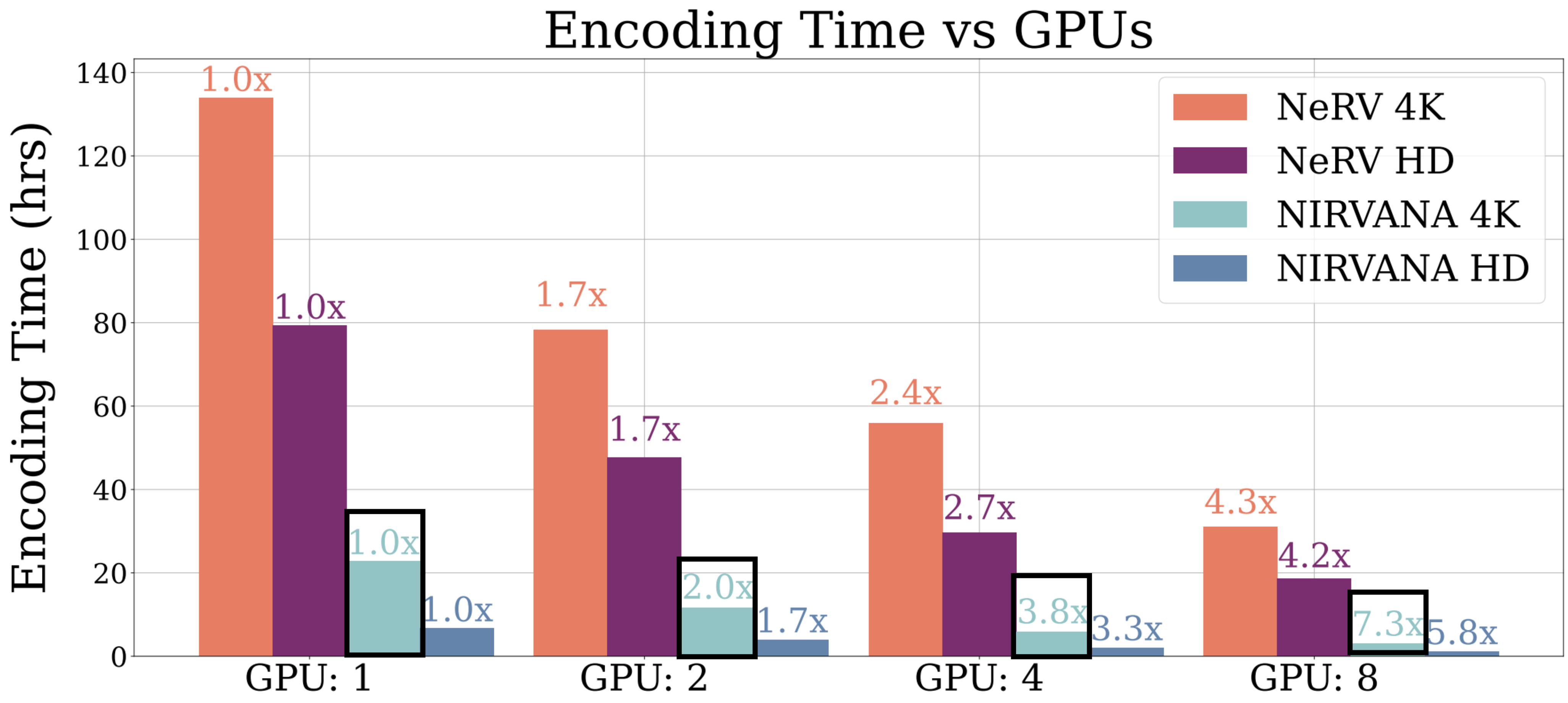
Qualitative comparisons

NIRVANA achieves better reconstructions preserving finer details in the video.

GPU scalability



GPU scalability



NIRVANA scales almost linearly with increasing GPUs in terms of encoding speed

Conclusion

- We present an autoregressive patchwise modeling approach to video INRs which exploits the spatial temporal redundancies present.

Project page:

Conclusion

- We present an autoregressive patchwise modeling approach to video INRs which exploits the spatial temporal redundancies present.
- Our approach has much faster training and inference speeds than prior works while maintaining similar reconstruction performance and model size.

Project page:

Conclusion

- We present an autoregressive patchwise modeling approach to video INRs which exploits the spatiotemporal redundancies present.
- Our approach has much faster training and inference speeds than prior works while maintaining similar reconstruction performance and model size.
- Our approach scales well to varying video resolutions and durations while also adapting to the video content.

Project page:

Conclusion

- We present an autoregressive patchwise modeling approach to video INRs which exploits the spatiotemporal redundancies present.
- Our approach has much faster training and inference speeds than prior works while maintaining similar reconstruction performance and model size.
- Our approach scales well to varying video resolutions and durations while also adapting to the video content.

Visit us at poster #194 on Wednesday evening session (4:30 PM – 6:30 PM) at CVPR 2023!

Project page: