GENERATIONS / VANCOUVER 12-16 AUGUST
SIGGRAPH2018

Real-Time Neural Rendering In Image Space

Anton Kaplanyan, Facebook Reality Labs

Why Neural Rendering at Facebook?

Facebook Reality Labs

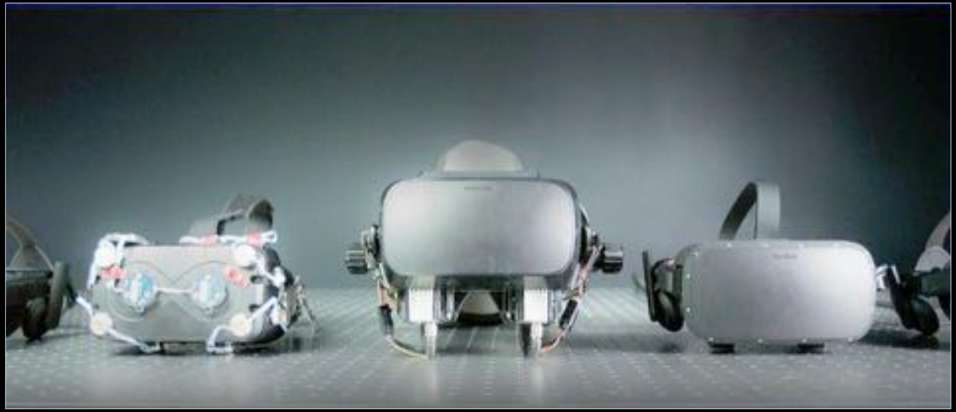
Former Oculus Research, located in Redmond, WA

Work on consumer VR/AR/MR

Graphics team: next generation graphics for VR/AR

- real-time ray casting
- machine learning
- perceptual rendering
- metaverse ecosystem

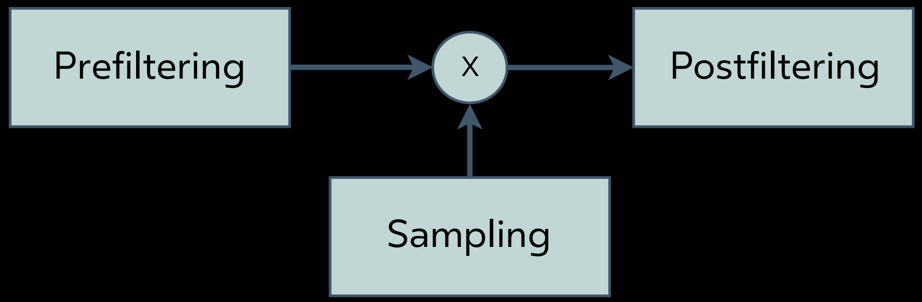
...next generation rendering for head-mounted displays

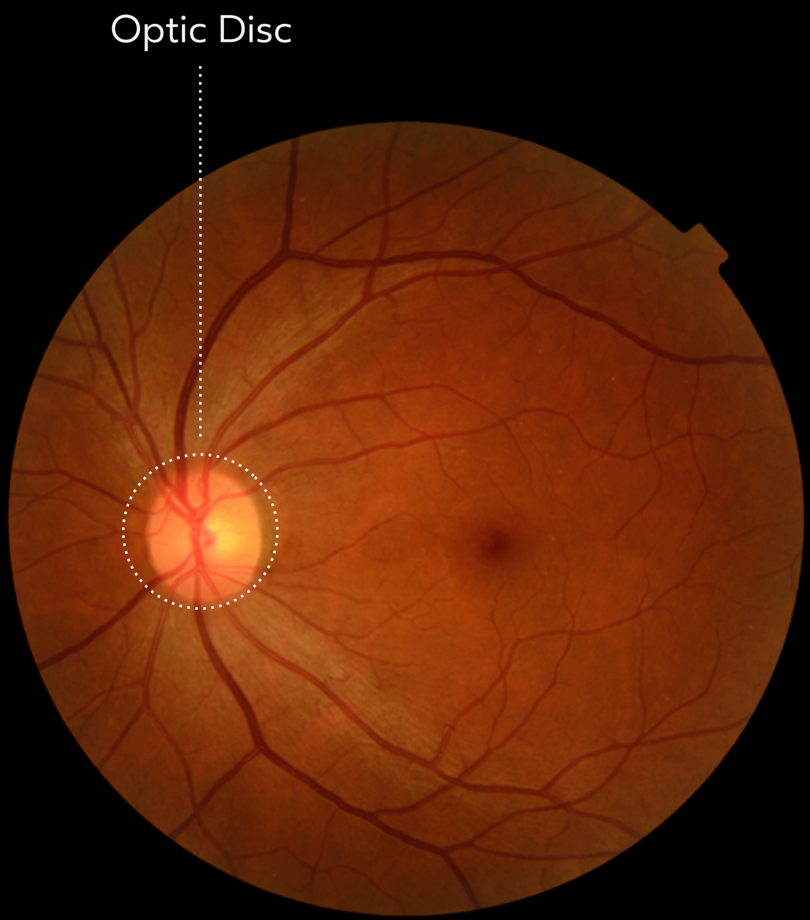


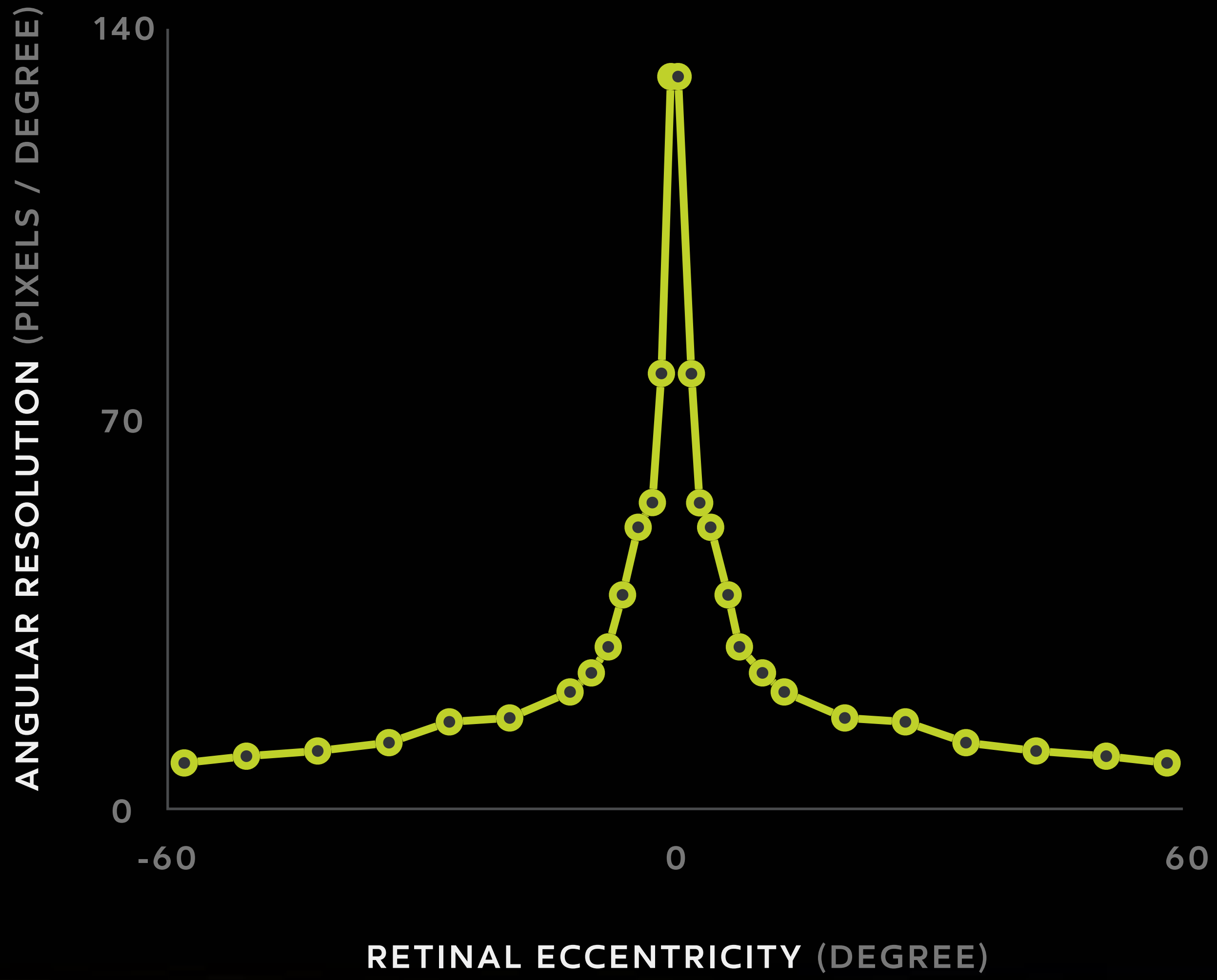
facebook Reality Labs

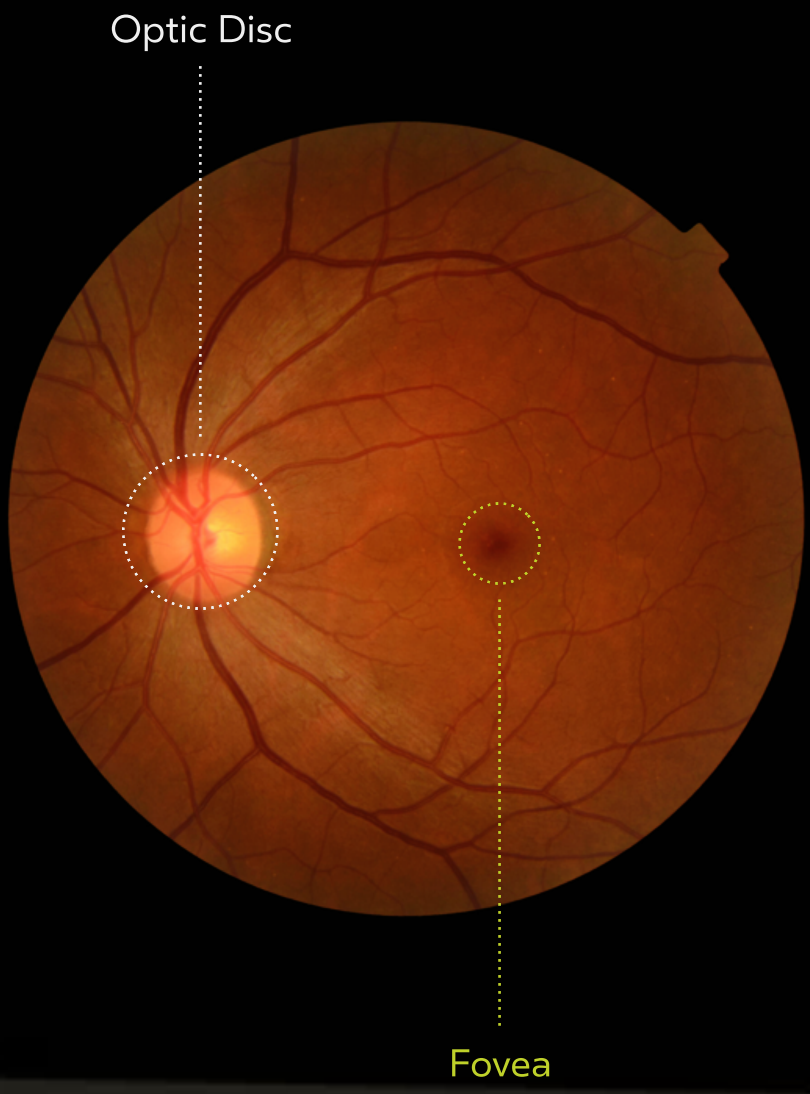
Why Neural Rendering?

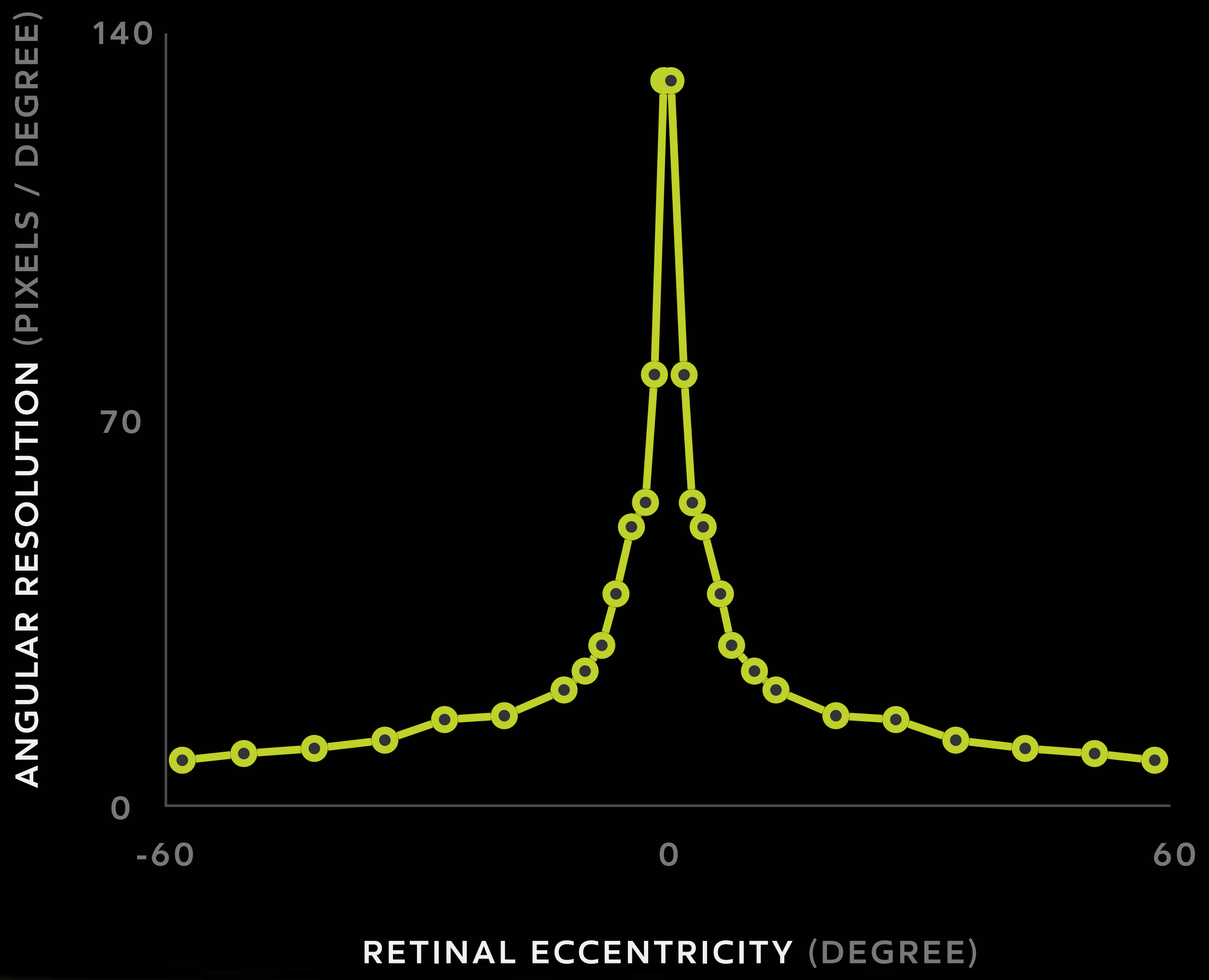
- Rendering as signal processing
  - Prefiltering
  - Sampling
  - Postfiltering
- Local approximations
  - New material models
  - Sampling and variance reduction
  - Texture compression
- Content creation
  - Texture synthesis




Optic Disc

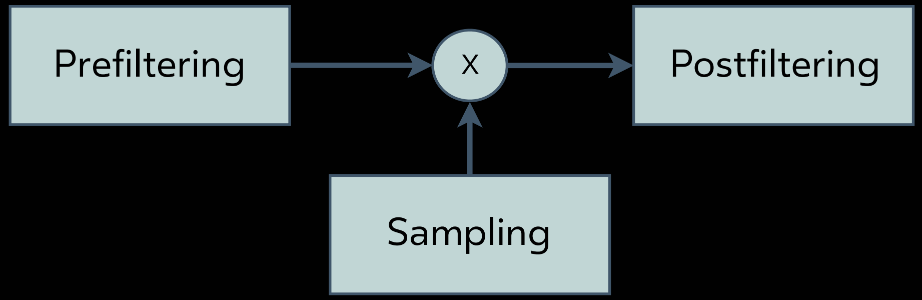

ANGULAR RESOLUTION (PIXELS / DEGREE)
140
70
0
-60
0
60
RETINAL ECCENTRICITY (DEGREE)

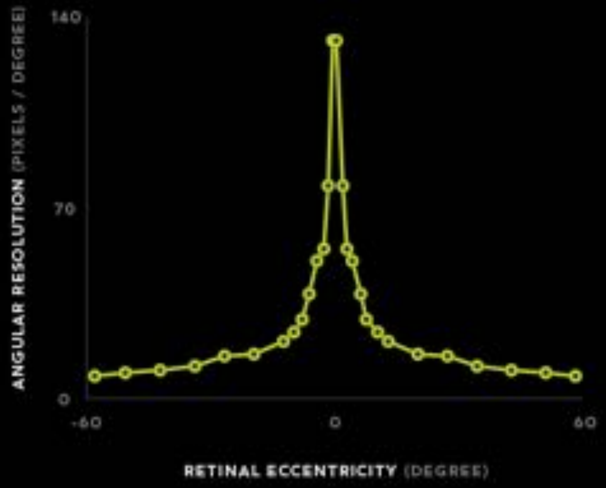

Optic Disc
Fovea


ANGULAR RESOLUTION (PIXELS / DEGREE)
140
70
0
-60
0
60
RETINAL ECCENTRICITY (DEGREE)

Why Neural Rendering?

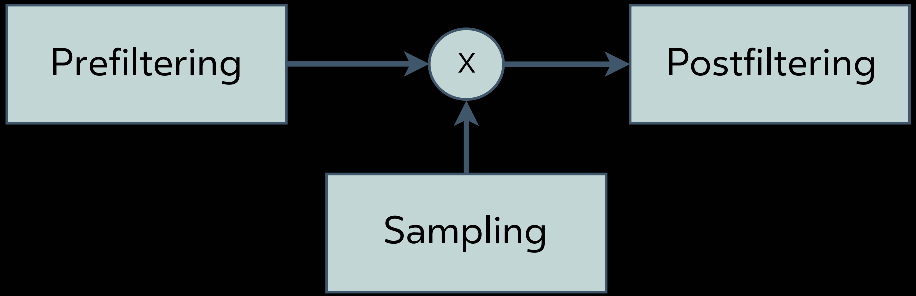
- Signal processing
  - Prefiltering
  - Sampling
  - Postfiltering
- Perceptual imagery

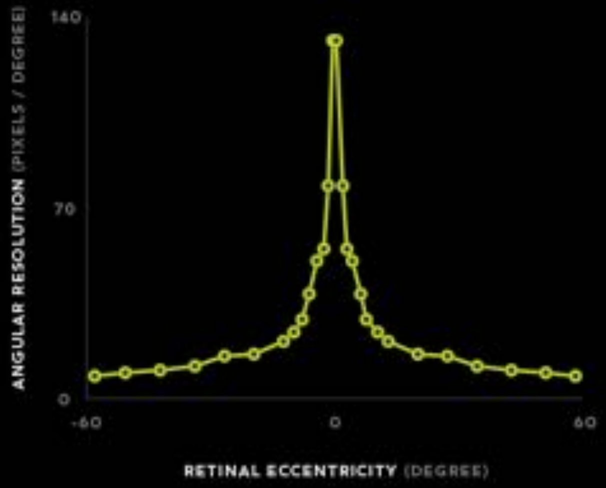




Why Neural Rendering?

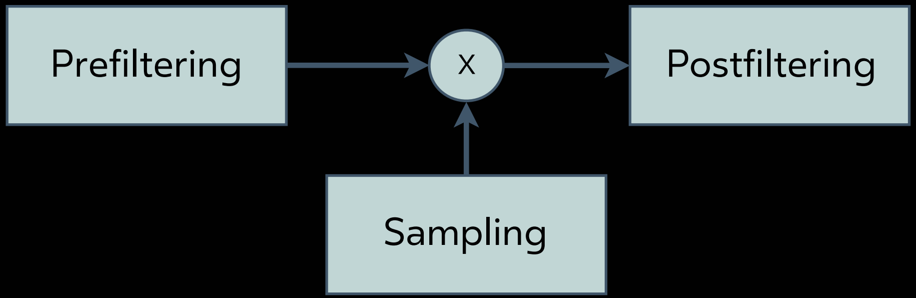
- Signal processing
  - Prefiltering
  - Sampling
  - Postfiltering
- Perceptual imagery
  - Foveation and peripheral degradation

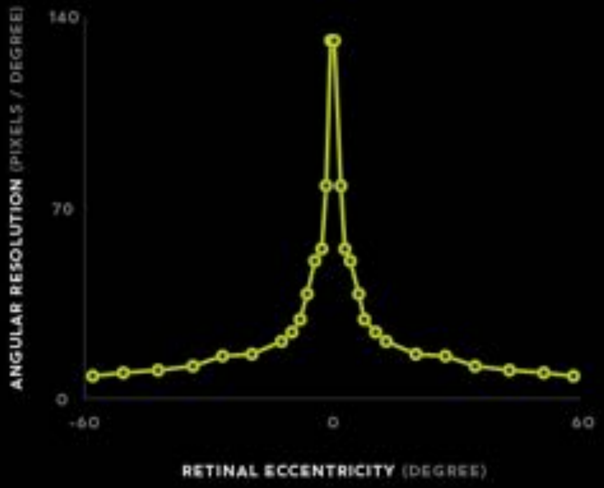




Why Neural Rendering?

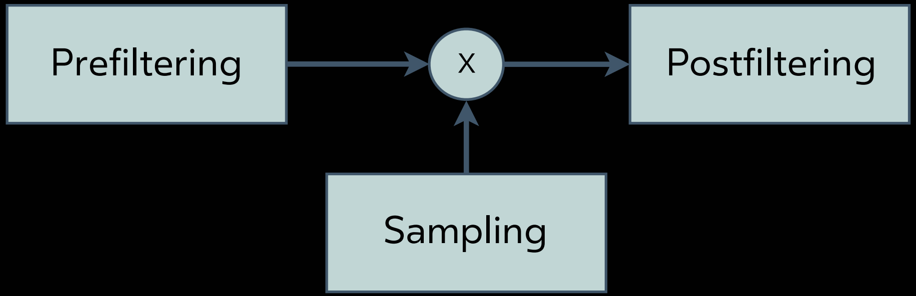
- Signal processing
  - Prefiltering
  - Sampling
  - Postfiltering
- Perceptual imagery
  - Foveation and peripheral degradation
  - Saliency and attention

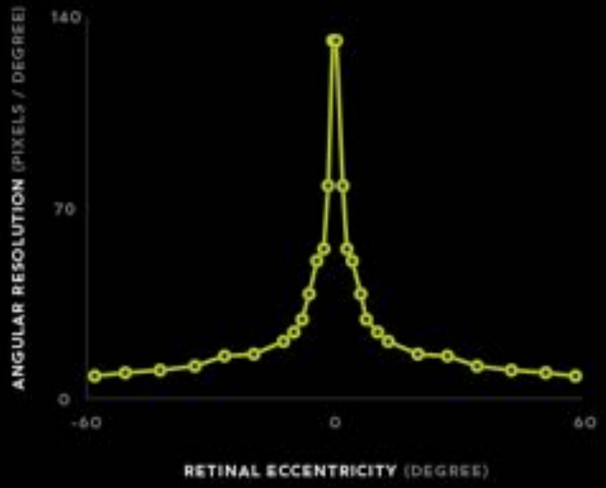




Why Neural Rendering?

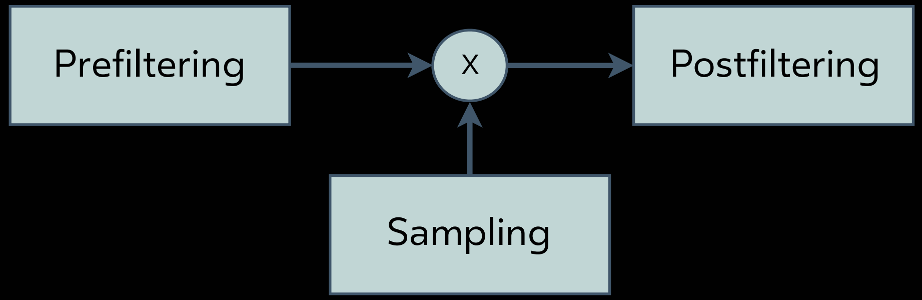
- Signal processing
  - Prefiltering
  - Sampling
  - Postfiltering
- Perceptual imagery
  - Foveation and peripheral degradation
  - Saliency and attention
  - Local image consistency

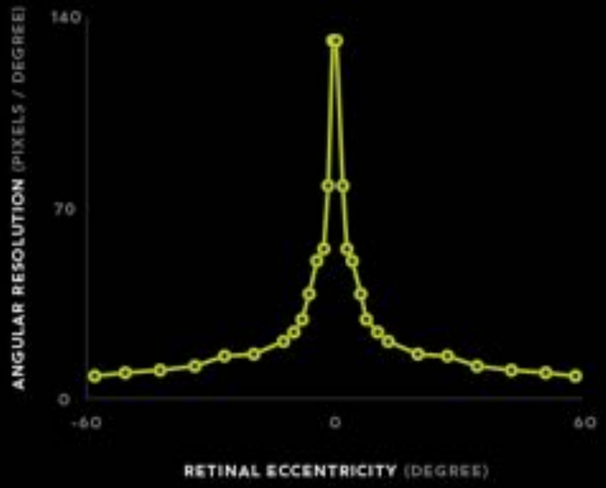




Why Neural Rendering?

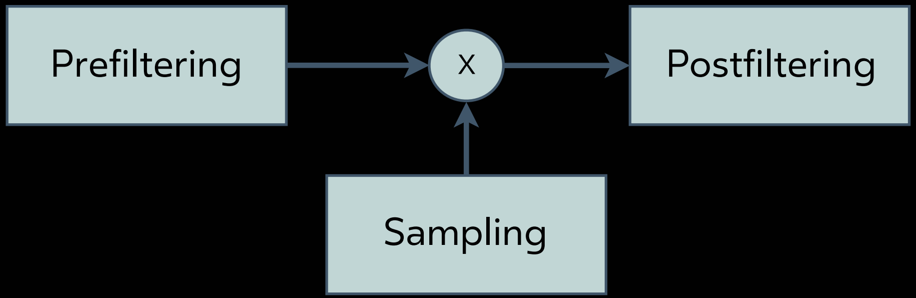
- Signal processing
  - Prefiltering
  - Sampling
  - Postfiltering
- Perceptual imagery
  - Foveation and peripheral degradation
  - Saliency and attention
  - Local image consistency
  - Temporal consistency

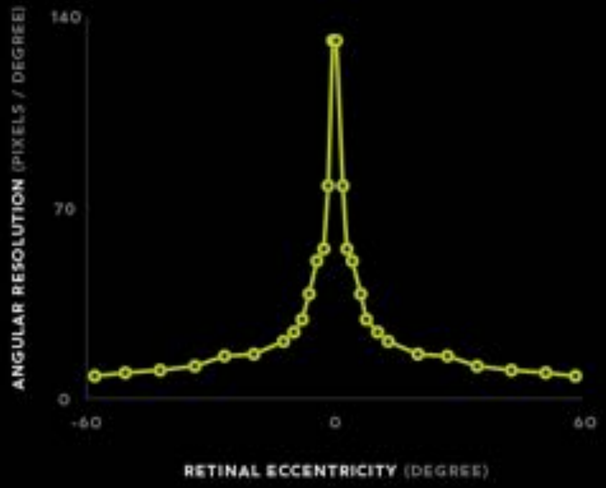




Why Neural Rendering?

- Signal processing
  - Prefiltering
  - Sampling
  - Postfiltering
- Perceptual imagery
  - Foveation and peripheral degradation
  - Saliency and attention
  - Local image consistency
  - Temporal consistency
  - High-level scene understanding





Some Prior Art

- Prefiltering
  - Global illumination with radiance regression functions [Ren13]
- Sampling
  - End-to-end Sampling Patterns [Leimkuehler18]
- Postfiltering
  - A machine learning approach for filtering Monte Carlo noise [Kalantari15]
  - Kernel-Predicting Convolutional Networks for Denoising Monte Carlo Renderings [Bako17]
  - Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder [Chaitanya17]
- Neural scene representation and rendering
  - Full CV and CG pipeline [DeepMind18]

Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder

work done at Nvidia with Chakravarty R. Alla Chaitanya, Christoph Schied,

Marco Salvi, Aaron Lefohn, Derek Nowrouzezahrai, Timo Aila

Global Illumination in Movies and Games

- Used in games
  - Precomputed lighting
  - Coarse real-time approximations

  - Movies
    - Monte Carlo noisy images
    - Denoising is essential

Real-Time Reconstruction

Real-Time Reconstruction

- Limited to a few rays per pixel @1080p @30Hz
  - Never enough to render an image!

Real-Time Reconstruction

- Limited to a few rays per pixel @1080p @30Hz
  - Never enough to render an image!

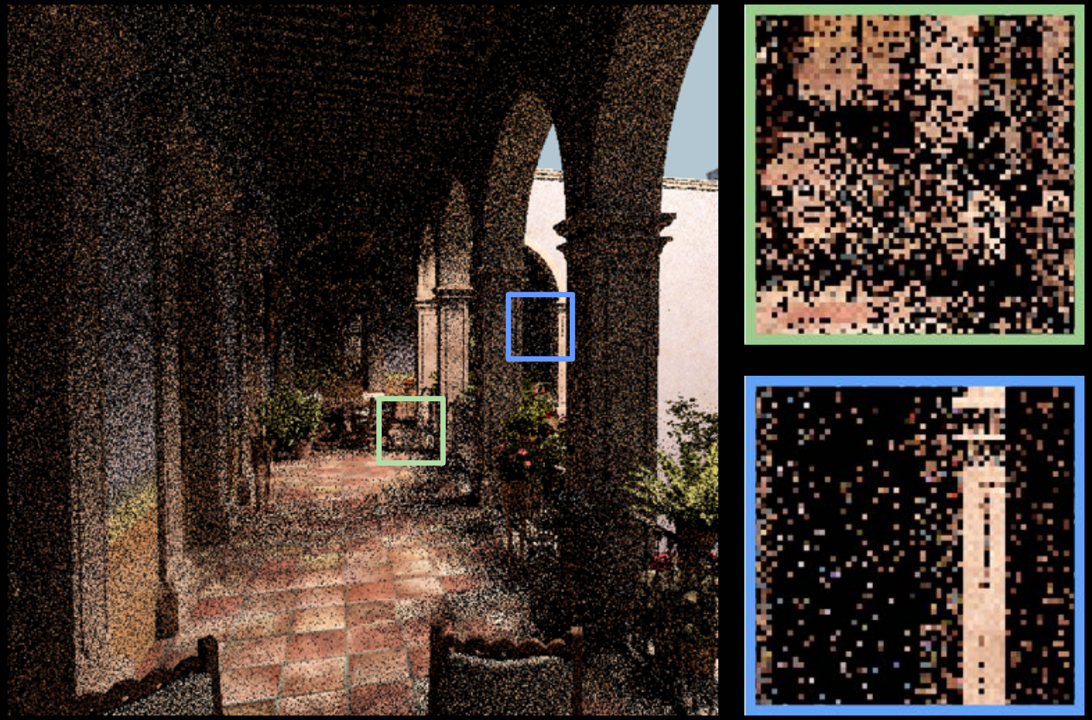
- Deep learning approach for interactive graphics
  - Handle generic effects
    - Soft shadows
    - Diffuse and specular reflections
    - Global illumination (1 bounce)

Real-Time Reconstruction

- Limited to a few rays per pixel @1080p @30Hz
  - Never enough to render an image!

- Deep learning approach for interactive graphics
  - Handle generic effects
    - Soft shadows
    - Diffuse and specular reflections
    - Global illumination (1 bounce)

Noisy input



Real-Time Reconstruction

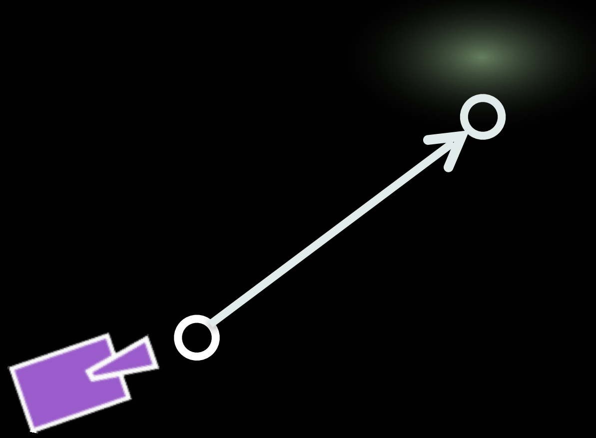
- Limited to a few rays per pixel @1080p @30Hz
  - Never enough to render an image!

- Deep learning approach for interactive graphics
  - Handle generic effects
    - Soft shadows
    - Diffuse and specular reflections
    - Global illumination (1 bounce)

Neural reconstruction

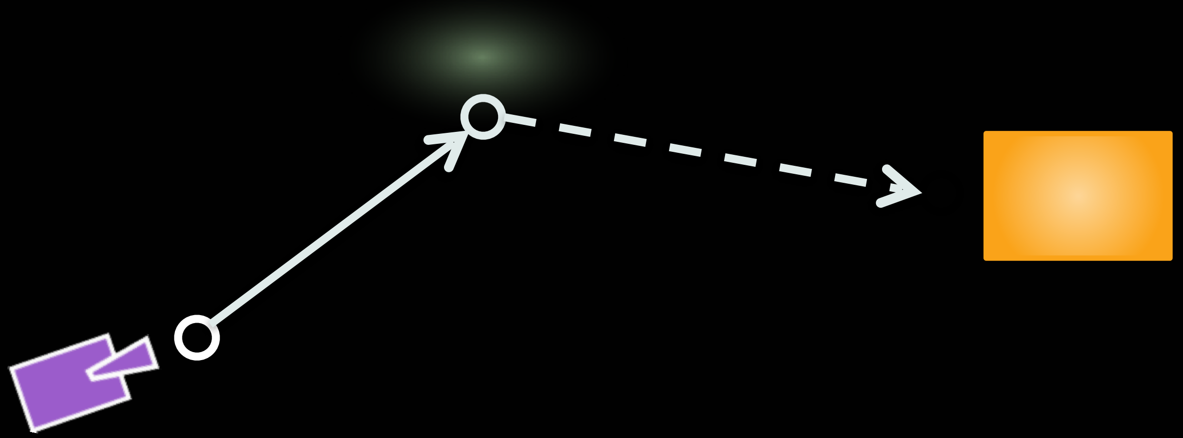
Problem Setup: Real-Time Path Tracing

Problem Setup: Real-Time Path Tracing



Rasterize primary hits into a G-Buffer

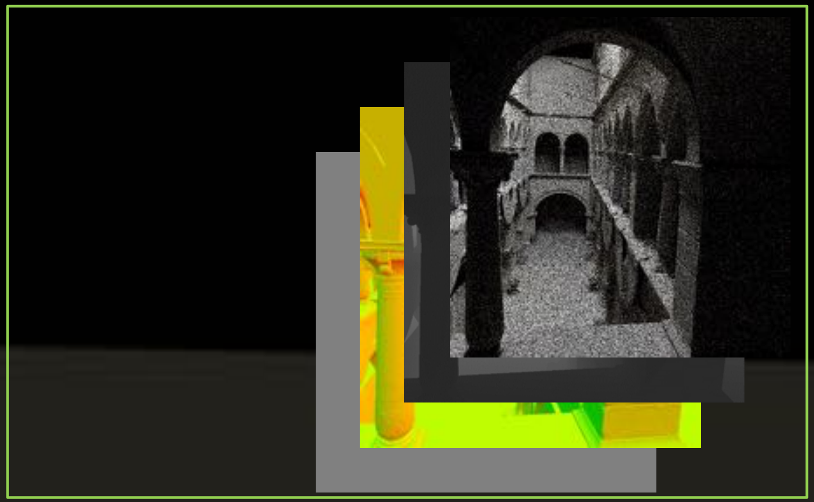
Problem Setup: Real-Time Path Tracing



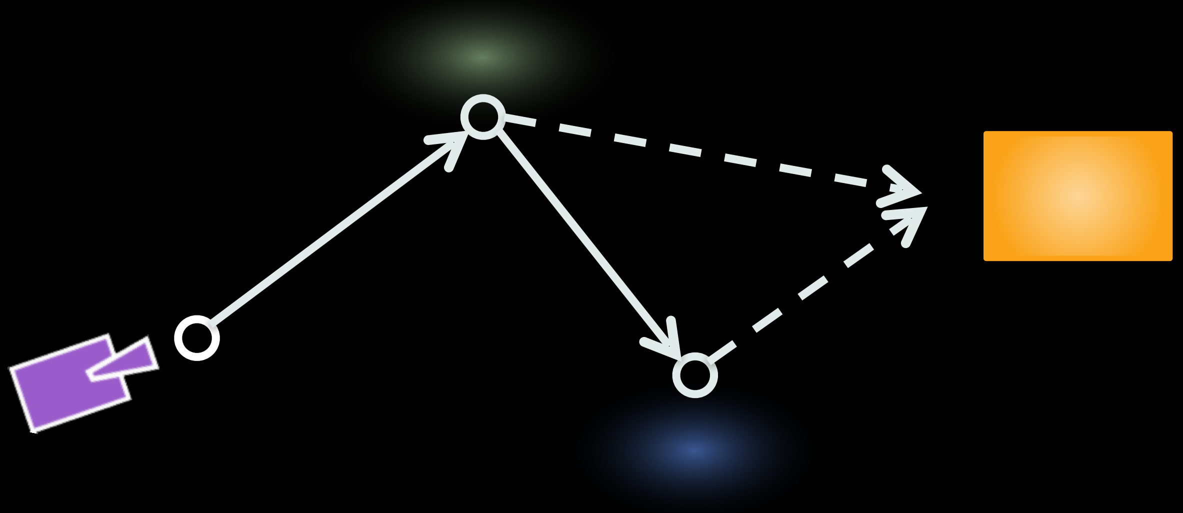
Rasterize primary hits into a G-Buffer

Path tracing from the primary hits

1 ray for direct shadows

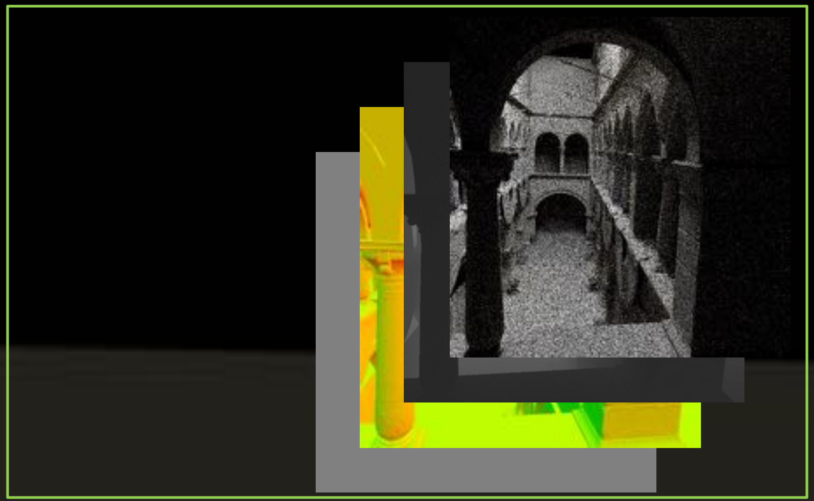


Problem Setup: Real-Time Path Tracing

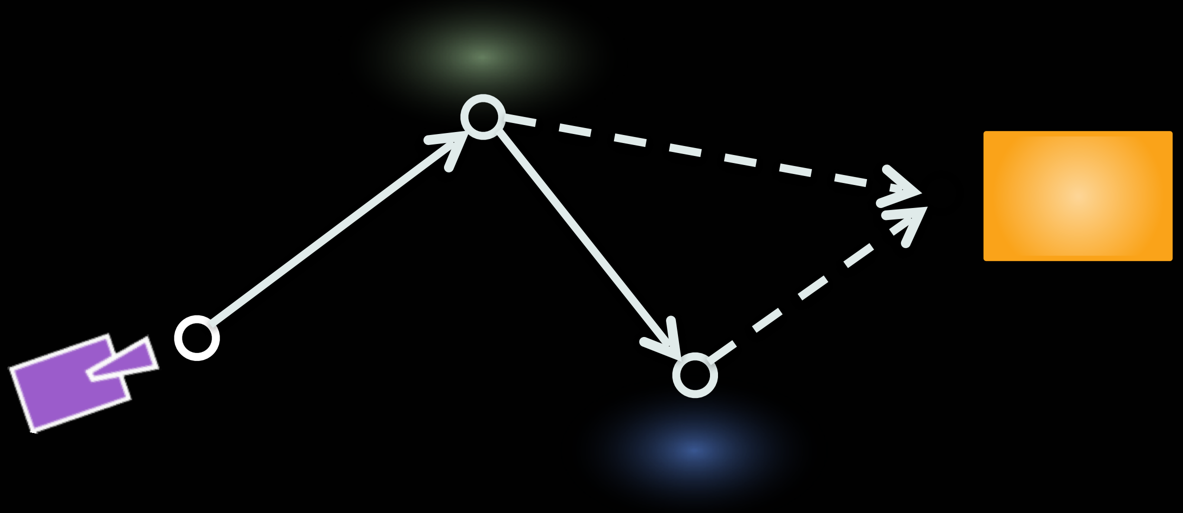


Rasterize primary hits into a G-Buffer

Path tracing from the primary hits
- 1 ray for direct shadows
- 2 rays for indirect (sample + connect)



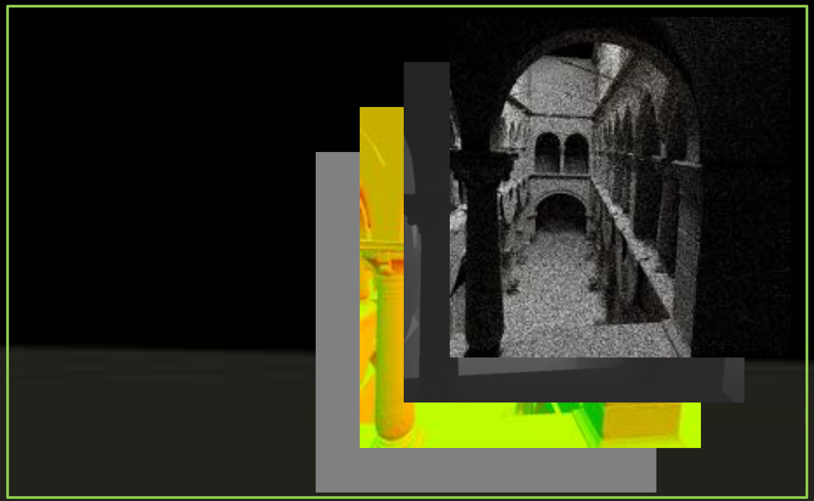
Problem Setup: Real-Time Path Tracing



Rasterize primary hits into a G-Buffer

Path tracing from the primary hits
- 1 ray for direct shadows
- 2 rays for indirect (sample + connect)

1 direct + 1 indirect path := 1spp



Related Work: Offline

Related Work: Offline

- Sheared filters [Egan11b]

Related Work: Offline

- Sheared filters [Egan11b]
- Light field reconstruction [Lehtinen11, Lehtinen12]

Related Work: Offline

- Sheared filters [Egan11b]
- Light field reconstruction [Lehtinen11, Lehtinen12]
- Stein's unbiased risk estimate based filter [Li12]
- Denoising using feature and color [Rousselle13]

Related Work: Offline

- Sheared filters [Egan11b]
- Light field reconstruction [Lehtinen11, Lehtinen12]
- Stein's unbiased risk estimate based filter [Li12]
- Denoising using feature and color [Rousselle13]
- Local regression models [Bitterli16, Moon15, Moon16]

Related Work: Interactive

Related Work: Interactive

- Frequency-space analysis of light transport [Mehta12, Mehta13, Mehta14, Yan15]

Related Work: Interactive

- Frequency-space analysis of light transport [Mehta12, Mehta13, Mehta14, Yan15]
- Edge-avoiding wavelet filter [Dammertz10]
- Guided image filters [Bauszat15]
- Texture space [Munkberg16]

Related Work: Machine Learning

Related Work: Machine Learning

- Image inpainting [Pathak16]
- Single-image super resolution [Ledig16]

Related Work: Machine Learning

- Image inpainting [Pathak16]
- Single-image super resolution [Ledig16]
- Image classification [Krizhevsky12]

Related Work: Machine Learning

- Image inpainting [Pathak16]
- Single-image super resolution [Ledig16]
- Image classification [Krizhevsky12]
- Image restoration [Mao16]

Related Work: Machine Learning

- Image inpainting [Pathak16]
- Single-image super resolution [Ledig16]
- Image classification [Krizhevsky12]
- Image restoration [Mao16]
- Learning based filter for Monte Carlo denoising [Kalantari15]

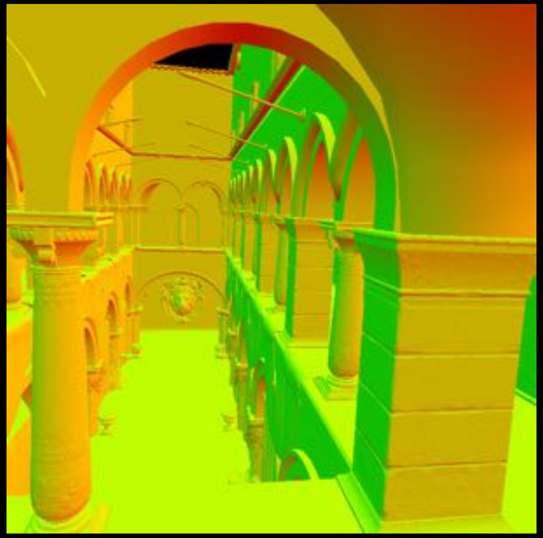
Related Work: Machine Learning

- Image inpainting [Pathak16]
- Single-image super resolution [Ledig16]
- Image classification [Krizhevsky12]
- Image restoration [Mao16]
- Learning based filter for Monte Carlo denoising [Kalantari15]
- Disney offline denoiser [Bako17, Vogels18]

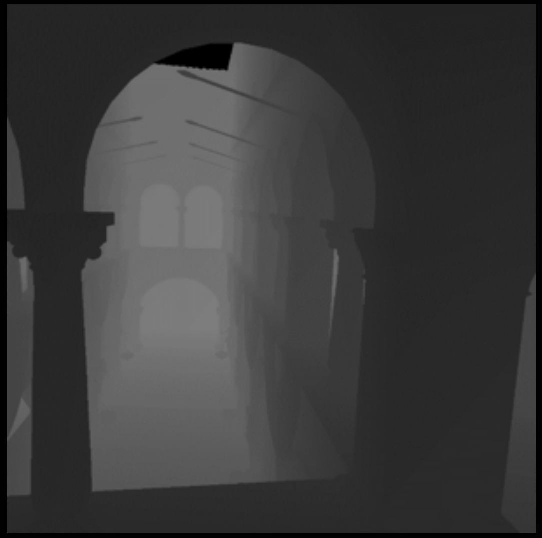
Input Features

- Additional features from primary visibility (G-Buffer)

Untextured illumination



View-space normals

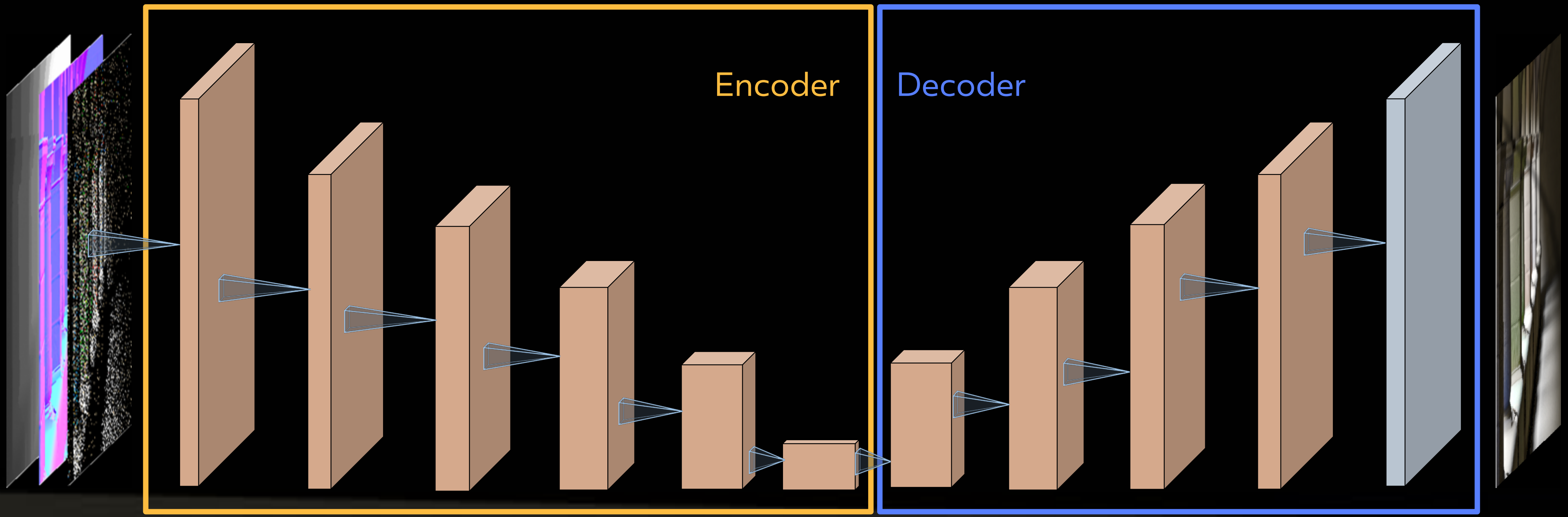


Linear depth and roughness

U-Net Design

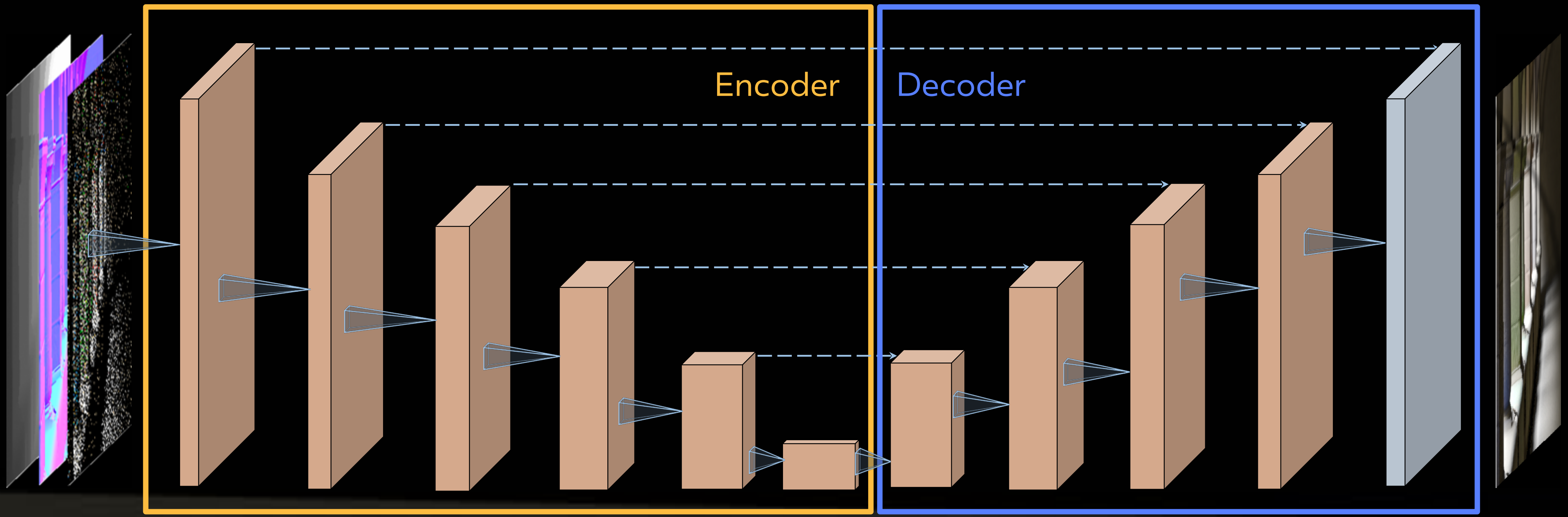
U-Net Design

- Encoder and decoder stages of a U-Net for hierarchical representation



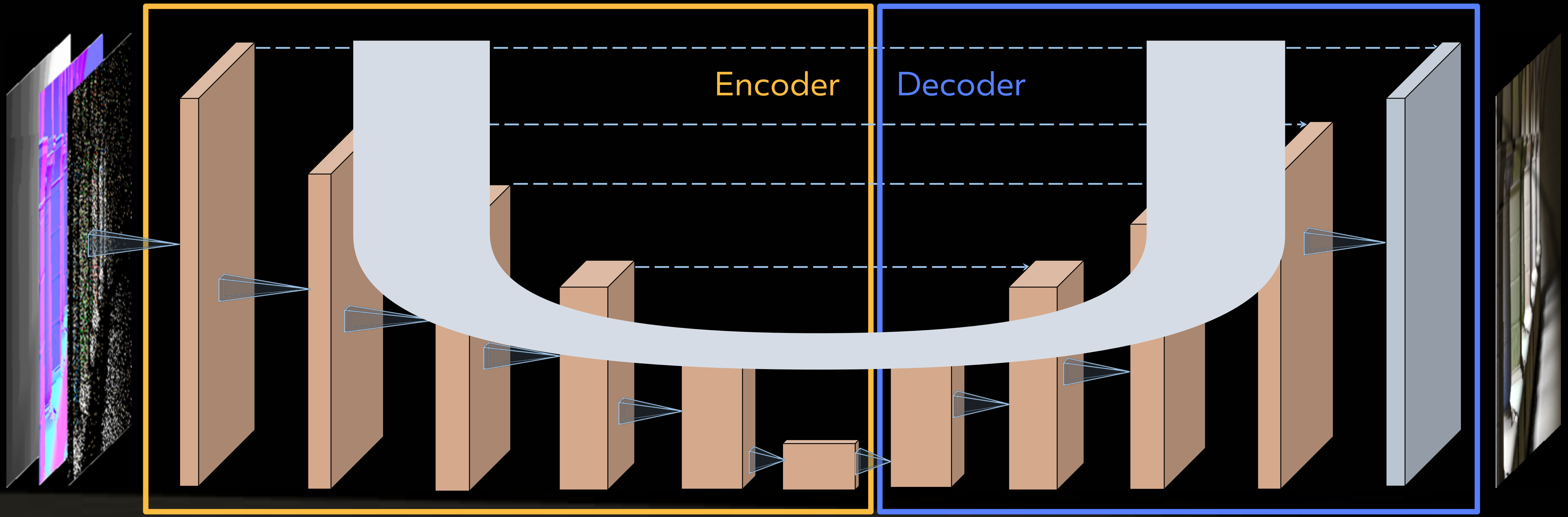
U-Net Design

- Encoder and decoder stages of a U-Net for hierarchical representation
- Skip connections to pass high frequencies and learn residuals



U-Net Design

- Encoder and decoder stages of a U-Net for hierarchical representation
- Skip connections to pass high frequencies and learn residuals



Training sequences

Sponza Diffuse

Sponza Glossy

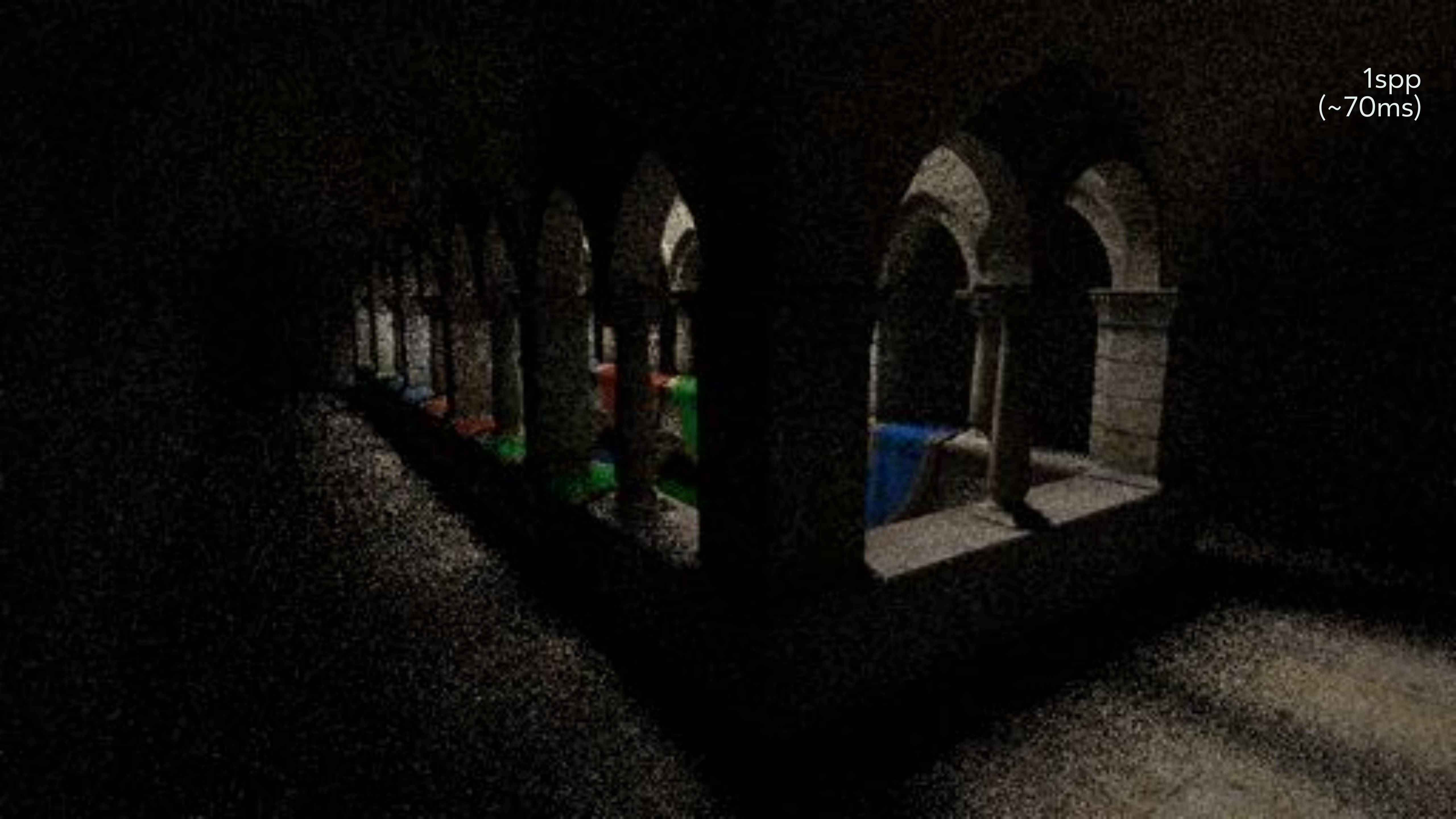
Classroom

Training sequences

Sponza Diffuse

Sponza Glossy

Classroom


1spp
(~70ms)


U-Net (1spp)
(~70ms + ~60ms)

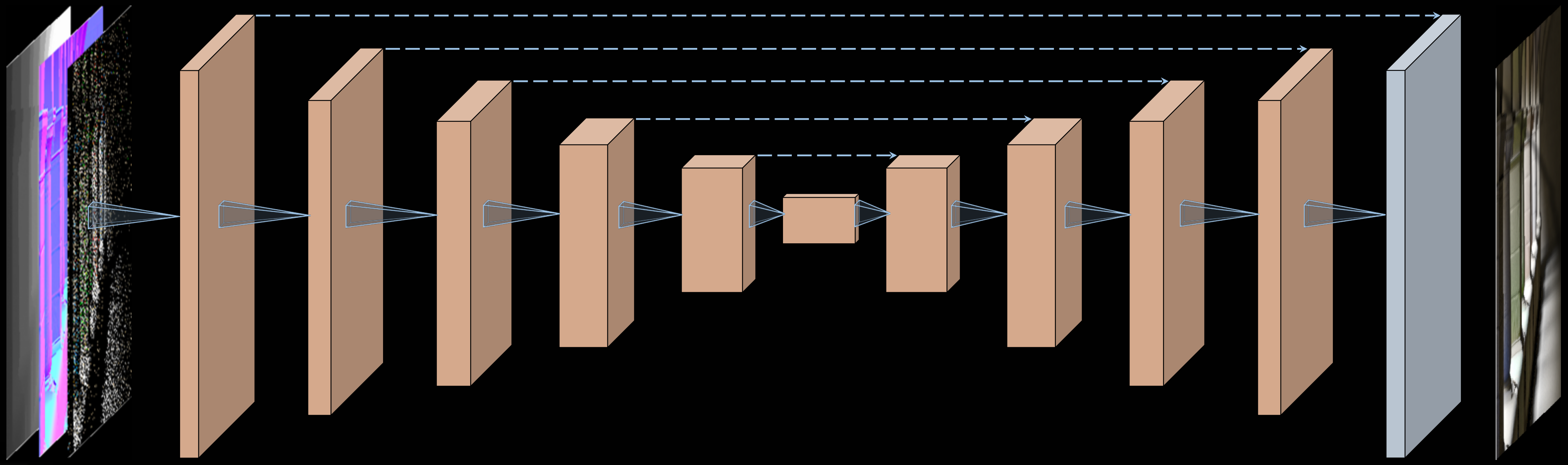

Reference (1024spp)
(~240s)

Image-to-image results

Image-to-image results

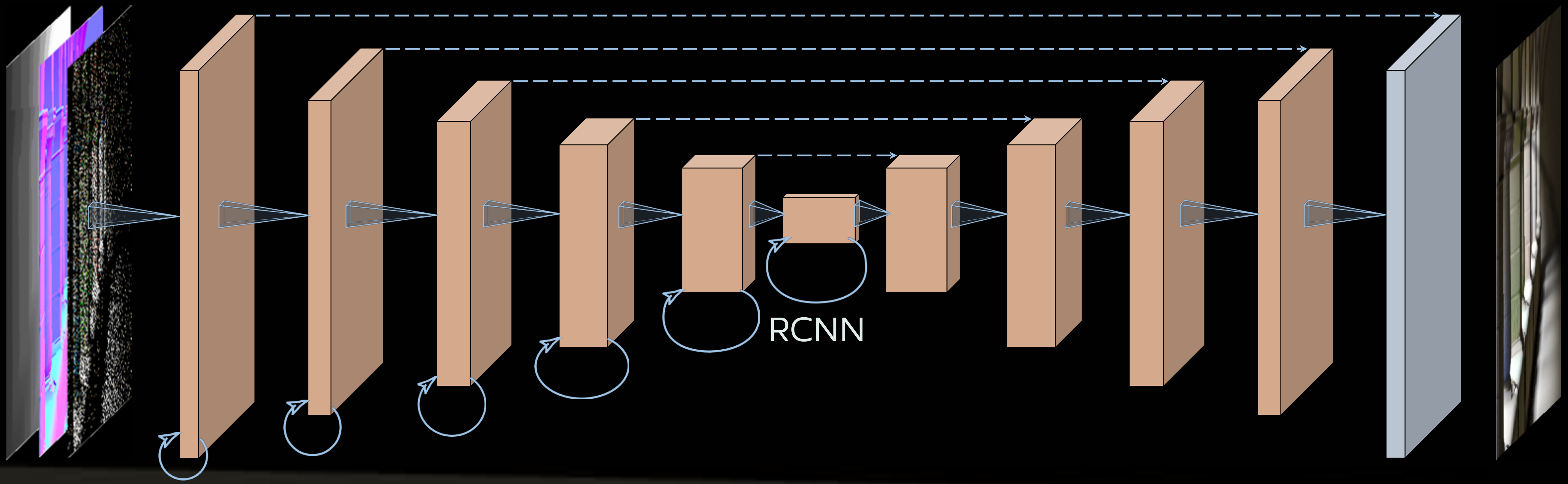
Temporal Stability

Recurrent U-Net



Recurrent U-Net

- Recurrent connections retain important features at different scales over time



Recurrent Block

Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution

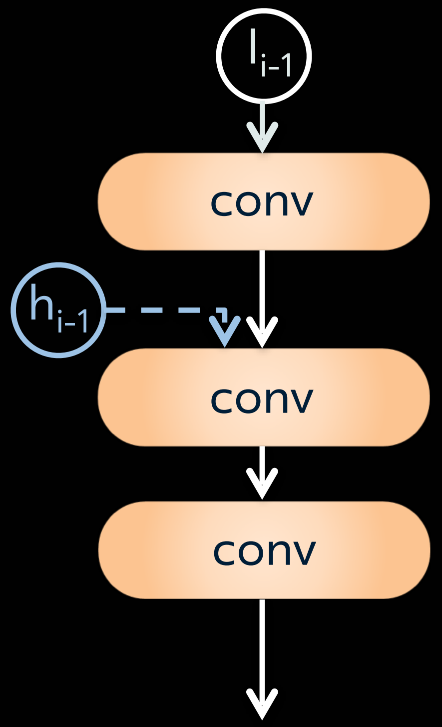
Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution
- 6 RNN blocks, one per pool layer in the encoder

Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution
- 6 RNN blocks, one per pool layer in the encoder
- Design
  - 1 conv layer (3x3) for current features
  - 2 conv layers (3x3) for previous features

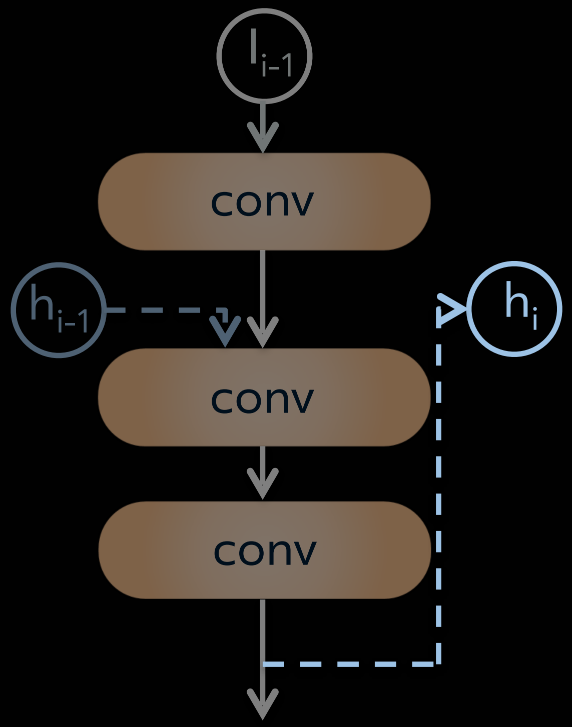
Recurrent Convolutional Block



Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution
- 6 RNN blocks, one per pool layer in the encoder
- Design
  - 1 conv layer (3x3) for current features
  - 2 conv layers (3x3) for previous features

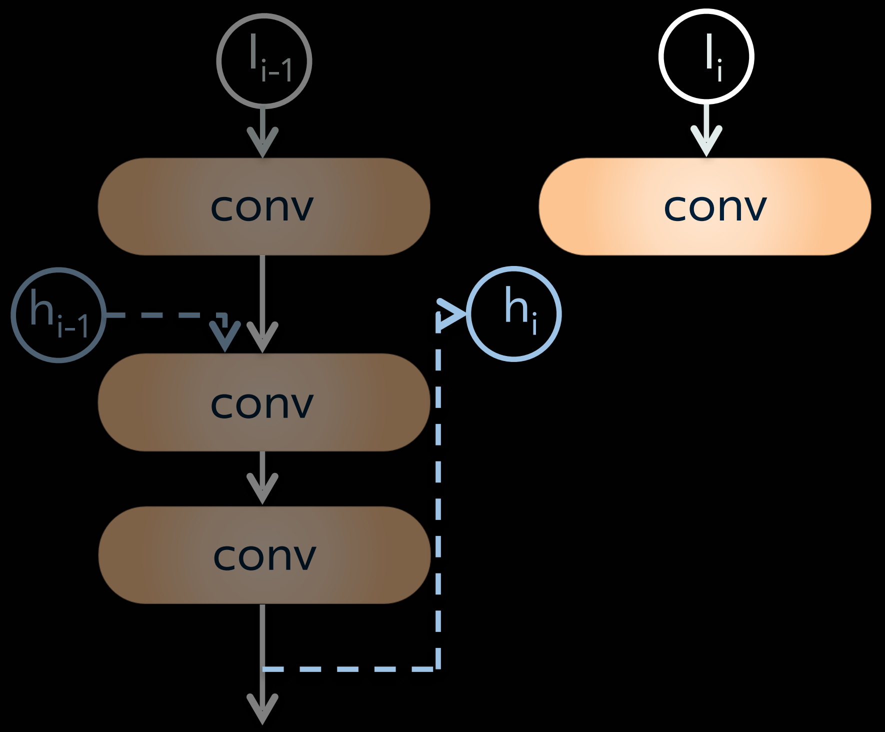
Recurrent Convolutional Block



Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution
- 6 RNN blocks, one per pool layer in the encoder
- Design
  - 1 conv layer (3x3) for current features
  - 2 conv layers (3x3) for previous features

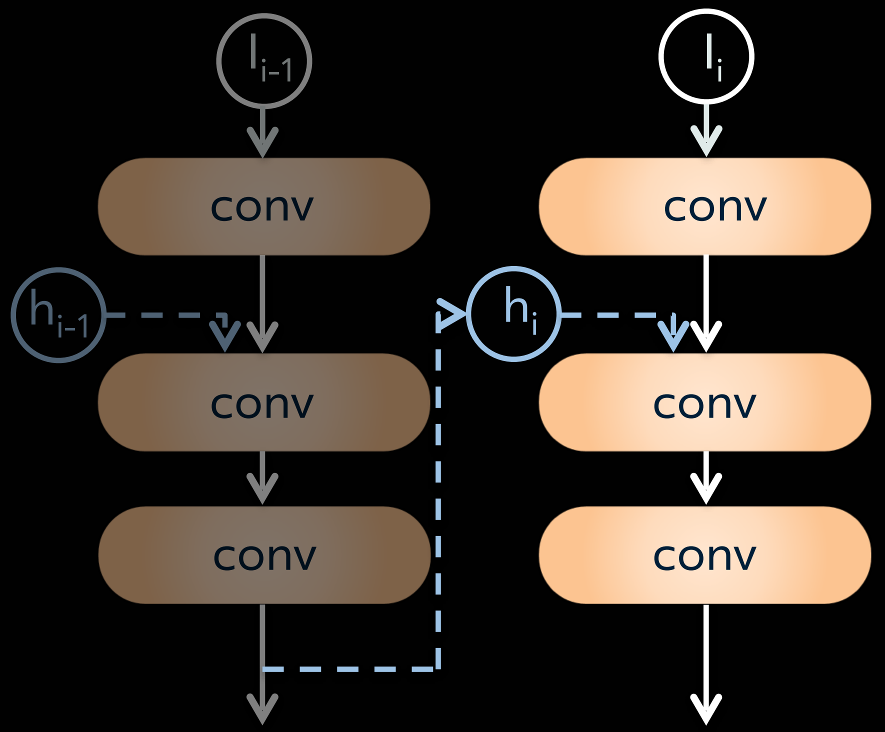
Recurrent Convolutional Block



Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution
- 6 RNN blocks, one per pool layer in the encoder
- Design
  - 1 conv layer (3x3) for current features
  - 2 conv layers (3x3) for previous features

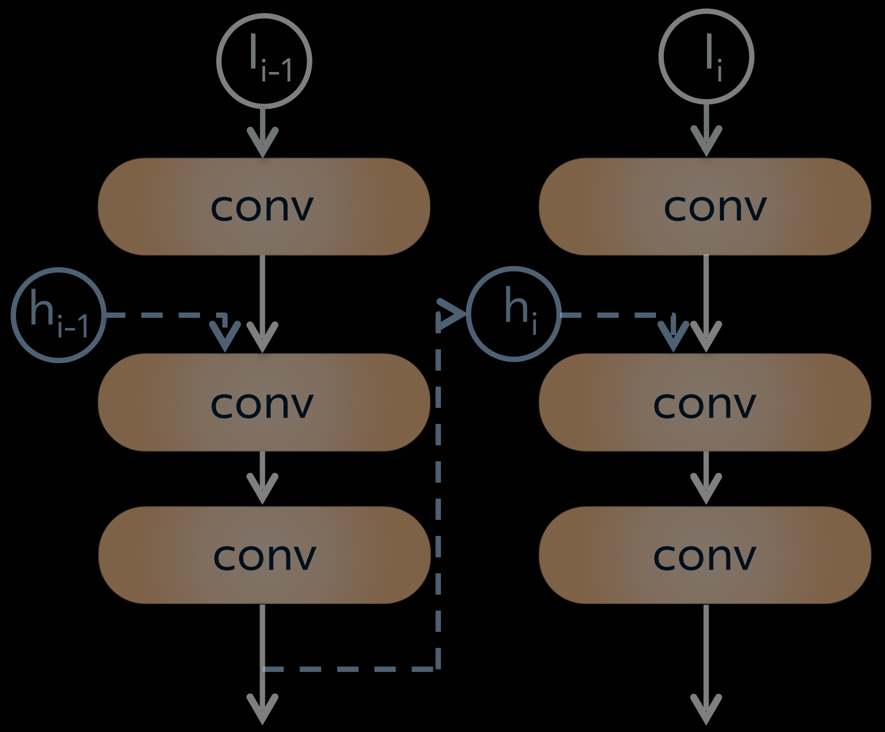
Recurrent Convolutional Block



Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution
- 6 RNN blocks, one per pool layer in the encoder
- Design
  - 1 conv layer (3x3) for current features
  - 2 conv layers (3x3) for previous features

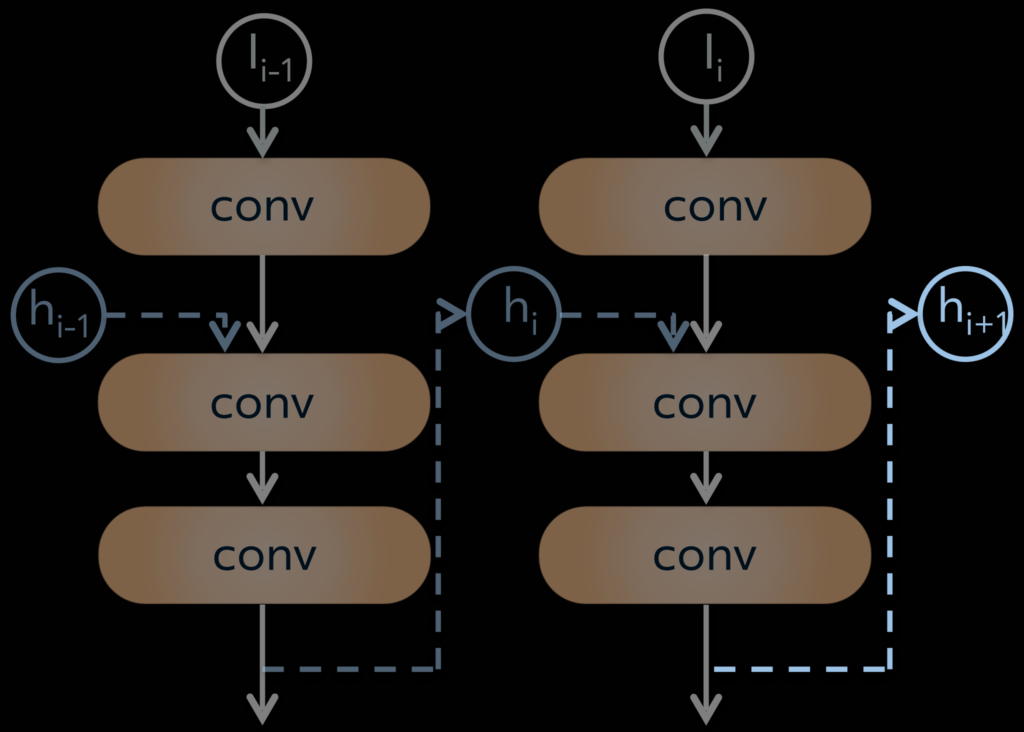
Recurrent Convolutional Block



Recurrent Block

- Fully convolutional blocks to support arbitrary image resolution
- 6 RNN blocks, one per pool layer in the encoder
- Design
  - 1 conv layer (3x3) for current features
  - 2 conv layers (3x3) for previous features

Recurrent Convolutional Block



Temporal Training

Temporal Training

- Sequence of 7 frames

Temporal Training

- Sequence of 7 frames
- Increase loss with number of frames

Temporal Training

- Sequence of 7 frames
- Increase loss with number of frames
- Augmentation: Play the sequence forward/backward

Temporal Training

- Sequence of 7 frames
- Increase loss with number of frames
- Augmentation: Play the sequence forward/backward
- Augmentation: Each frame can either advance or freeze the camera

Loss Function

Spatial loss for more emphasis on dark regions

Temporal loss for better temporal stability

High Frequency Error Norm loss for stable edges [Ravishankar11]

Final training loss is a weighted combination

Loss Function

Spatial loss for more emphasis on dark regions

$$L_s = \frac{1}{N}\sum_{i}^{N}|P_i - T_i|$$

Temporal loss for better temporal stability

High Frequency Error Norm loss for stable edges [Ravishankar11]

Final training loss is a weighted combination

Loss Function

Spatial loss for more emphasis on dark regions

$$L_s = \frac{1}{N}\sum_{i}^{N}|P_i - T_i|$$

High Frequency Error Norm loss for stable edges [Ravishankar11]

$$L_g = \frac{1}{N}\sum_{i}^{N}|\nabla P_i - \nabla T_i|$$

Temporal loss for better temporal stability

Final training loss is a weighted combination

Loss Function

Spatial loss for more emphasis on dark regions

$$L_s = \frac{1}{N}\sum_{i}^{N}|P_i - T_i|$$

Temporal loss for better temporal stability

$$L_t = \frac{1}{N}\sum_{i}^{N}\left(\left|\frac{\partial P_i}{\partial t} - \frac{\partial T_i}{\partial t}\right|\right)$$

High Frequency Error Norm loss for stable edges [Ravishankar11]

$$L_g = \frac{1}{N}\sum_{i}^{N}|\nabla P_i - \nabla T_i|$$

Final training loss is a weighted combination

Loss Function

Spatial loss for more emphasis on dark regions

$$L_s = \frac{1}{N}\sum_{i}^{N}|P_i - T_i|$$

Temporal loss for better temporal stability

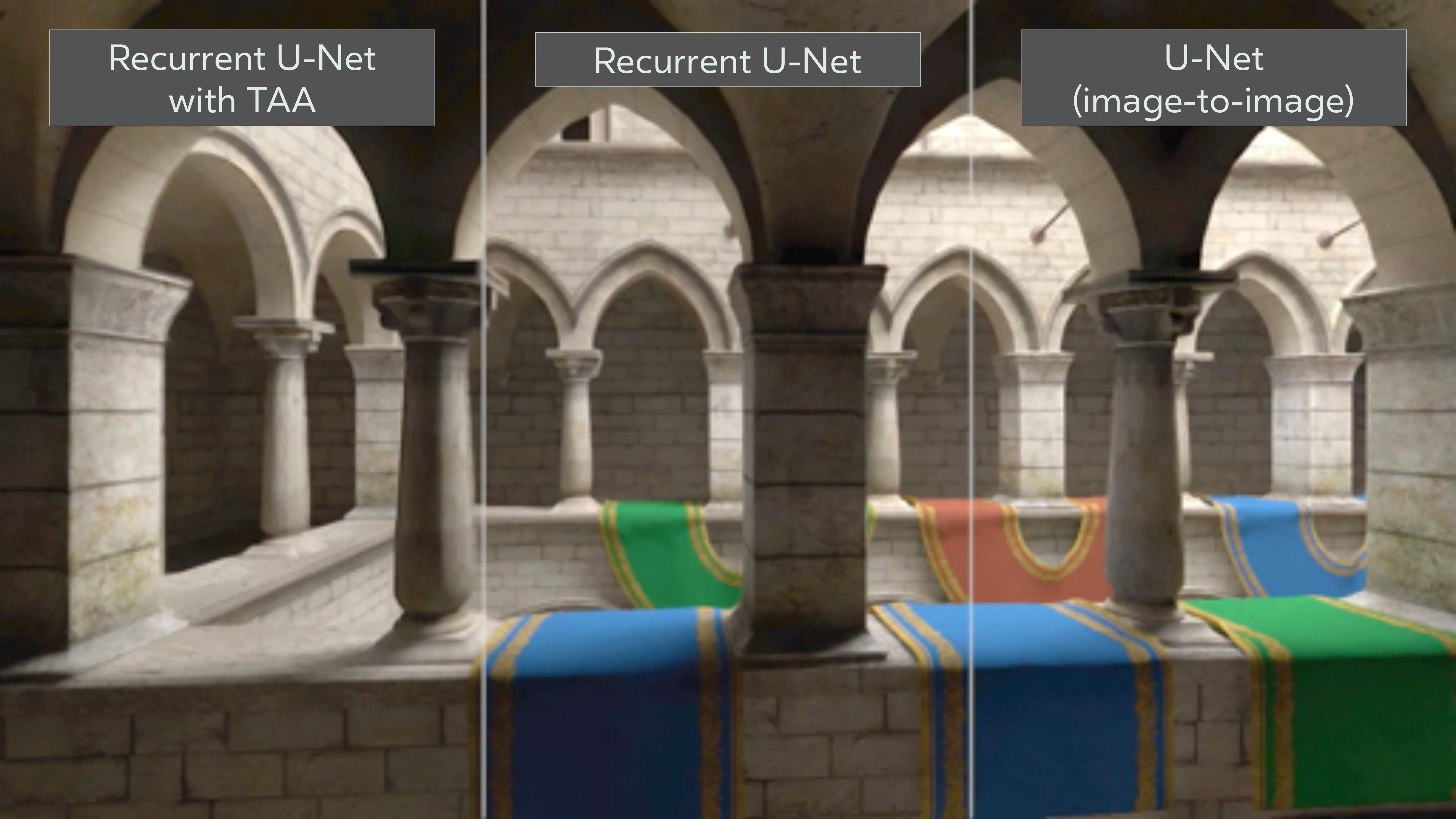
$$L_t = \frac{1}{N}\sum_{i}^{N}\left(\left|\frac{\partial P_i}{\partial t} - \frac{\partial T_i}{\partial t}\right|\right)$$

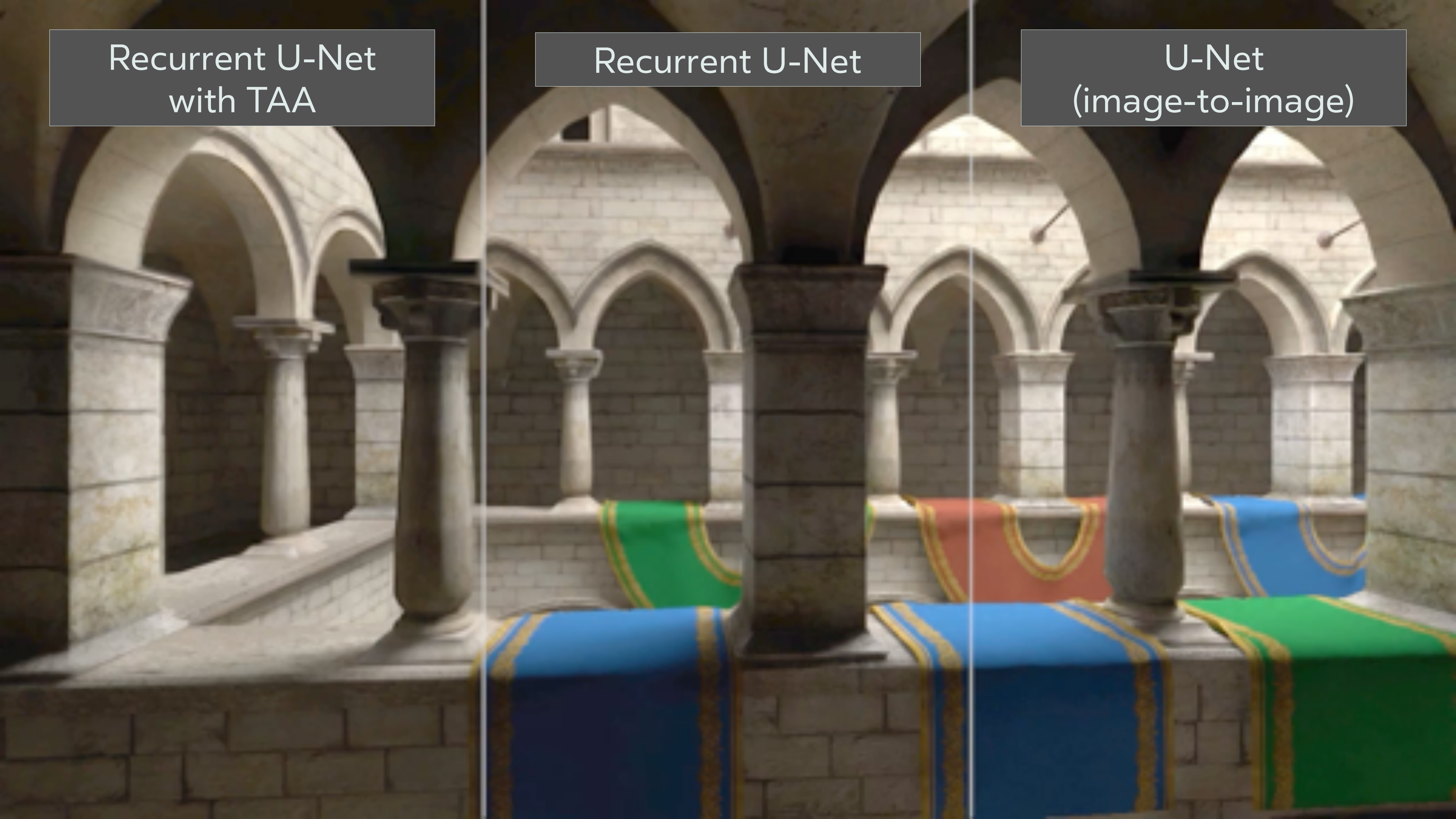
High Frequency Error Norm loss for stable edges [Ravishankar11]

$$L_g = \frac{1}{N}\sum_{i}^{N}|\nabla P_i - \nabla T_i|$$

Final training loss is a weighted combination

$$L = w_s L_s + w_g L_g + w_t L_t$$


Recurrent U-Net
with TAA
Recurrent U-Net
U-Net
(image-to-image)

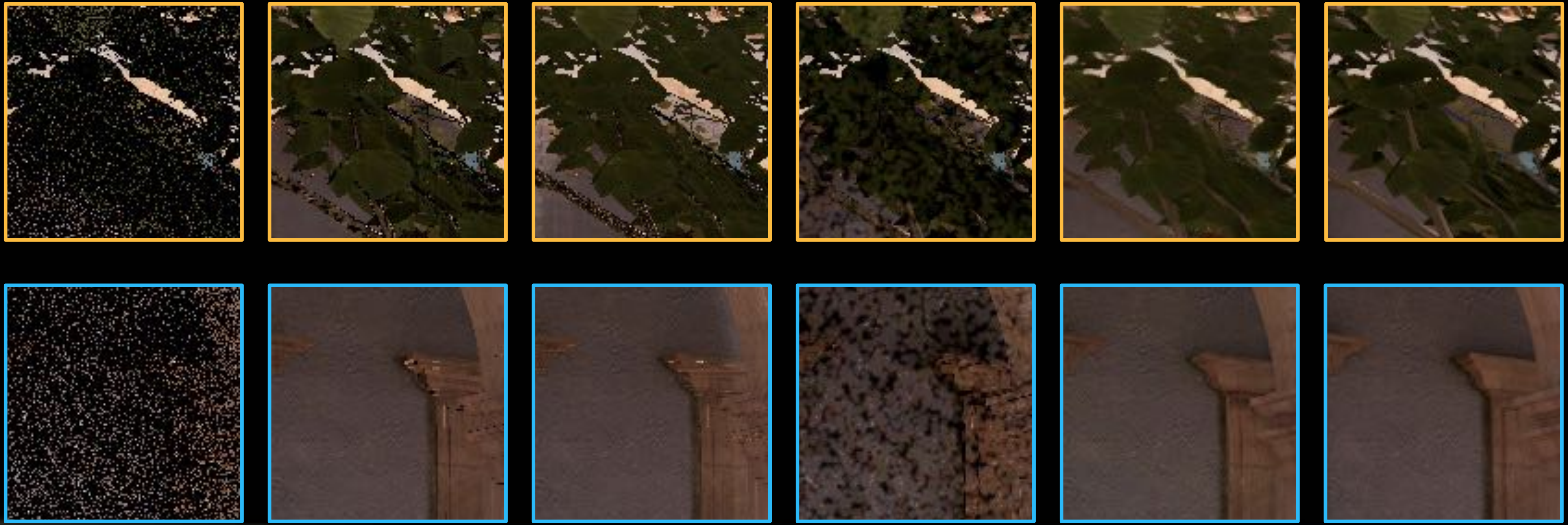

Recurrent U-Net
with TAA
Recurrent U-Net
U-Net
(image-to-image)

Reconstruction Results

San Miguel Results

San Miguel Comparison Results

MC Input | AAF | EAW | SBF | Our | Reference



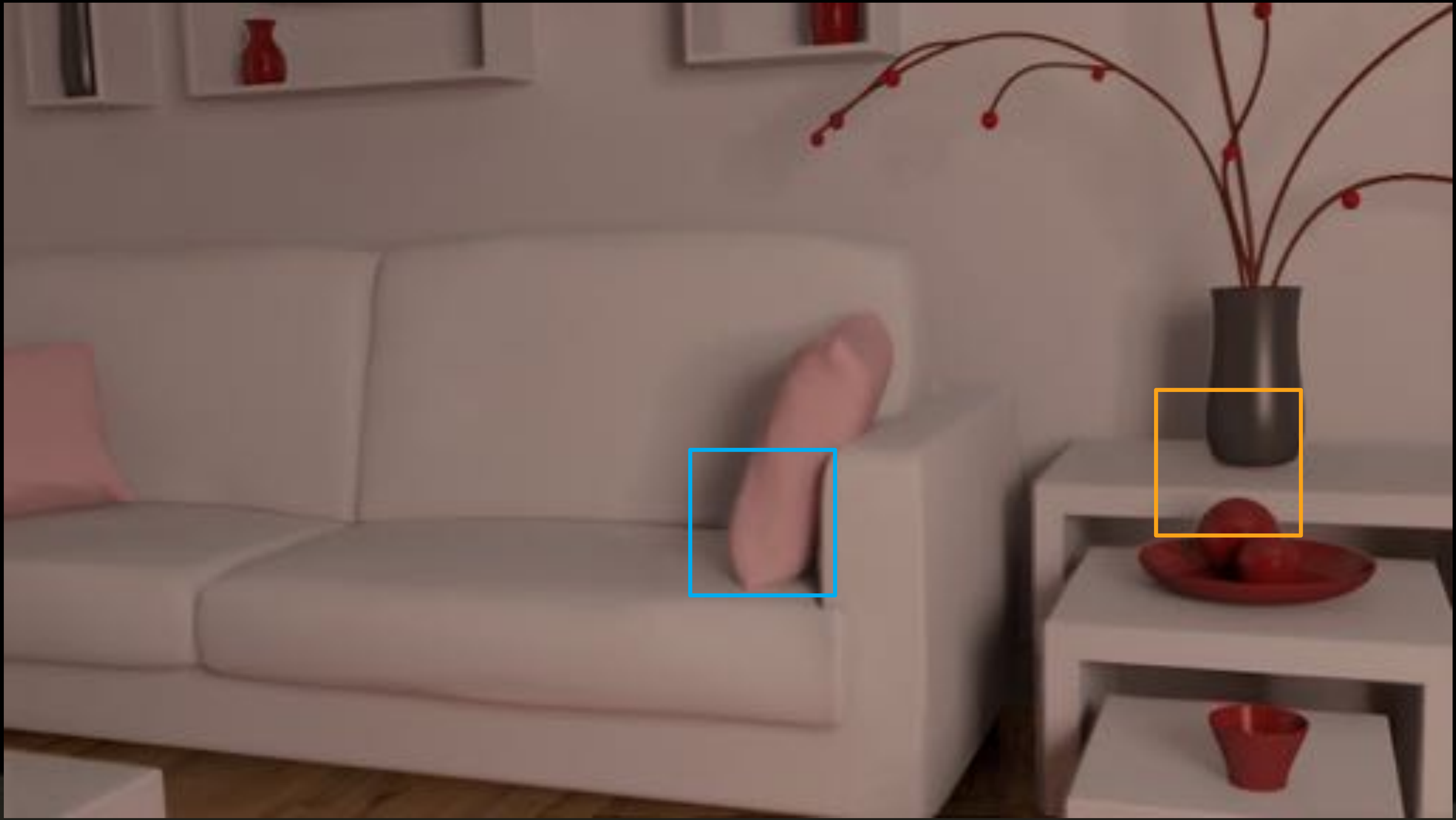
RMSE: 0.079 | RMSE: 0.088 | RMSE: 0.087 | RMSE: 0.055


1 sample/pixel input


1 sample/pixel input

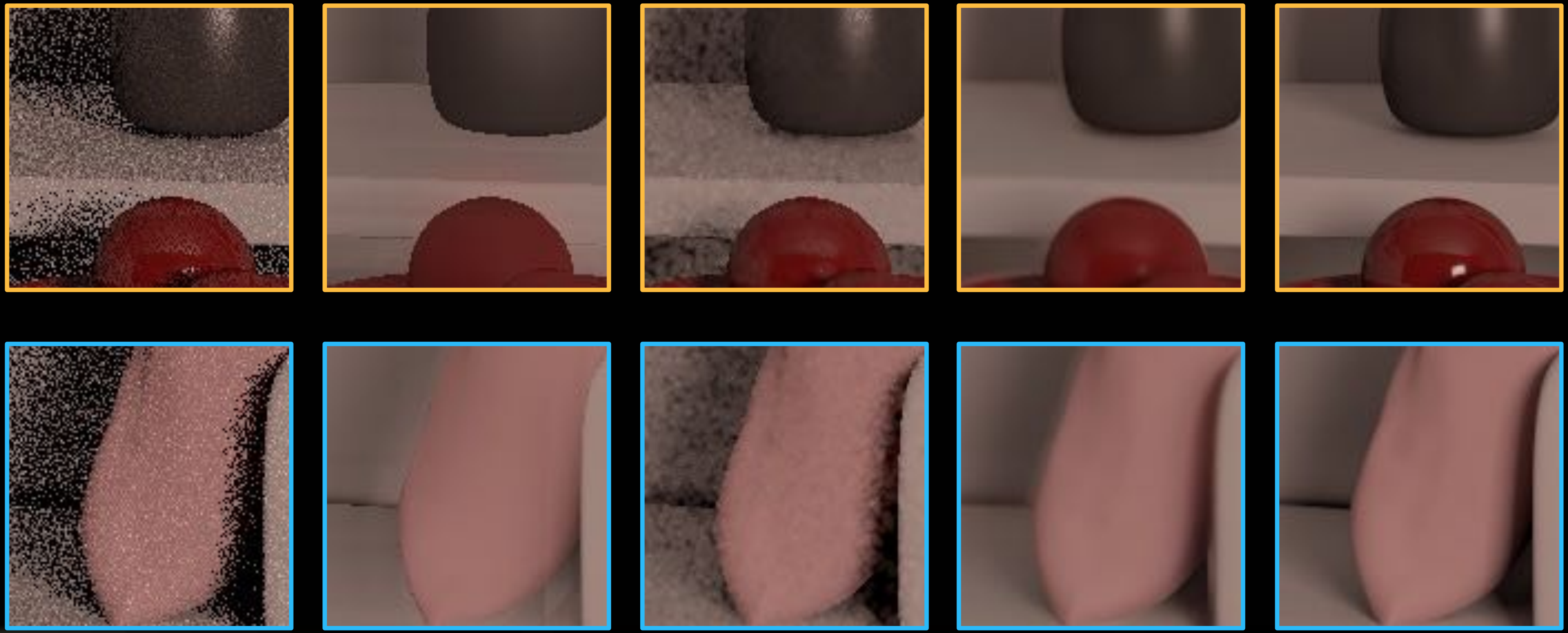

1 sample/pixel input

Red Room Results



Red Room Comparison Results

MC Input EAW SBF Our Reference



RMSE: 0.041 RMSE: 0.052 RMSE: 0.029

GENERALIZATION:
OFFLINE 256SPP INPUT

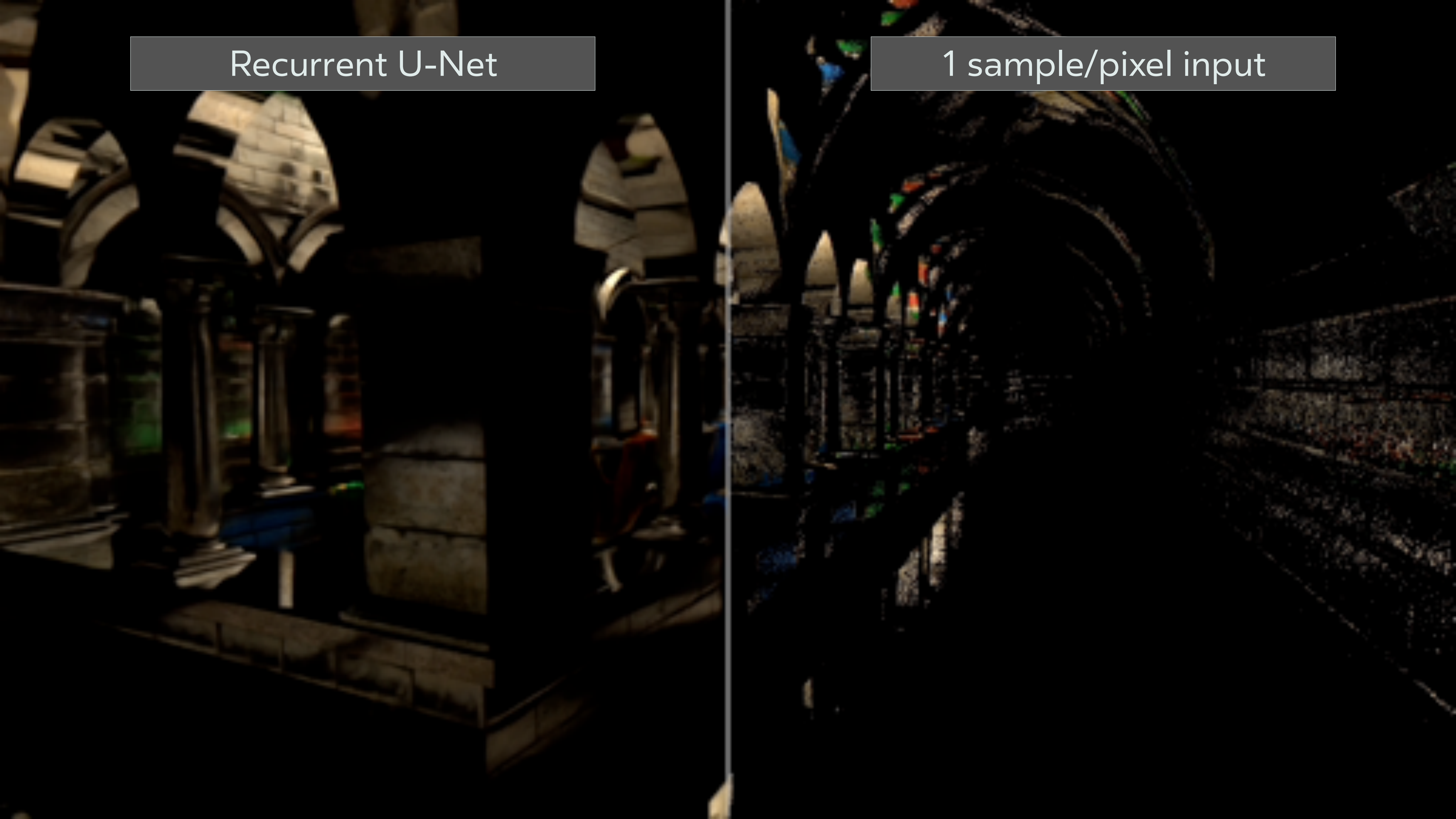
Horse Room, 256spp

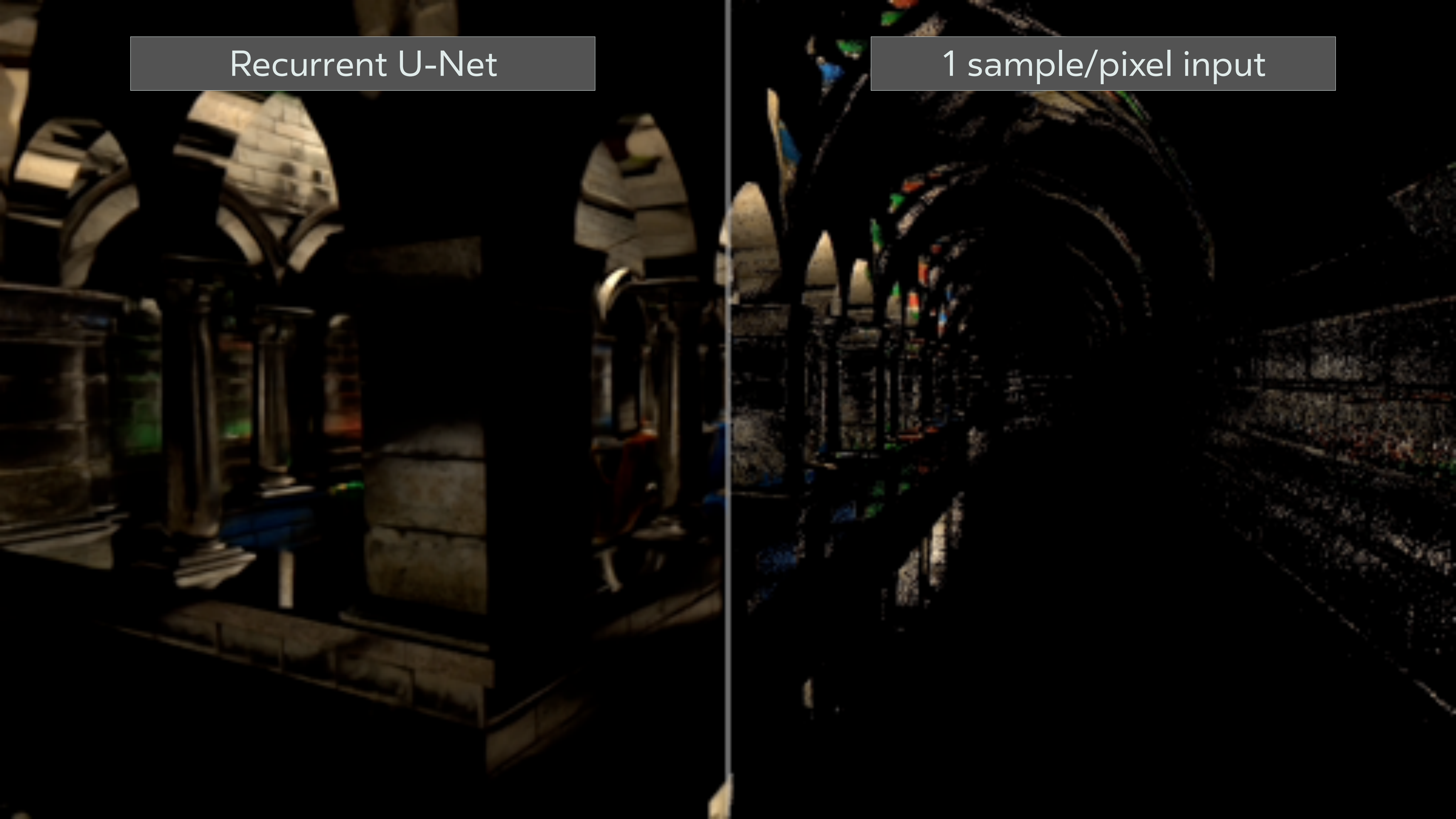
Horse Room ComparisonS

MC Input	EAW	SBF	NFOR	Our	Reference

	EAW	SBF	NFOR	Our	
	RMSE: 0.094	RMSE: 0.040	RMSE: 0.018	RMSE: 0.034	
	10.3ms	74.2ms	110s	54.9ms	

GENERALIZATION: SPECULAR MATERIALS

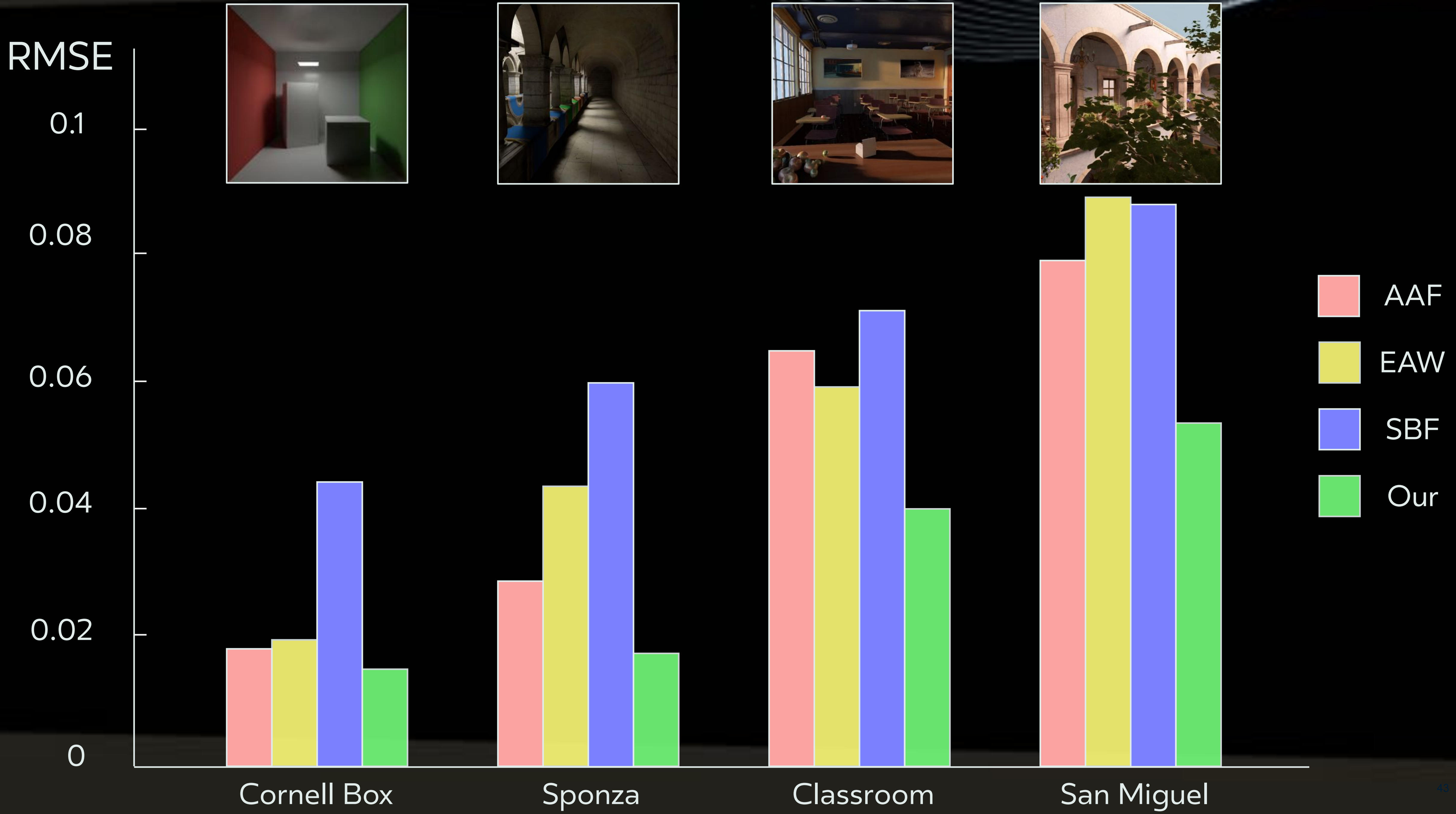

Recurrent U-Net
1 sample/pixel input


Recurrent U-Net
1 sample/pixel input

Performance

- Optimized CUDA and cuDNN inference
  - Kudos to Jon Hasselgren and Jacob Munkberg

- **54.9ms** on NVIDIA Titan X (Pascal) on a 720p image
  - Volta is 3x faster, Turing is 3x3=9x faster?



RMSE
0.1
0.08
0.06
0.04
0.02
0
AAF
EAW
SBF
Our
Cornell Box
Sponza
Classroom
San Miguel

Conclusion

- Deep learning application to 1spp reconstruction

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability
  - Interactive performance

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability
  - Interactive performance

- Follow up work

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability
  - Interactive performance

– Follow up work
  - Deep Adaptive Sampling for Low Sample Count Rendering [Kuznetsov18]

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability
  - Interactive performance

- Follow up work
  - Deep Adaptive Sampling for Low Sample Count Rendering [Kuznetsov18]
- OptiX 5 denoiser is based on this work

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability
  - Interactive performance

- Follow up work
  - Deep Adaptive Sampling for Low Sample Count Rendering [Kuznetsov18]
- OptiX 5 denoiser is based on this work
  - 19ms performance on Titan V (1080p)

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability
  - Interactive performance

- Follow up work
  - Deep Adaptive Sampling for Low Sample Count Rendering [Kuznetsov18]
- OptiX 5 denoiser is based on this work
  - 19ms performance on Titan V (1080p)
  - Wide adaption in interactive rendering

Conclusion

- Deep learning application to 1spp reconstruction
  - Recurrent U-Net for temporal stability
  - Interactive performance

- Follow up work
  - Deep Adaptive Sampling for Low Sample Count Rendering [Kuznetsov18]
- OptiX 5 denoiser is based on this work
  - 19ms performance on Titan V (1080p)
  - Wide adaption in interactive rendering
  - Limited to image-to-image, firefly filter required

GENERATIONS / VANCOUVER

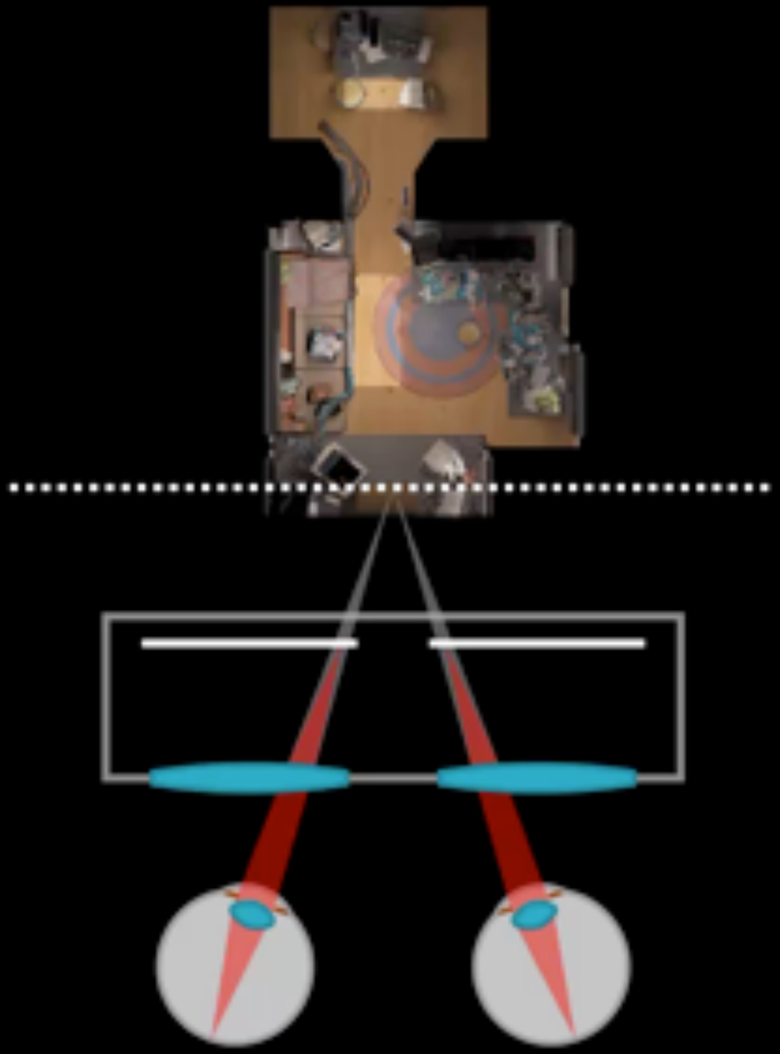
SIGGRAPH2018

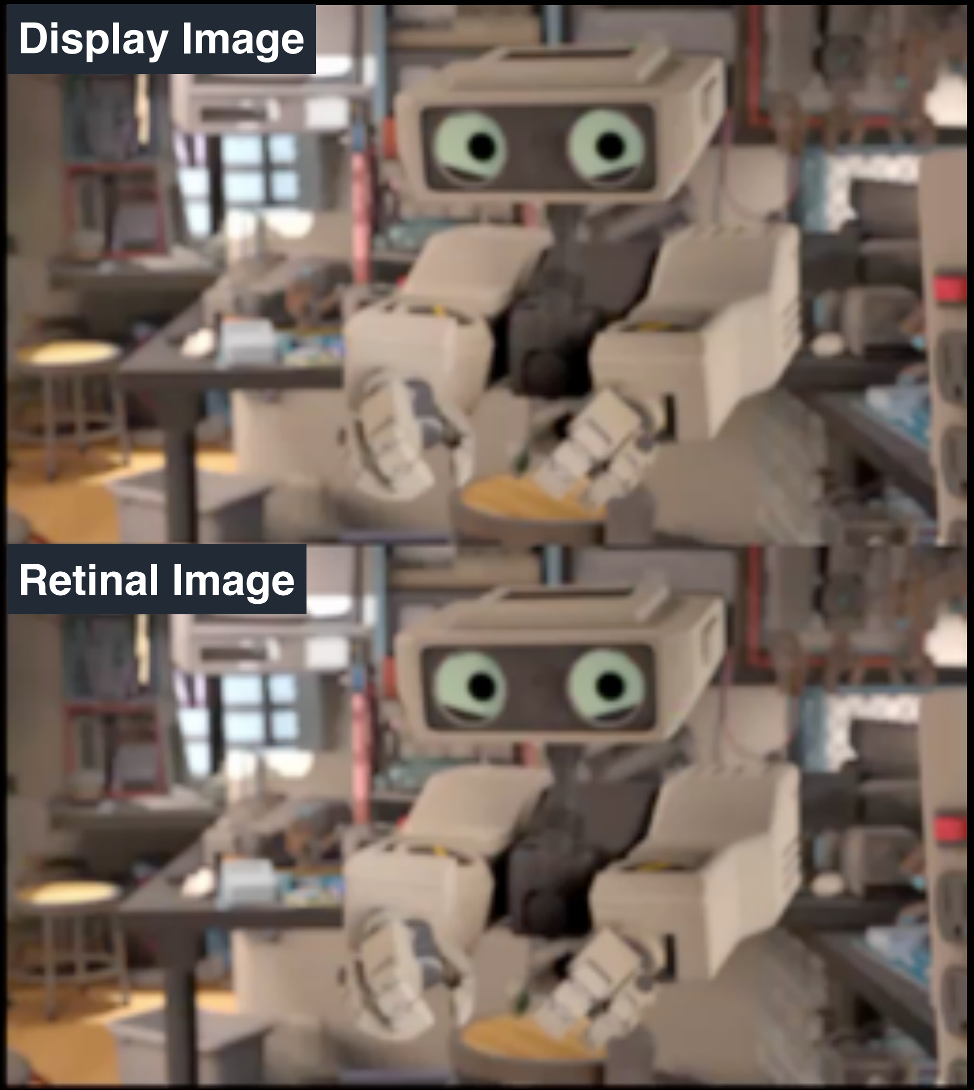
DeepFocus

with Lei Xiao, Alex Fix, Matt Chapman, Doug Lanman

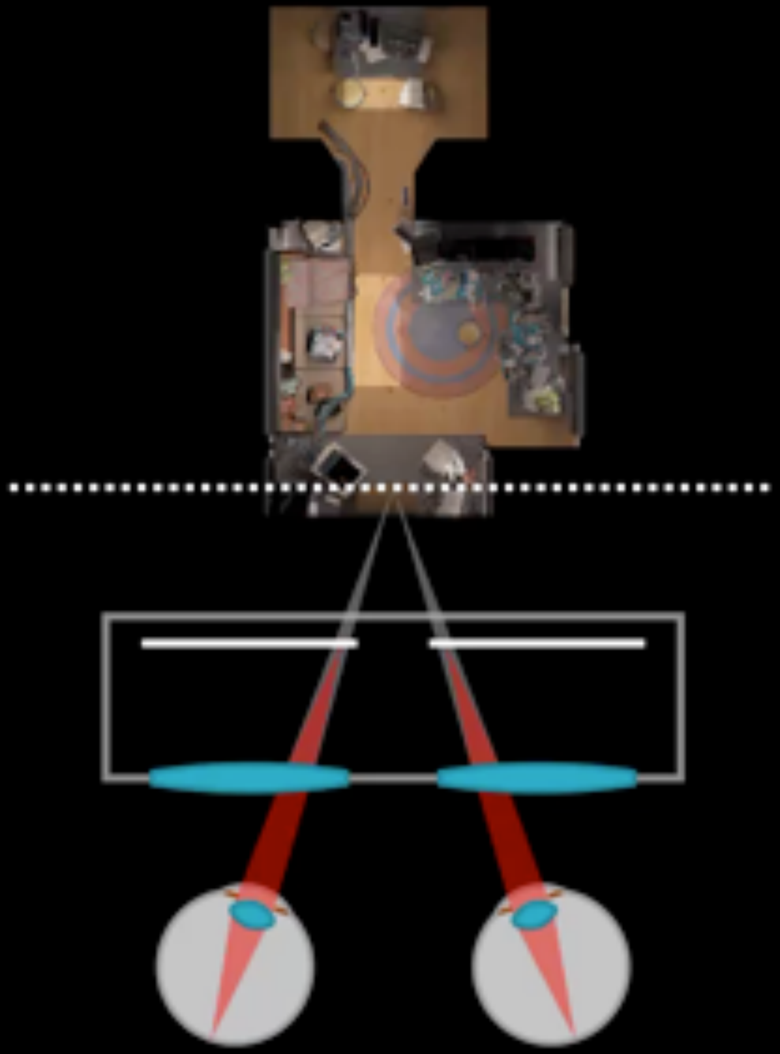
at Facebook Reality Labs

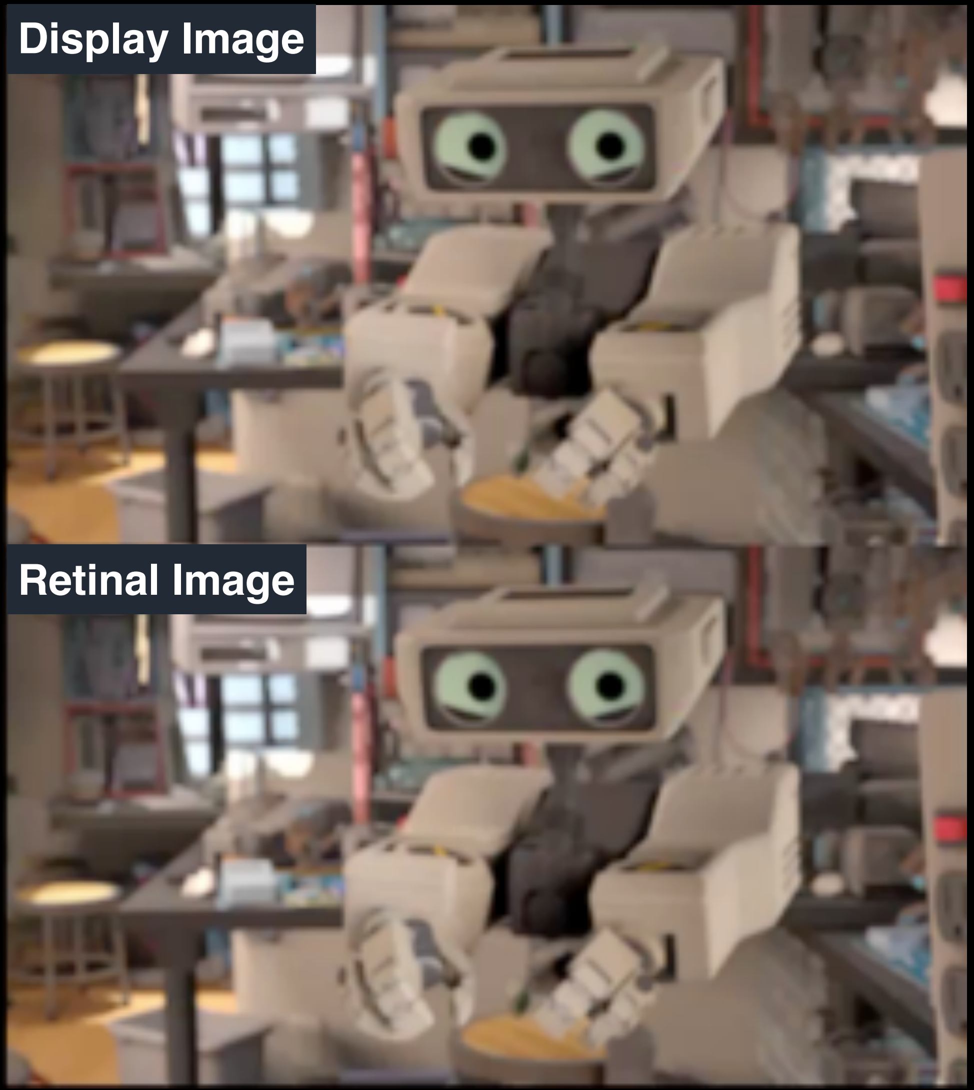
Gaze-Contingent Varifocal Display



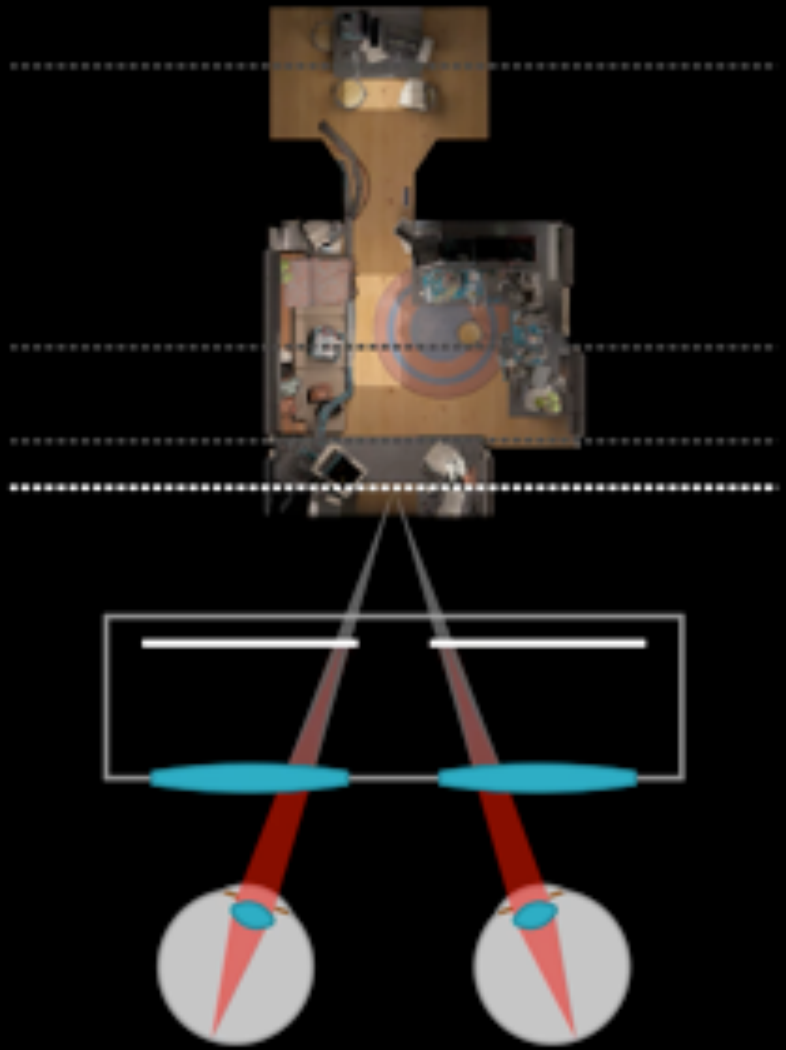


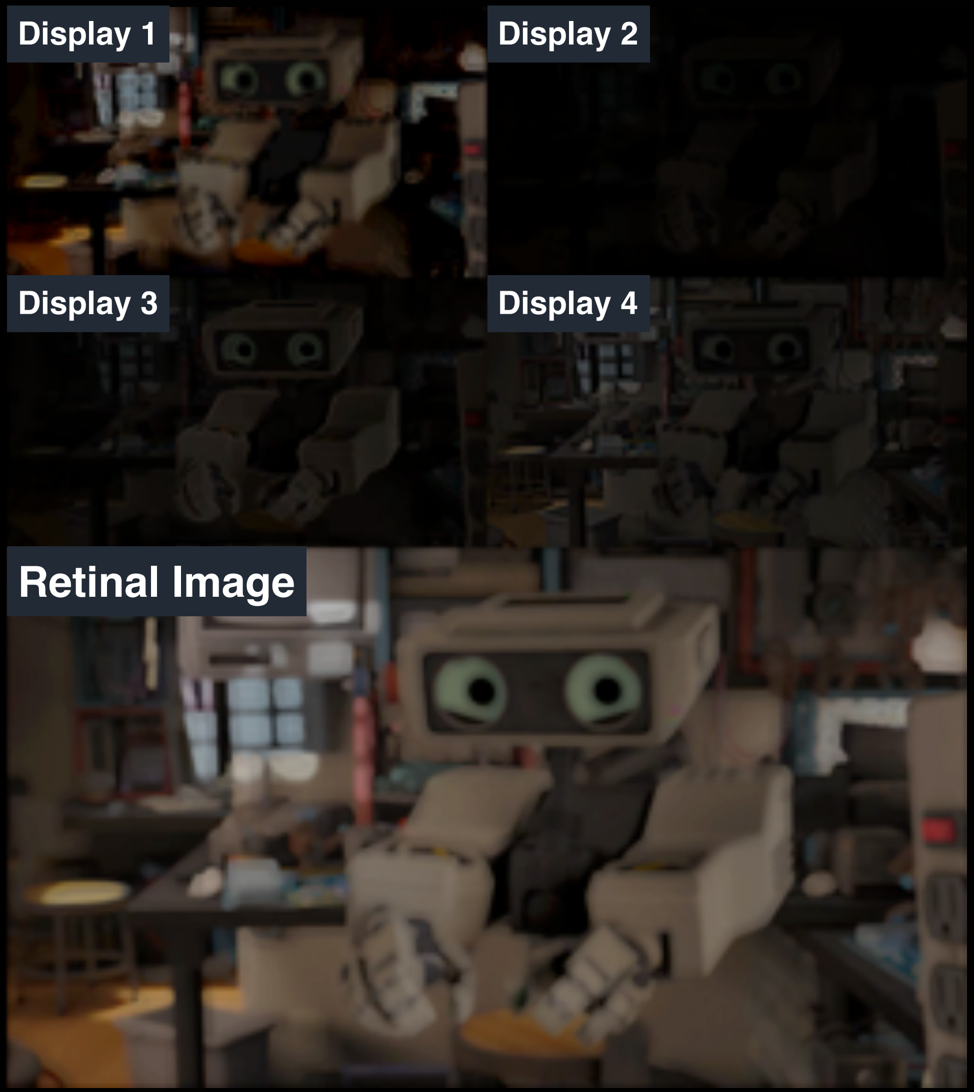
Gaze-Contingent Varifocal Display



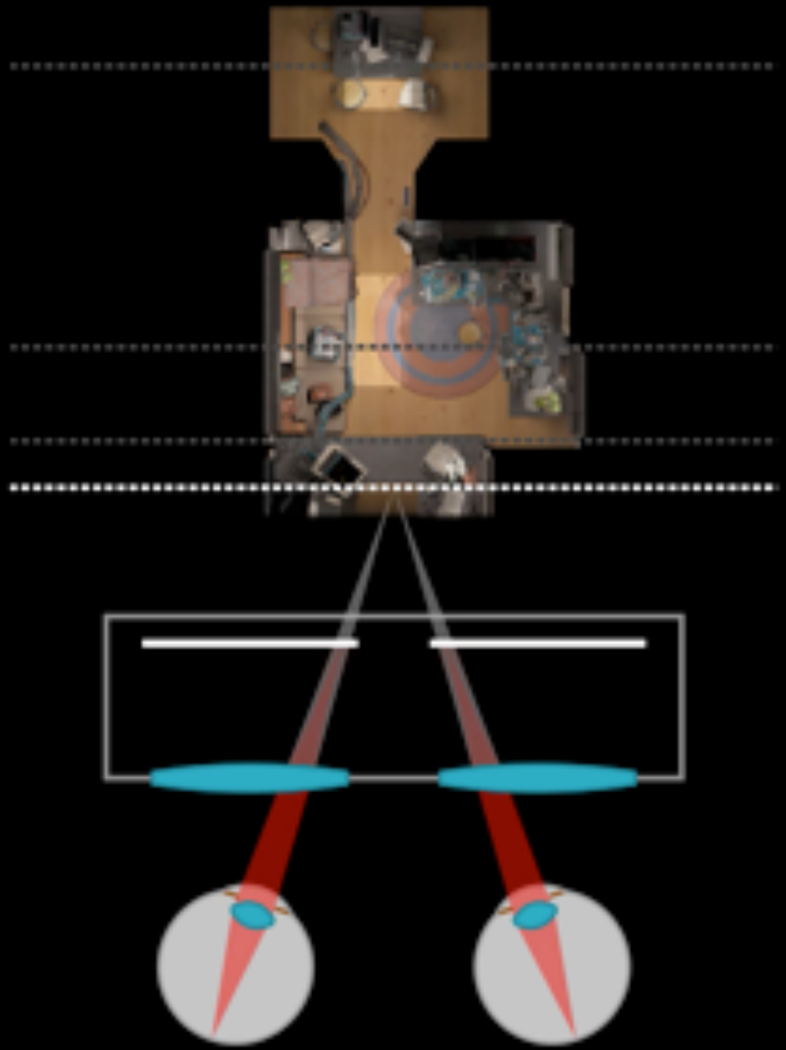


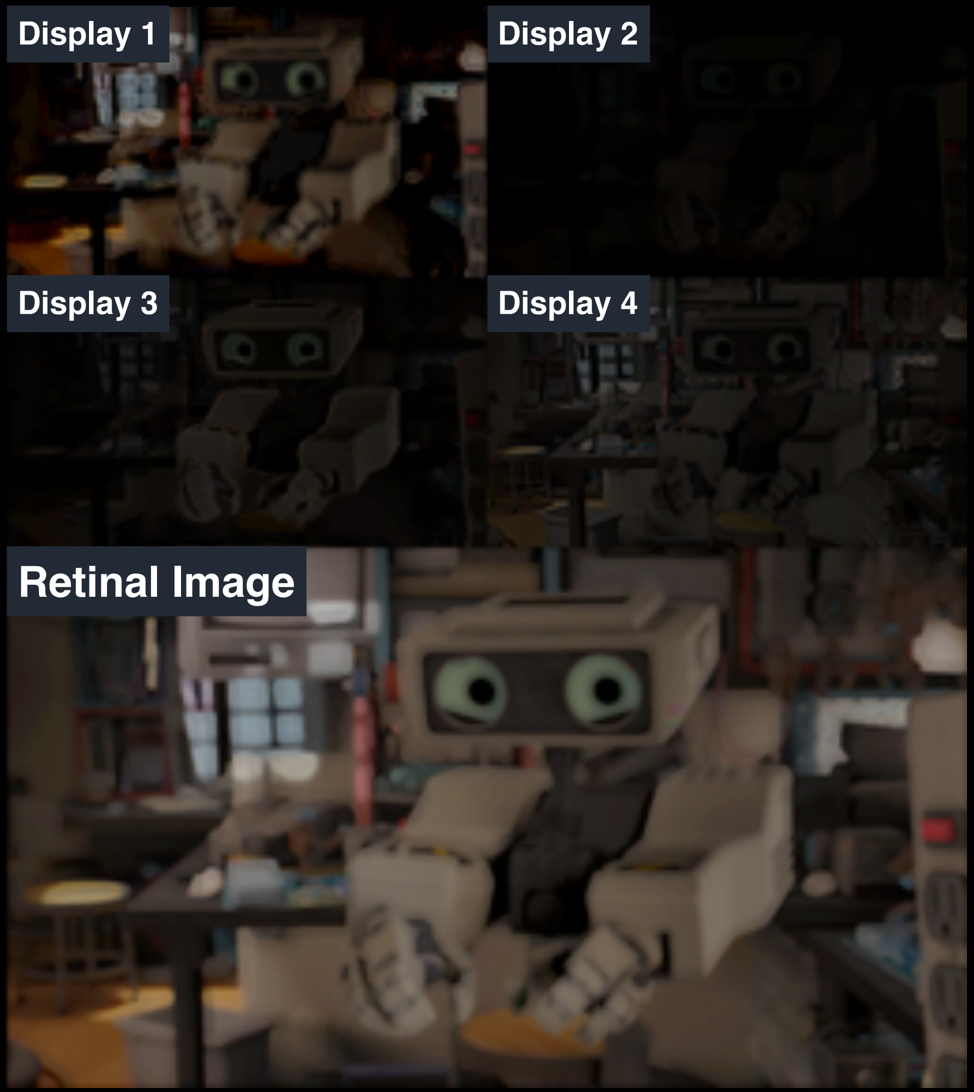
Multifocal Display



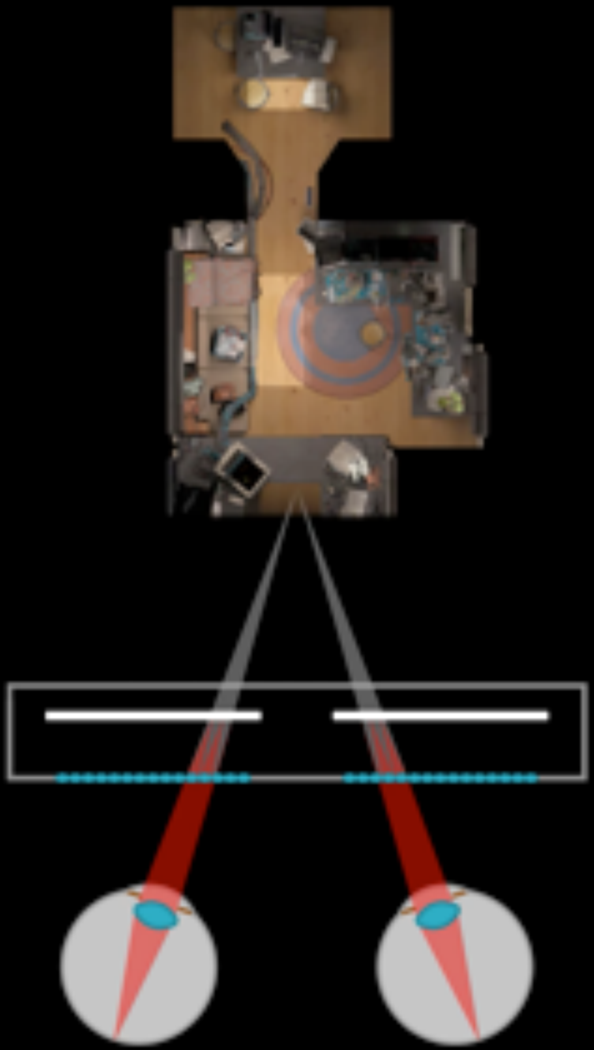


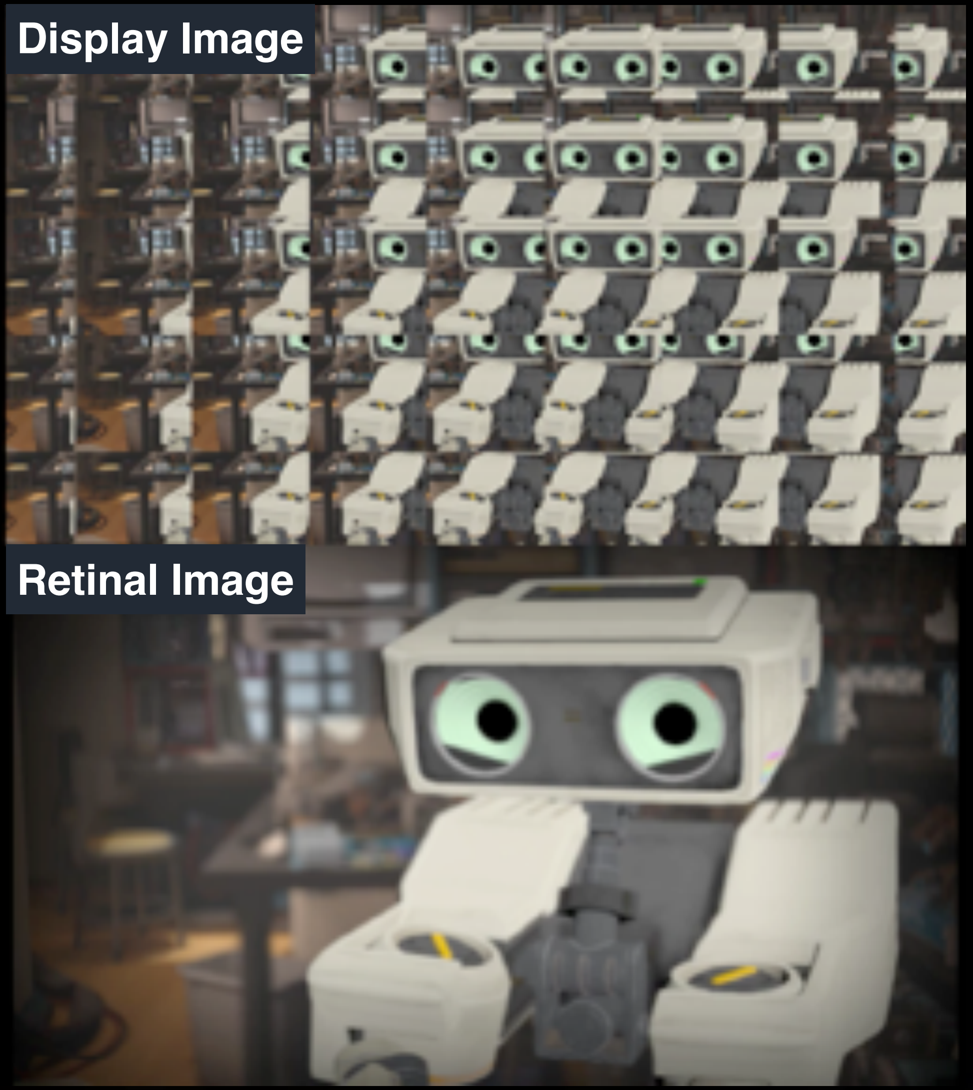
Multifocal Display



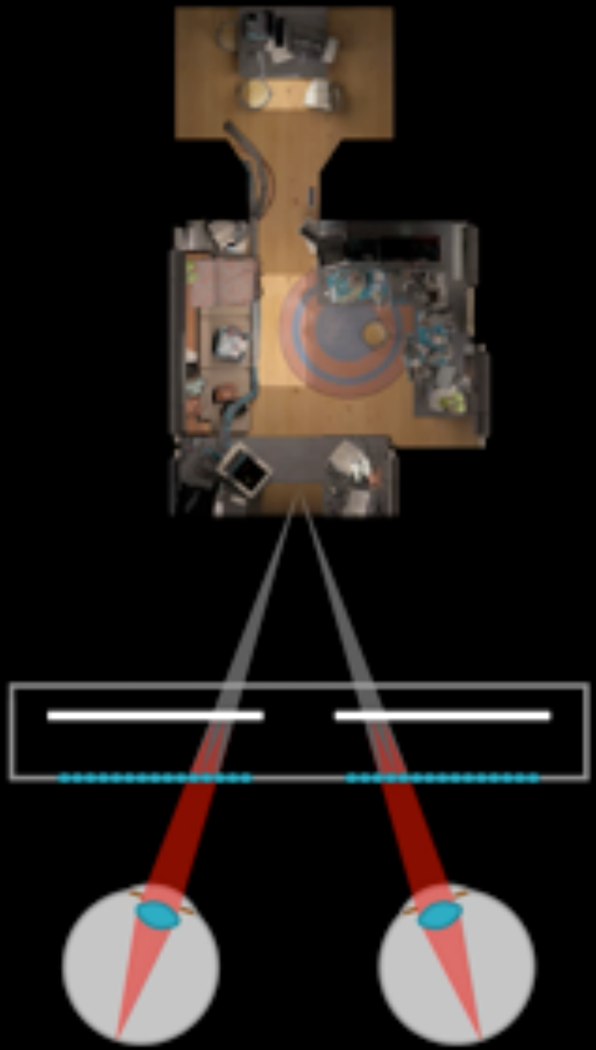


Near-Eye Light Field Display





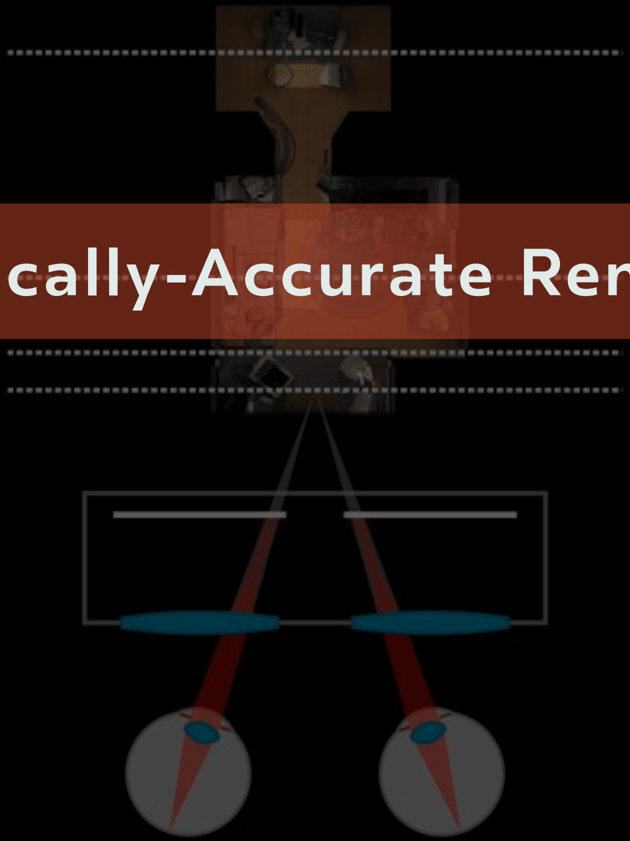
Near-Eye Light Field Display



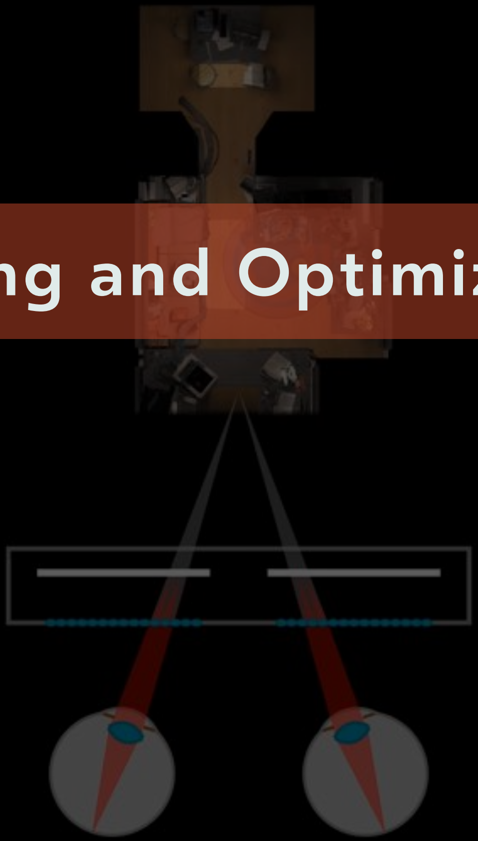
Accommodation-Supporting Displays

Challenge: Real-Time Physically-Accurate Rendering and Optimization

Gaze-Contingent Varifocal Display



Multifocal Display



Near-Eye Light Field Display

OUTPUT

Varifocal HMDs

Defocus Blur

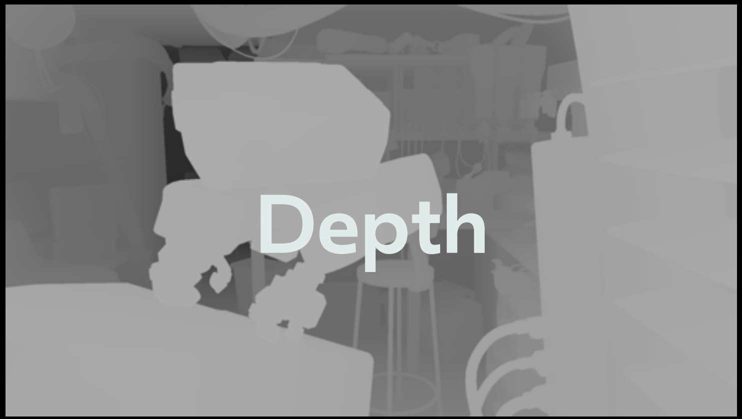
Multifocal HMDs

Multilayer Decompositions

Light Field HMDs

Multiview Imagery

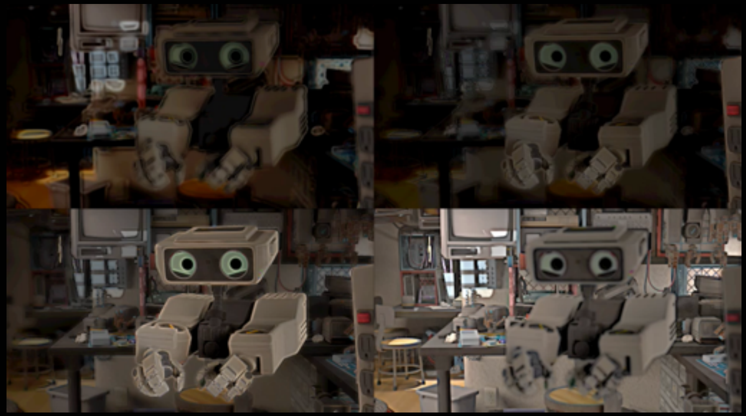
INPUT



OUTPUT

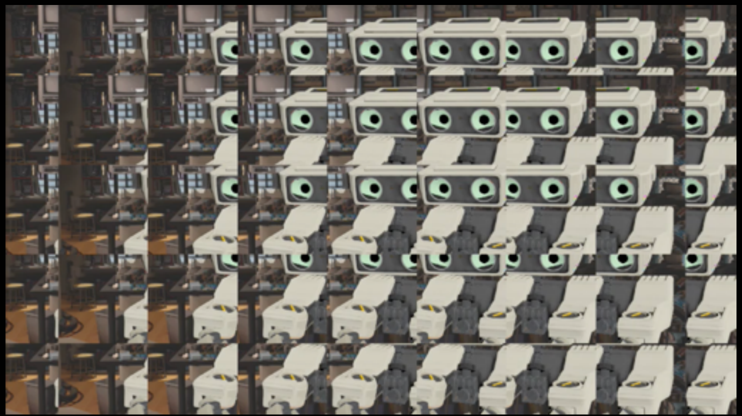
Varifocal HMDs

Defocus Blur



Multifocal HMDs

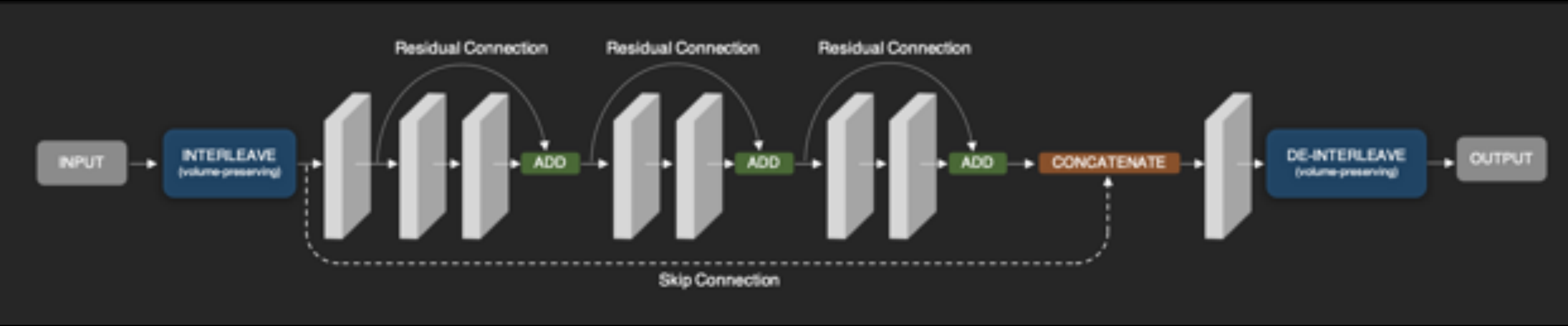
Multilayer Decompositions



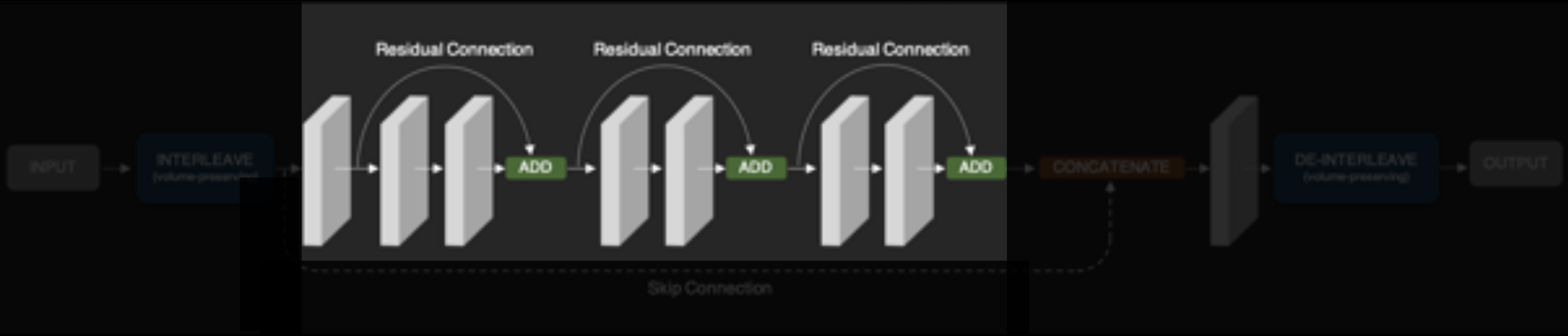
Light Field HMDs

Multiview Imagery

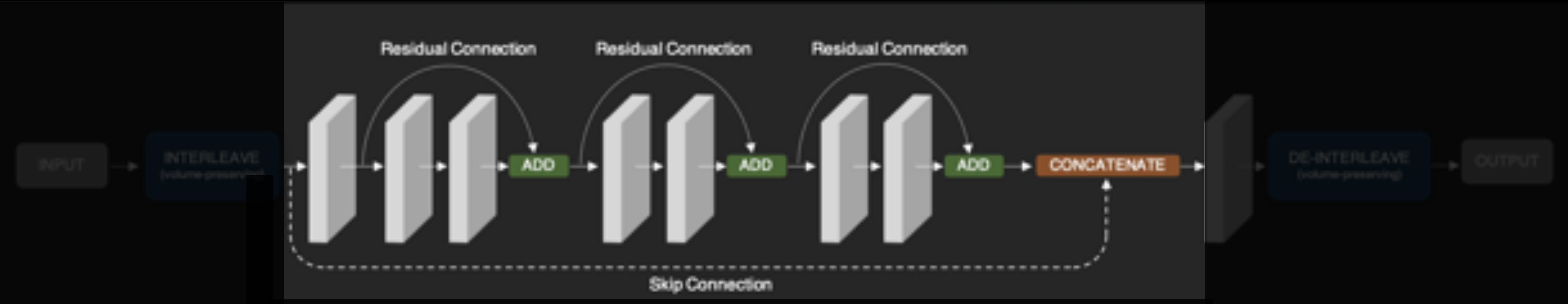
DeepFocus Network



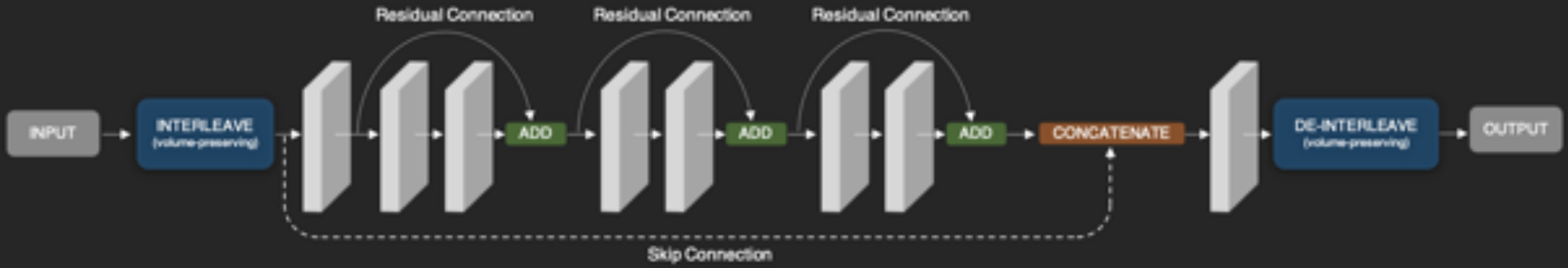
DeepFocus Network



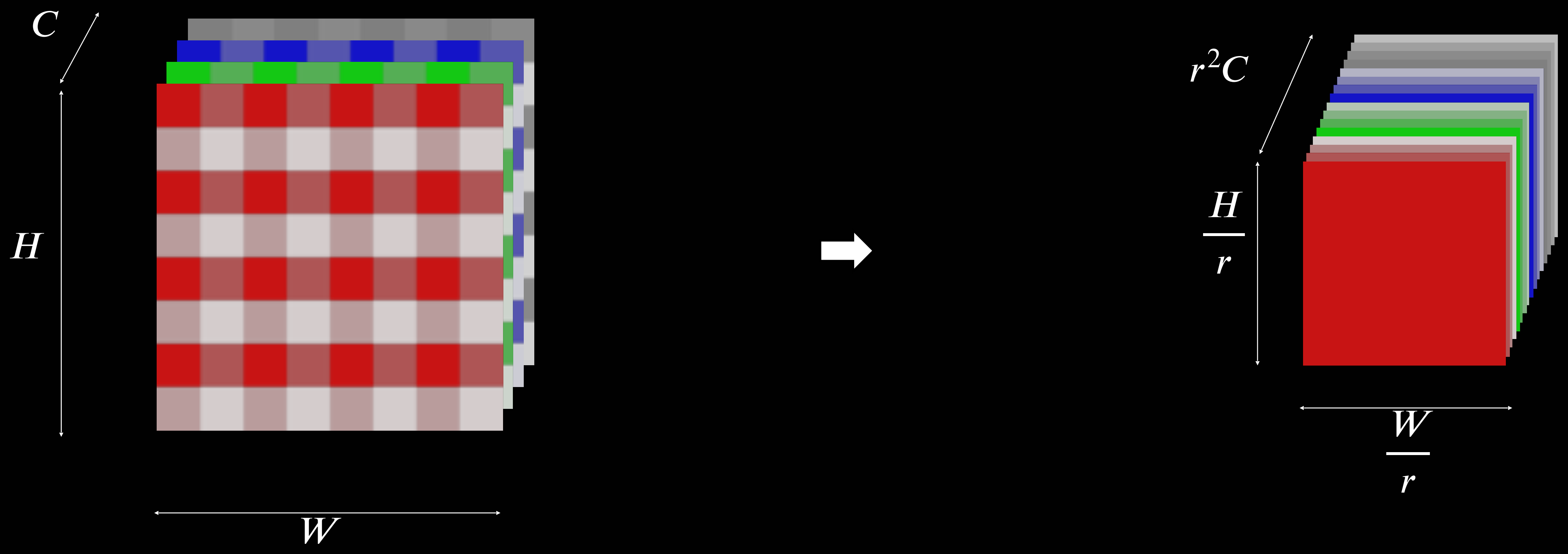
DeepFocus Network



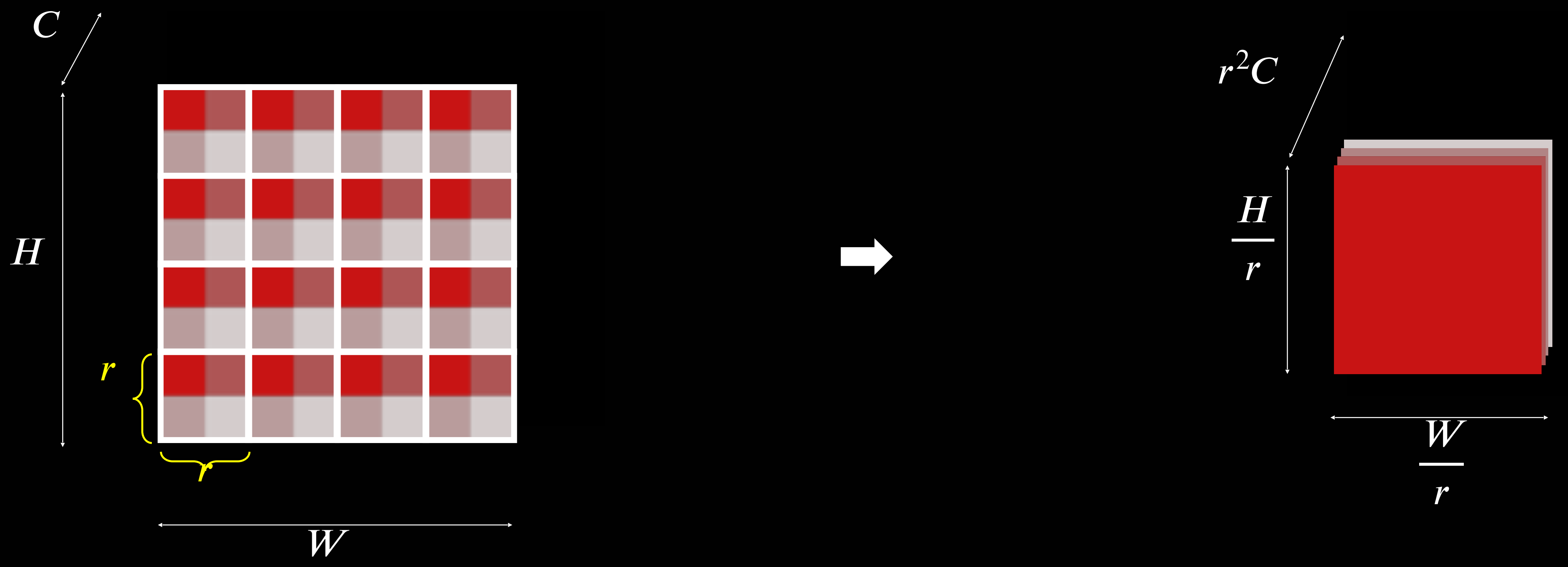
DeepFocus Network



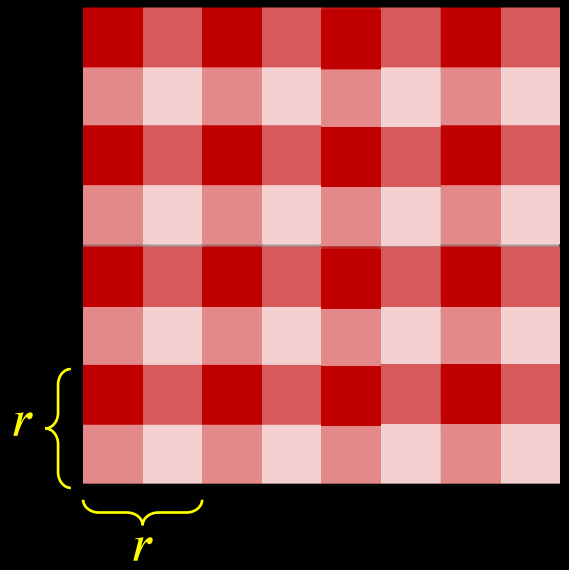
Volume-Preserving Interleaving Layer



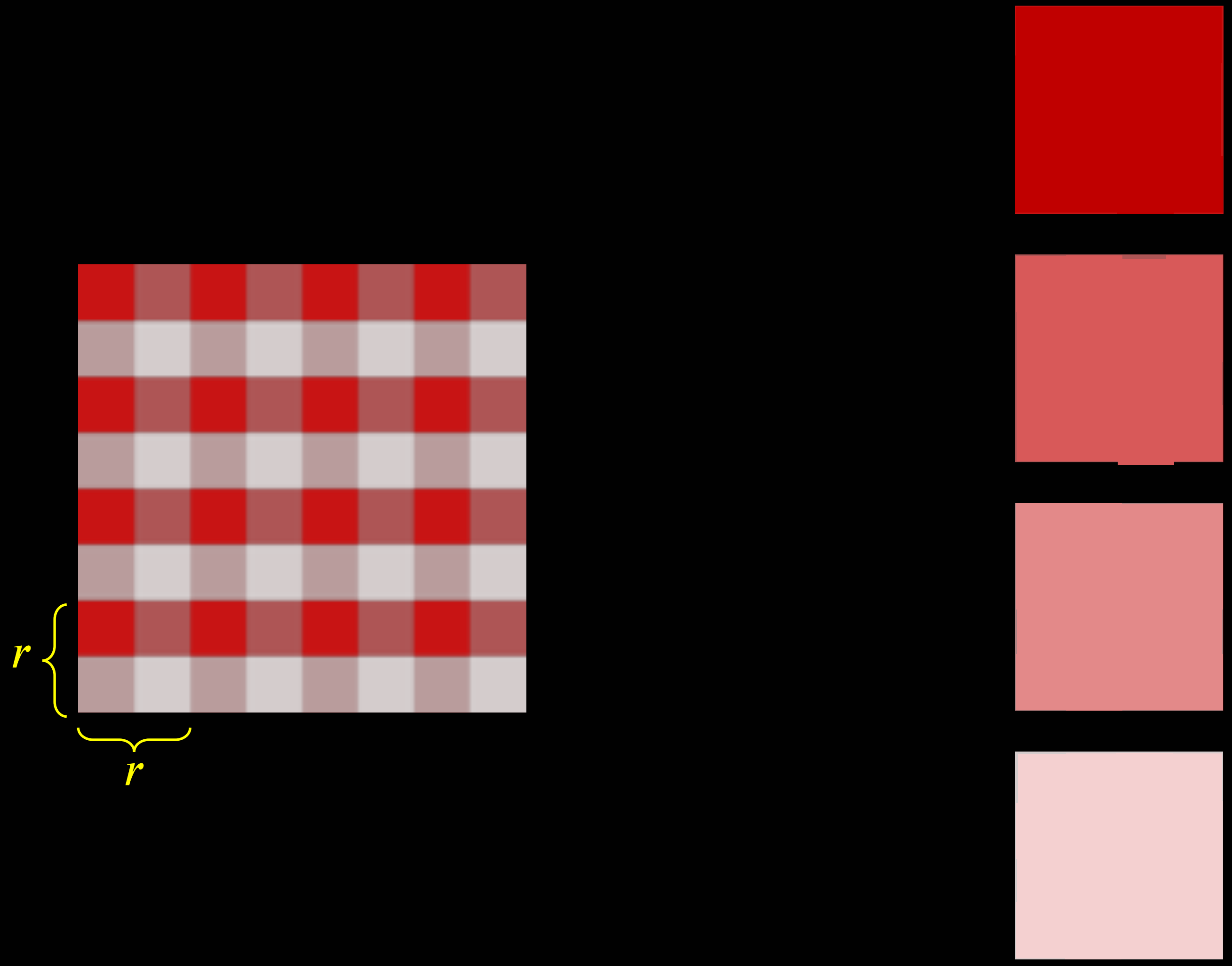
Volume-Preserving Interleaving Layer



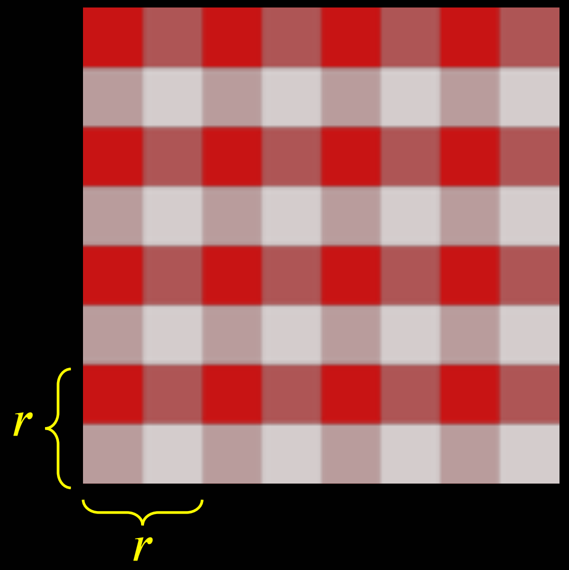
Volume-Preserving Interleaving Layer

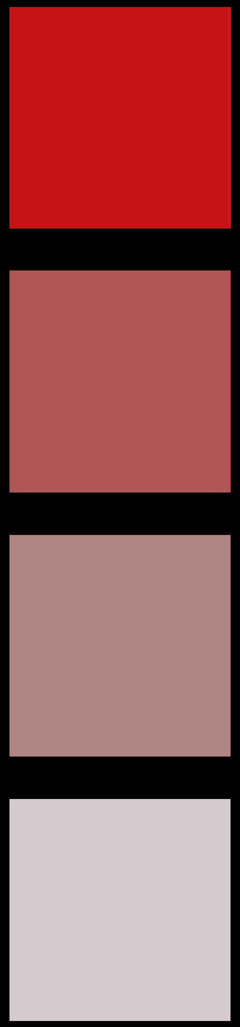


Volume-Preserving Interleaving Layer

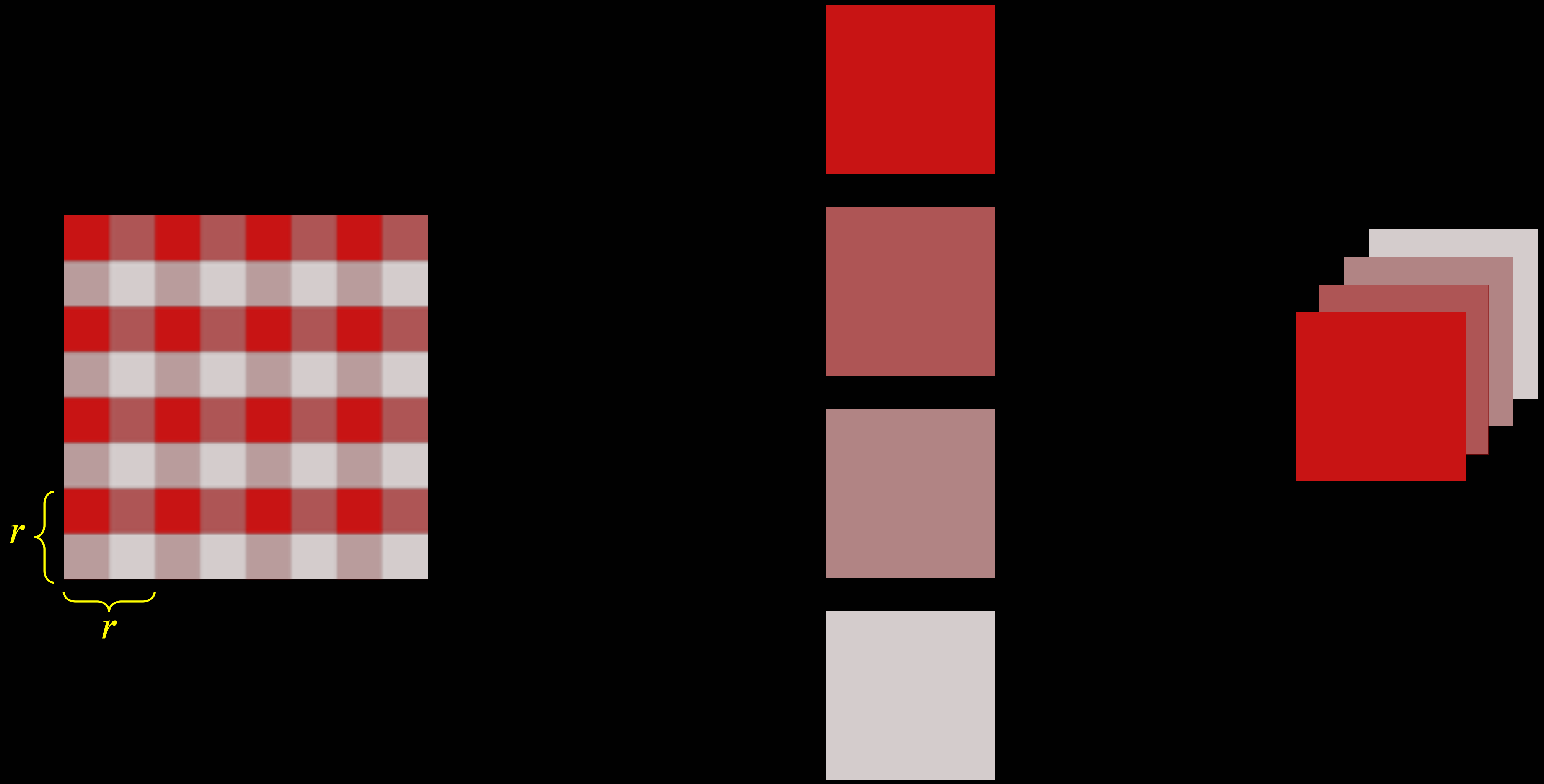


Volume-Preserving Interleaving Layer

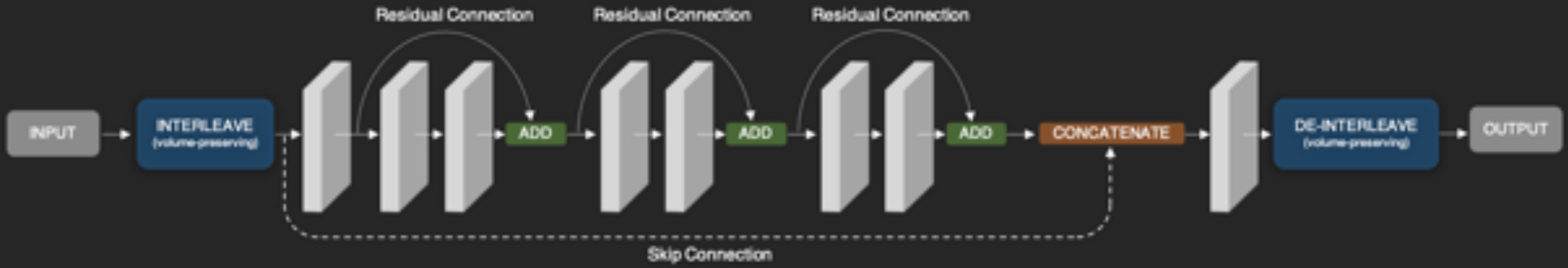


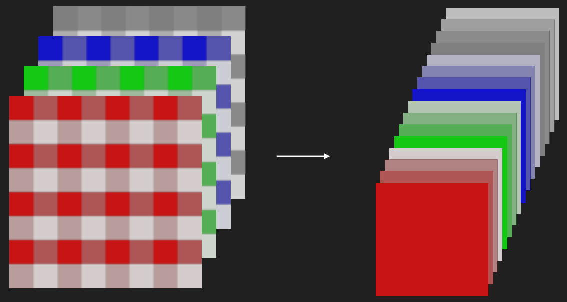


Volume-Preserving Interleaving Layer

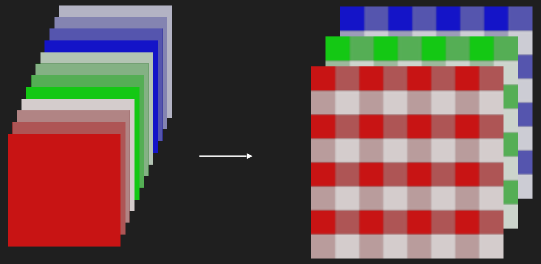


DeepFocus Network



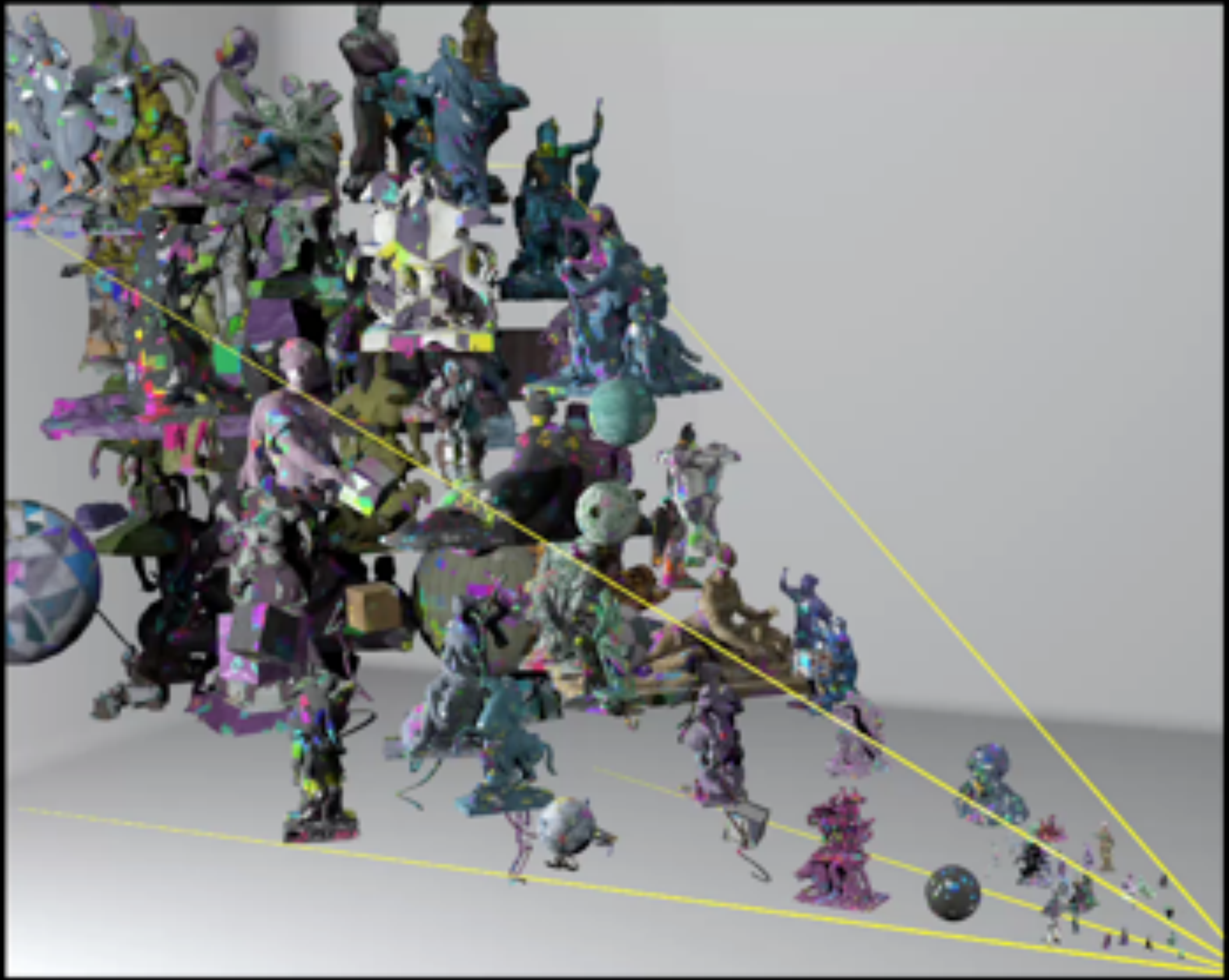


Volume-Preserving Interleaving



Volume-Preserving De-Interleaving

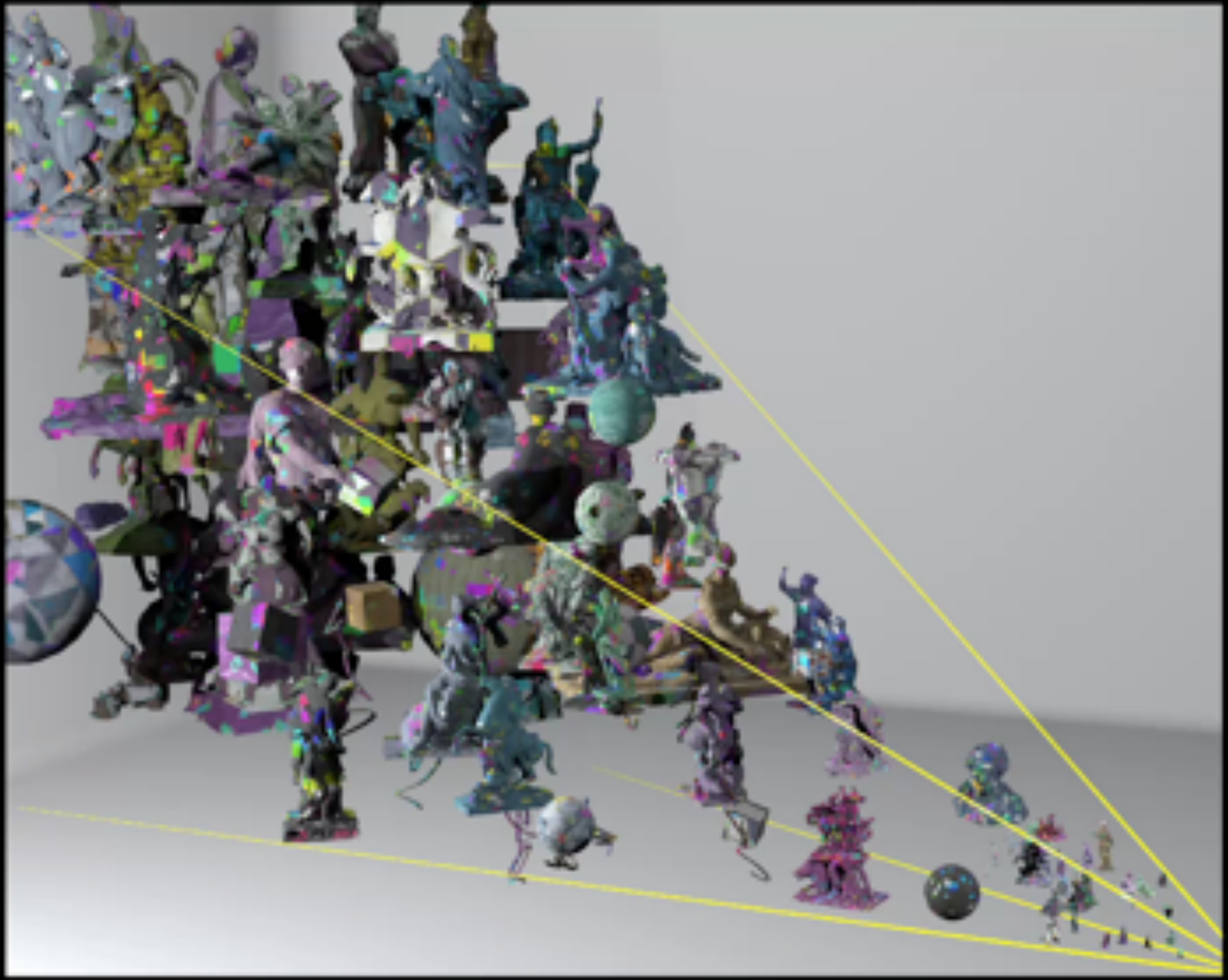
Training Dataset: Path-Traced Random 3D Scenes



Random 3D scenes

RGB-D

Training Dataset: Path-Traced Random 3D Scenes



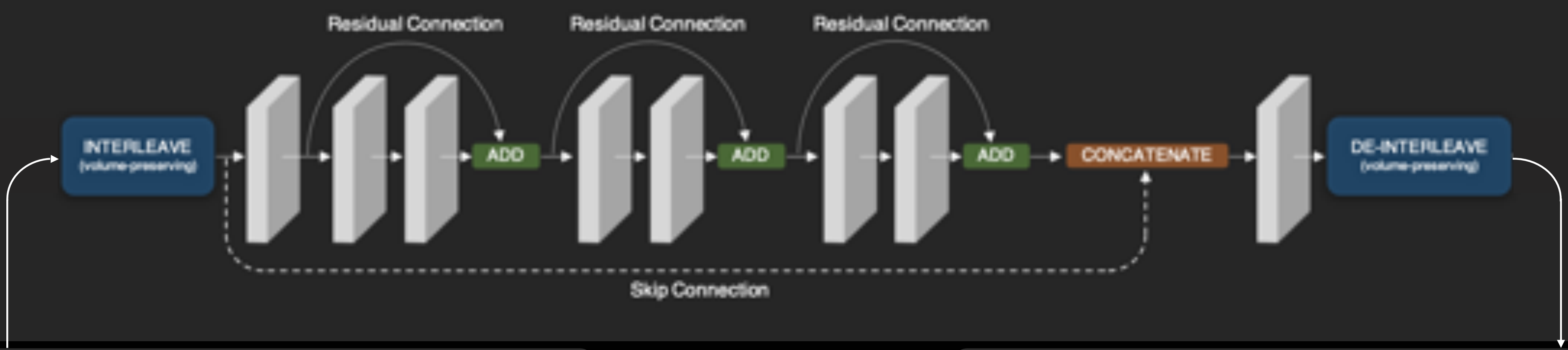
Random 3D scenes

RGB-D

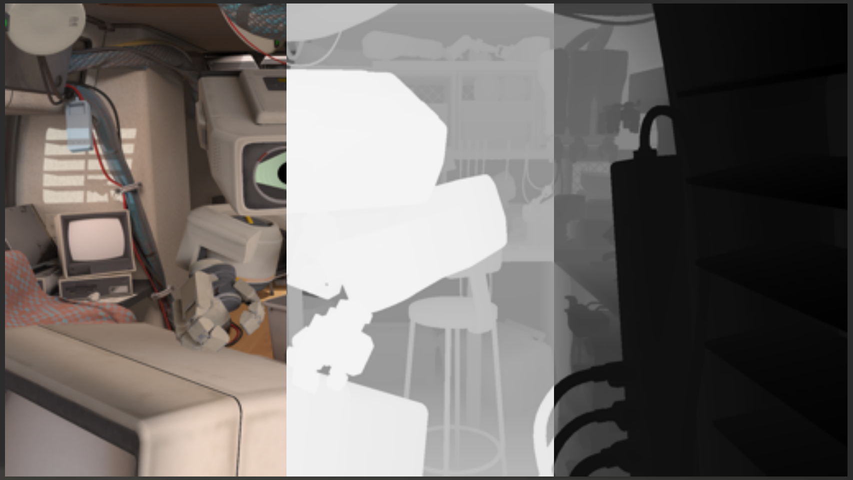
Application #1: Varifocal HMDs

Inferring Gaze-Contingent Defocus Blur from RGB-D

DeepFocus: Gaze-Contingent Defocus Blur from RGB-D



INPUT



RGB, Depth, CoC Map

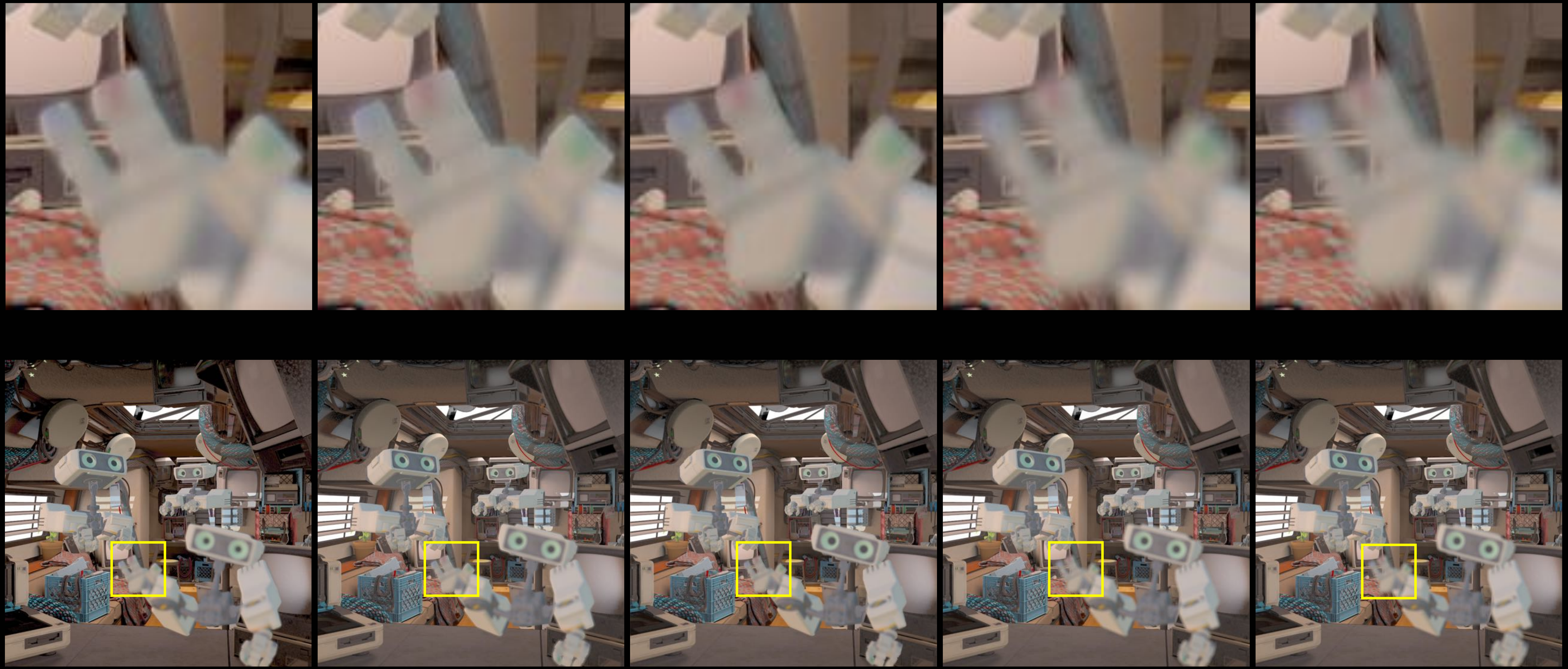
9.8ms, 1024x1024

OUTPUT

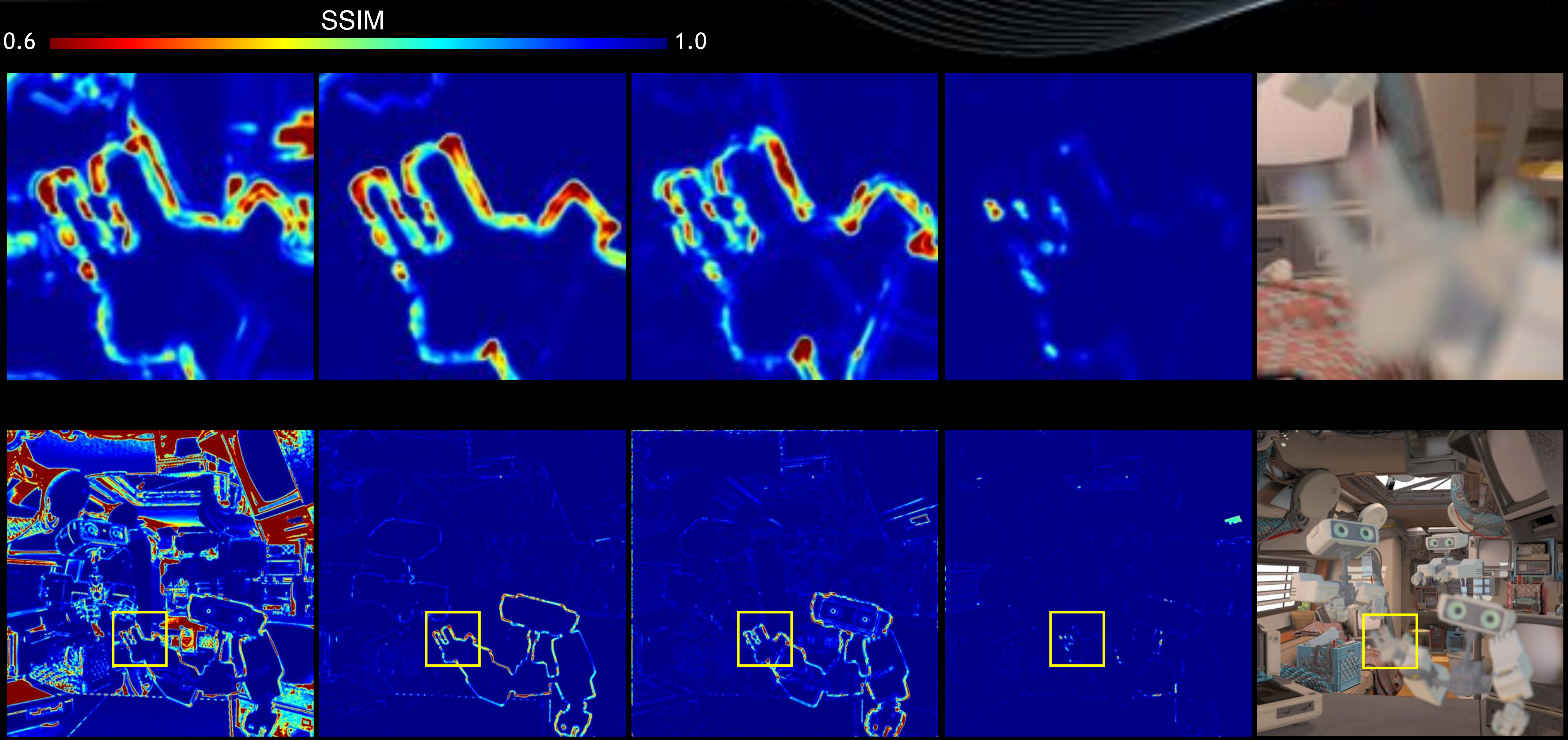
Gaze-Contingent Defocus Blur

Input RGB DeepFocus Ground truth

Input RGB DeepFocus Ground truth



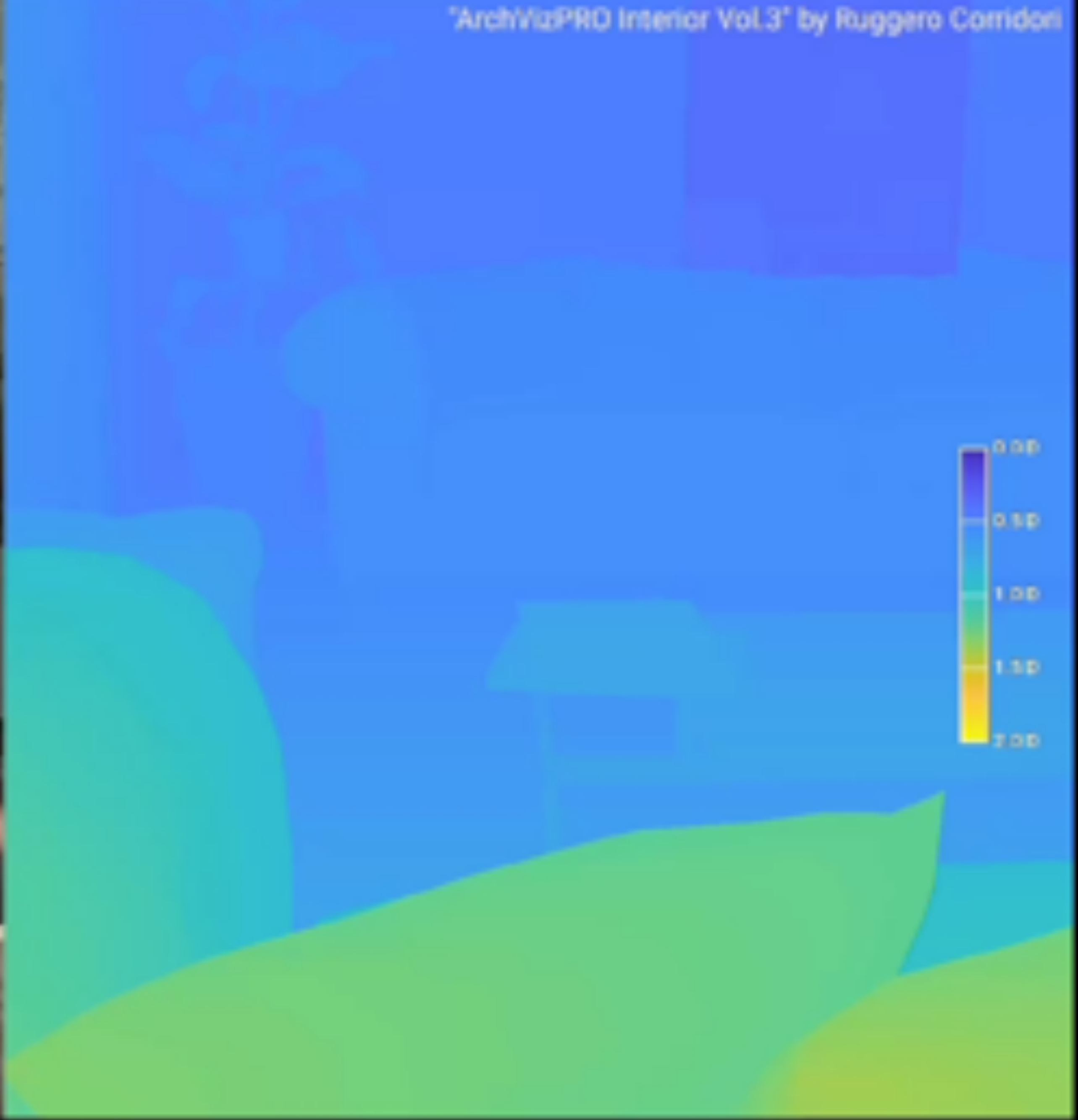
Unity	Nuke	Nalbach et al. 2017	DeepFocus	Ground truth
25.3dB	40.1dB	37.0dB	45.6dB	



Unity | Nuke | Nalbach et al. 2017 | DeepFocus | Ground truth

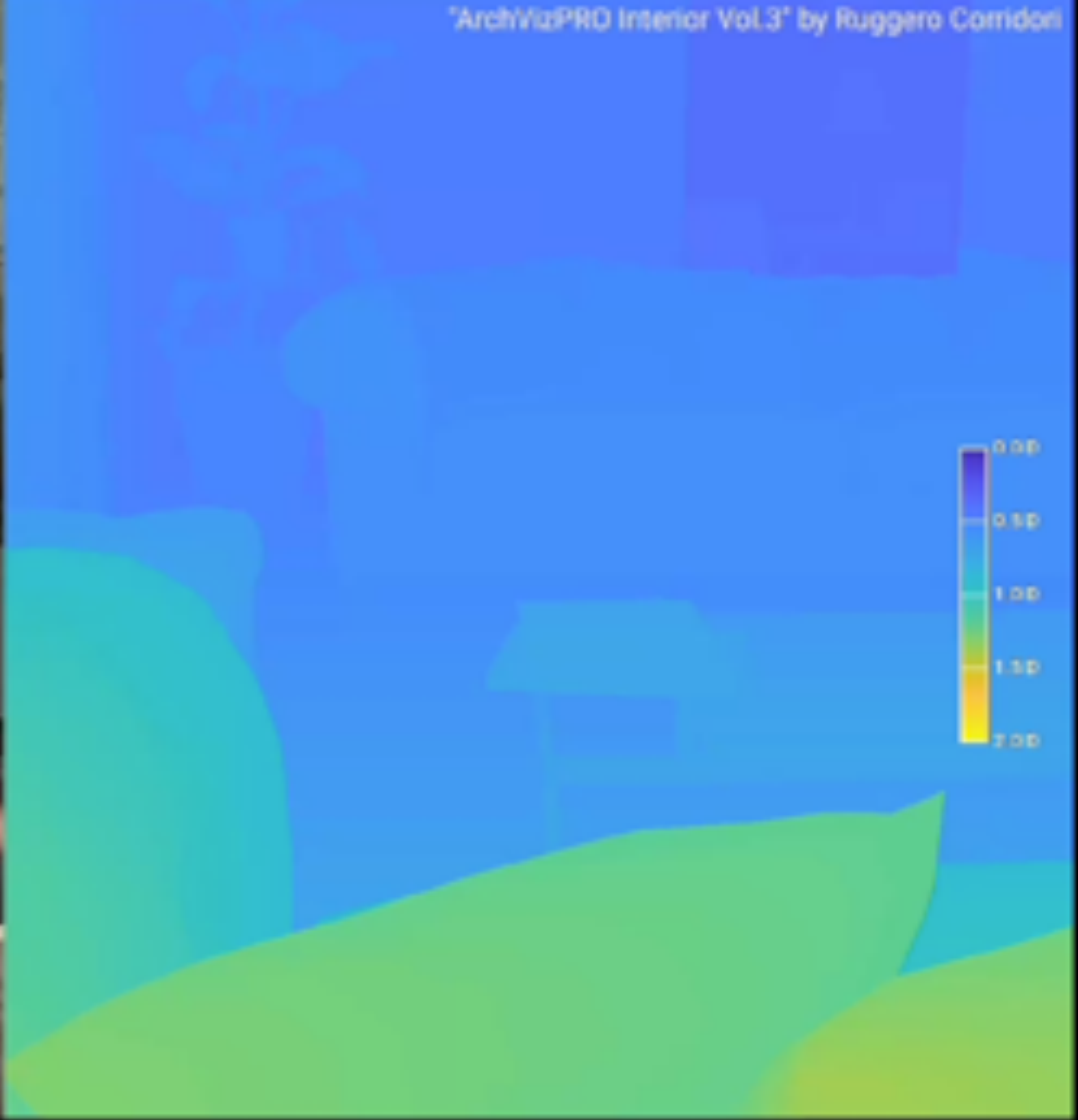
0.887 | 0.993 | 0.989 | 0.999

Input Color Image



Input Depth Map

Input Color Image

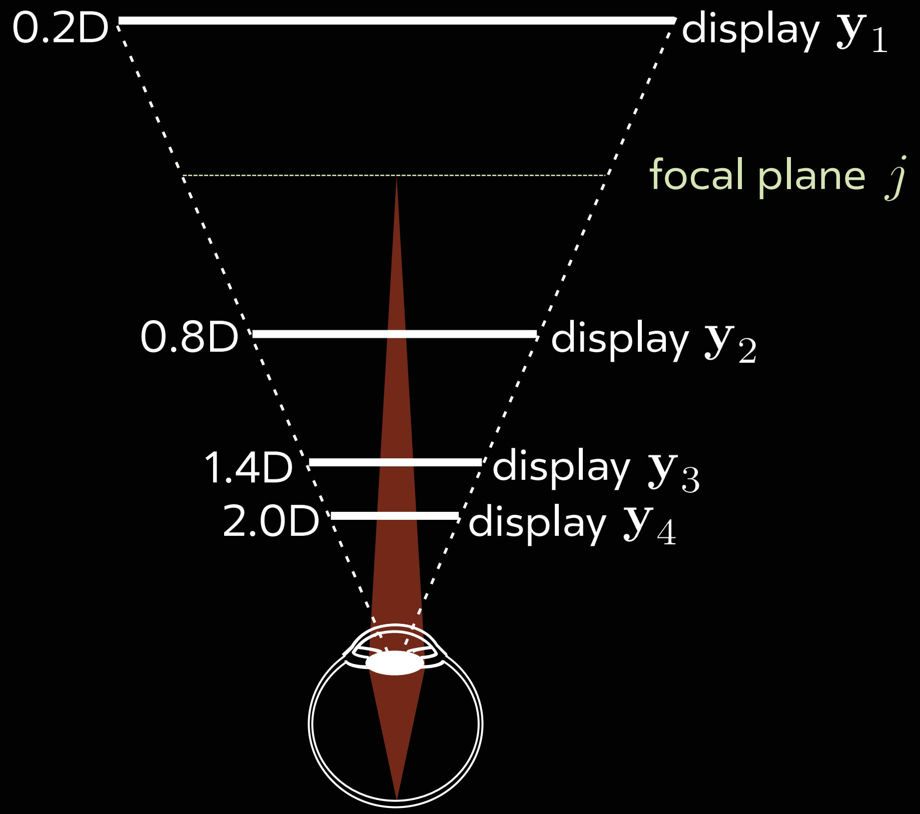


Input Depth Map

Application #2: Multifocal HMDs

Inferring Focal Stack and Multilayer Decomposition

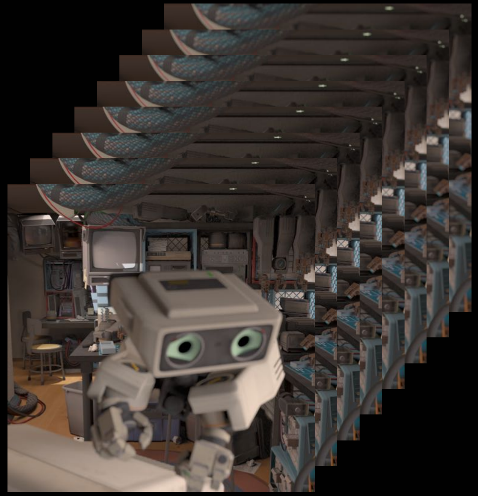
Multifocal Displays



$$\overbrace{\mathbf{z}_j}^{\text{Retinal Image}} = \sum_{i=1}^{4} \underbrace{\mathbf{k}_{ij}}_{\text{Kernel}} * \overbrace{\mathbf{y}_i}^{\text{Display Image}}$$

Multifocal Displays

3D scene

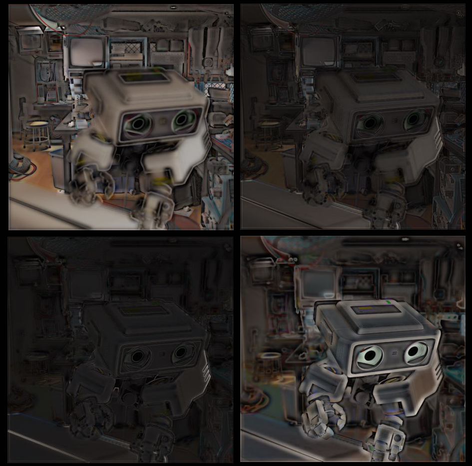


Render Dense Focal Stacks

$$\mathbf{y}_i = \underset{\mathbf{y}_i}{\operatorname{argmin}} \sum_{j=1}^{N} ||\mathbf{z}_j - \sum_{i=1}^{M} \mathbf{k}_{ij} * \mathbf{y}_i||_2^2,$$

$$s.t. \quad 0 \leq \mathbf{y}_i \leq 1, \quad i = 1, 2, ..., M$$

Solve Iterative Optimization



Optimized Multilayers

[Narain et al. 2015, Mercier et al. 2017]

Multifocal Displays

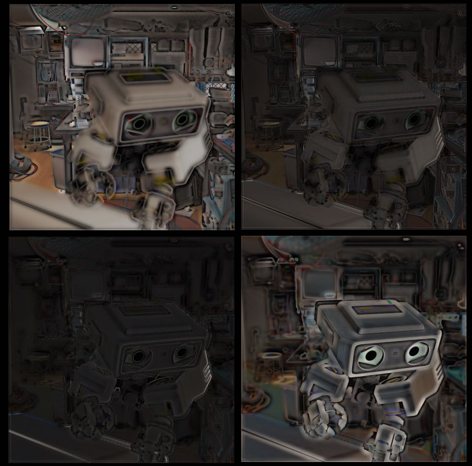
3D scene

Render Dense Focal Stacks

COMPUTATIONALLY EXPENSIVE

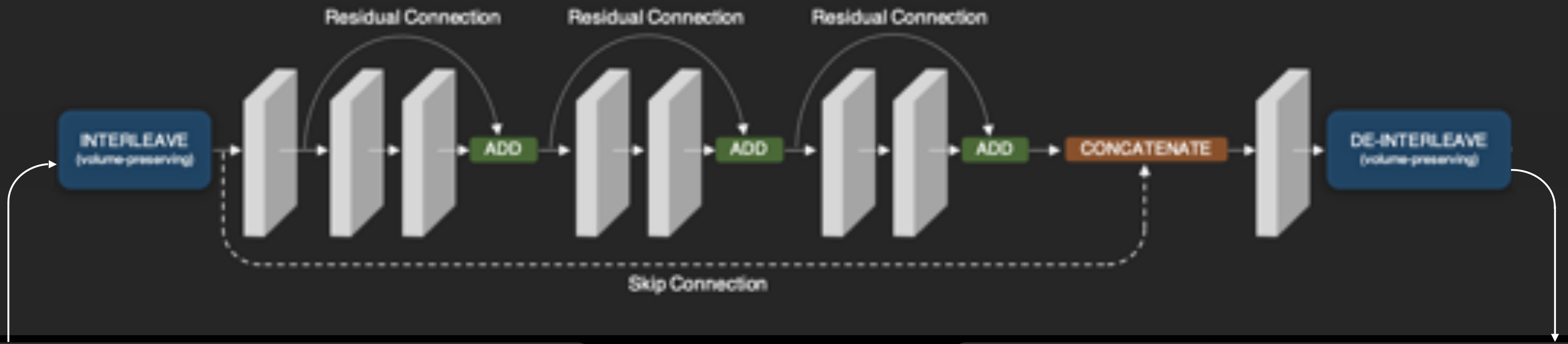
$$s.t. \quad 0 \leq \mathbf{y}_i \leq 1, \quad i = 1, 2, ..., M$$

Solve Iterative Optimization



Optimized Multilayers

[Narain et al. 2015, Mercier et al. 2017]



INPUT

RGB and Depth

10.0ms, 1024x1024

OUTPUT

Multilayer Decomposition

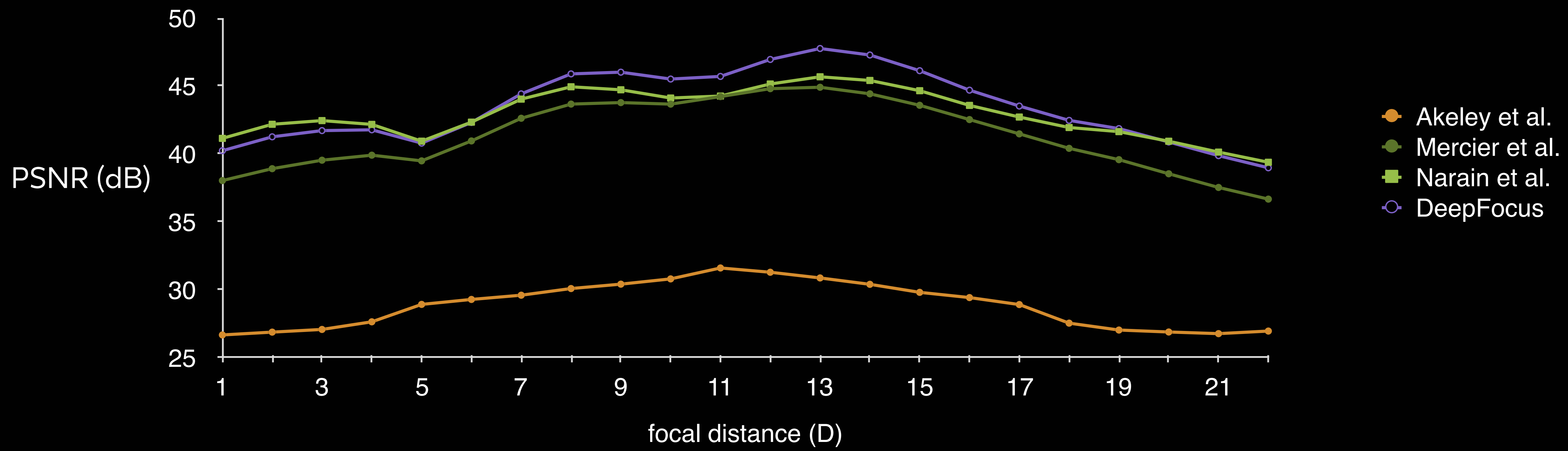
Akeley et al. [2004]

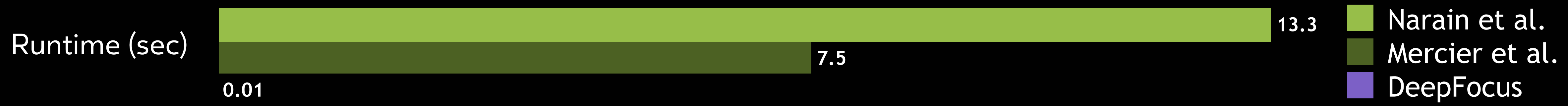
DeepFocus

Akeley et al. [2004]

DeepFocus

DeepFocus: Multilayer Decomposition from Dense Focal Stacks





Application #3: Light Field HMDs

Inferring Multiple Viewpoints

Near-Eye Light Field Displays

Input Light Field

81 RGB-D Images

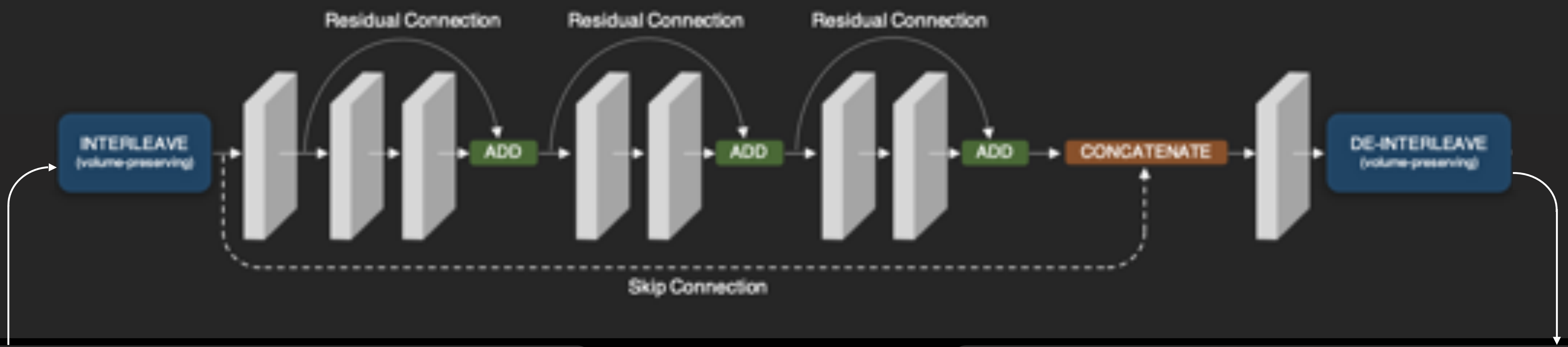
Simulated Retinal Image

Near-Eye Light Field Displays

Input Light Field

81 RGB-D Images

Simulated Retinal Image



INPUT

Sparse RGB and Depth

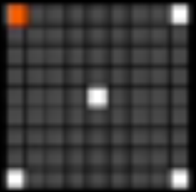
19.7ms, 81x512x512

OUTPUT

Multiview Imagery

View Interpolation from 5 RGB-D Images

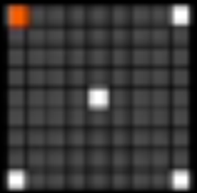
Target RGB Image



Inferred RGB Image

View Interpolation from 5 RGB-D Images

Target RGB Image



Inferred RGB Image

Limitations and Conclusion

RGB-D Limitations

Input RGB

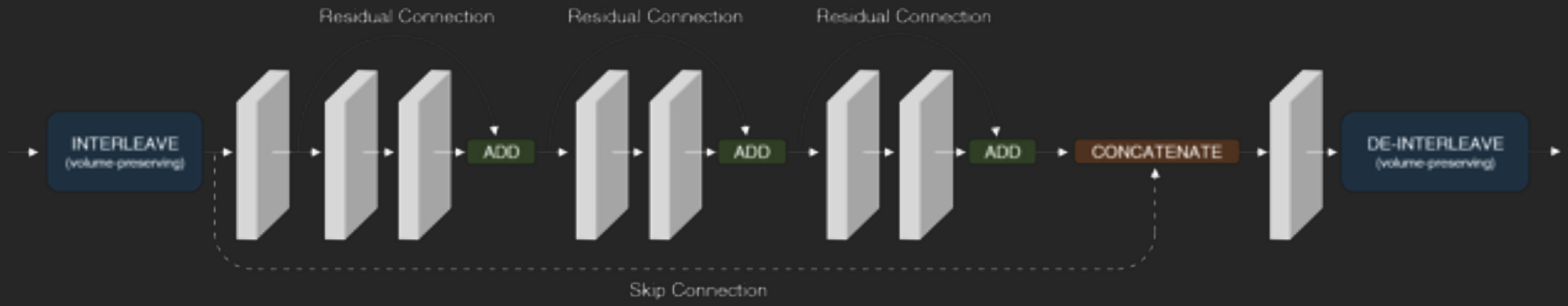
Input Depth

RGB-D Limitations

DeepFocus (near focus)

Ground Truth (near focus)

Conclusion



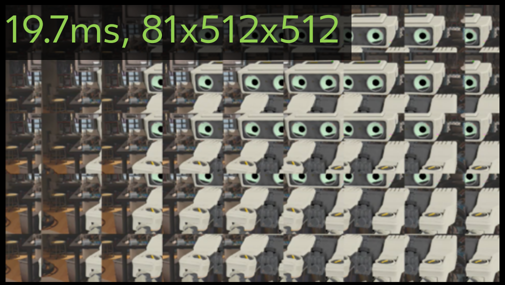
DeepFocus

Varifocal HMDs

Defocus Blur

Multifocal HMDs

Multilayer Decompositions



Light Field HMDs

Multiview Imagery

Outlook

Machine Learning: Challenges

- Easy to get to 80%, very hard to get to 95%
- Not a silver bullet!
  - Inversion is hard
  - Validation/coverage is hard
    - Worst case accuracy?
  - Hyperparameters!
    - Keep your experiments organized
    - Needs more compute

Machine Learning: Applications

- New framework for rendering
  - Approximation
  - Compression
  - Learning distribution
- Closest to human perception

- Differentiable programming as a generic optimization framework for existing methods, e.g. see [Li18]

Thank You!

Opportunities in Redmond, WA:

- Research Scientist, Machine Learning and Graphics
- Research Scientist, Materials and Multiscale Appearance
- Postdoctoral Research Scientist, Graphics
- Graphics Compression Lead
- Cloud Streaming Network Engineer
- **PhD 2019 Internships**

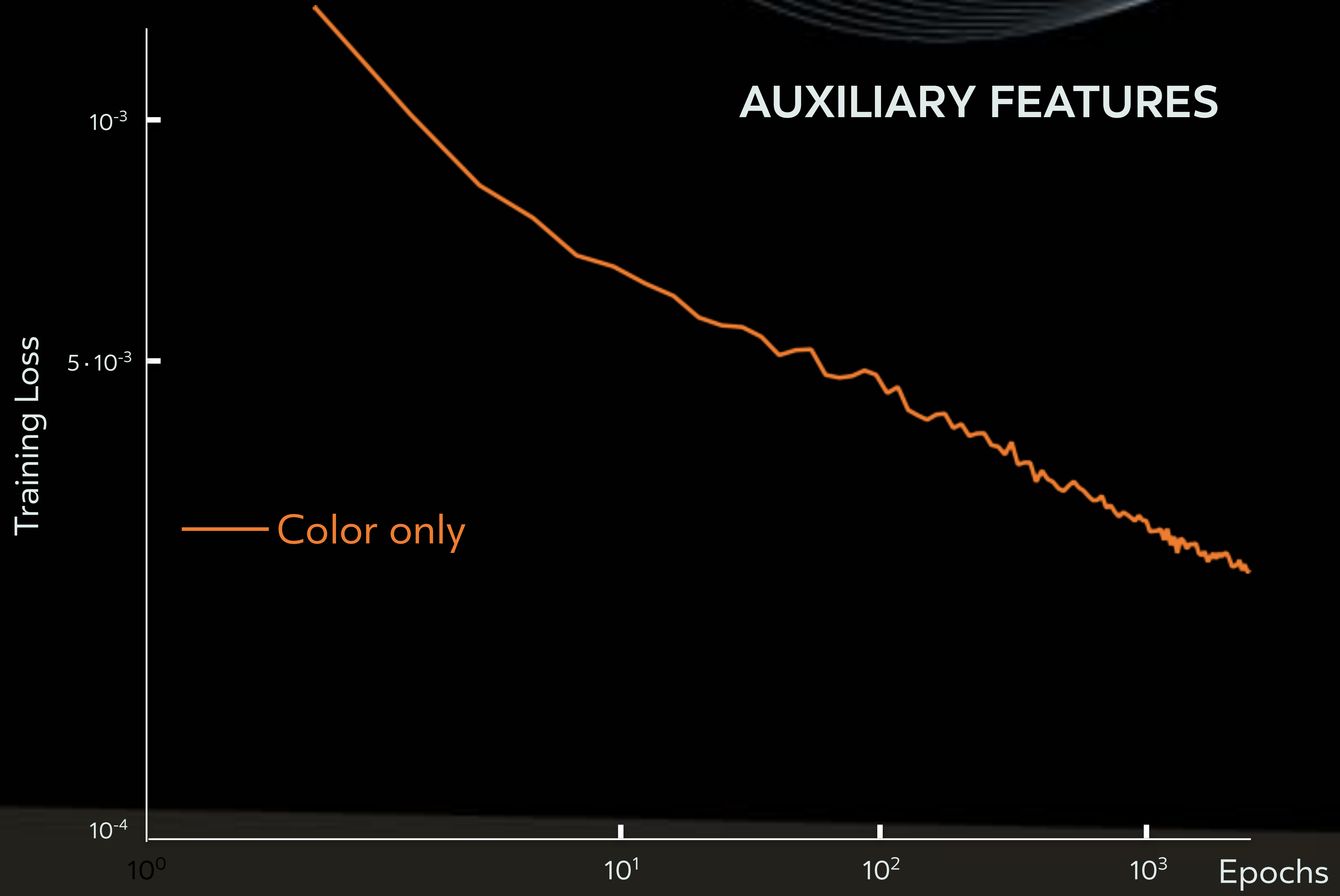
Graphics team at Facebook Reality Labs is seeking researchers and engineers for next generation graphics for virtual and augmented reality: ray tracing, metaverse ecosystem, perceptual rendering, and machine learning.

Contact me (anton.kaplanyan@oculus.com) or Nicole Doyle (nicole.doyle@oculus.com) if you are interested.

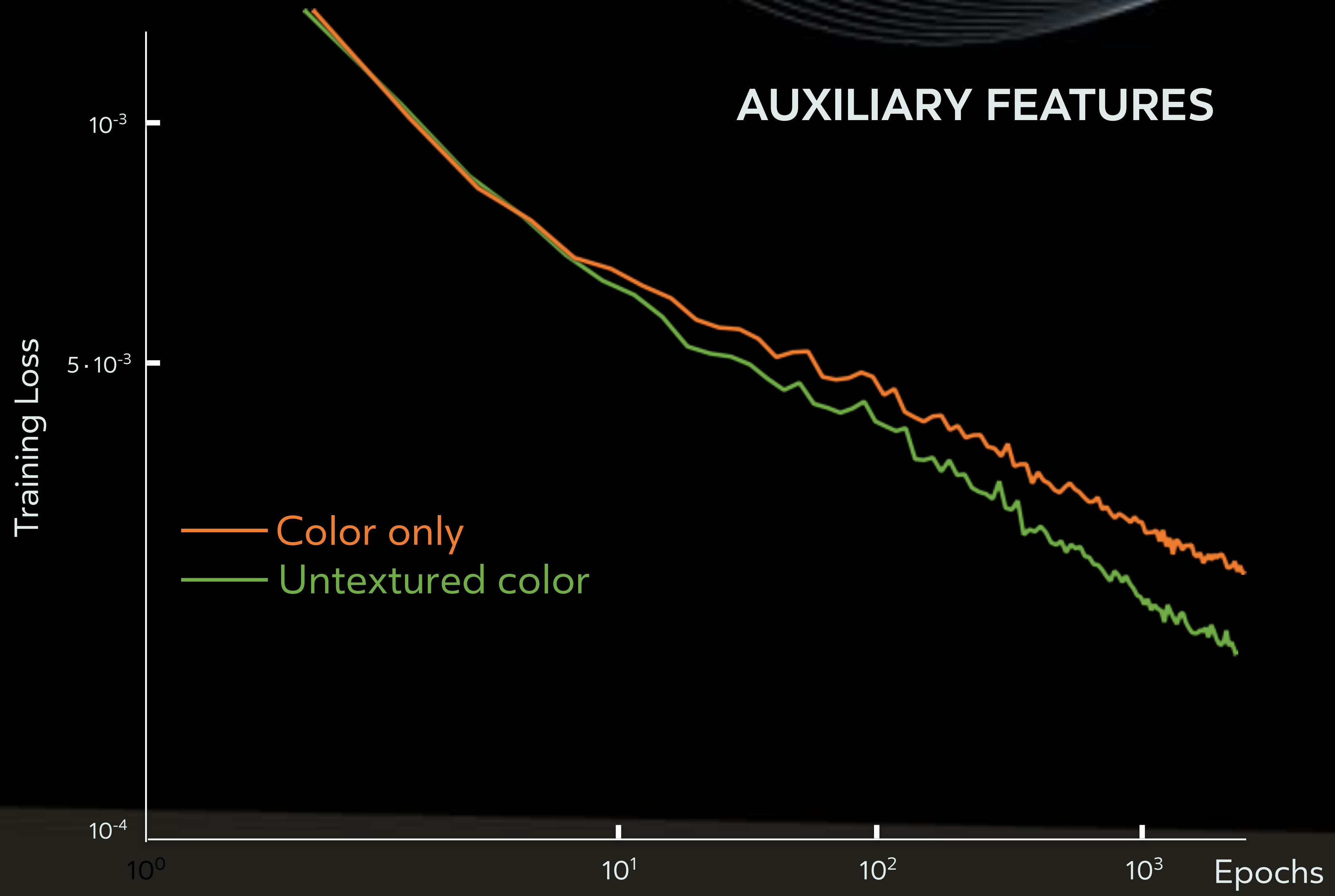
Backup Slides

AUXILIARY FEATURES

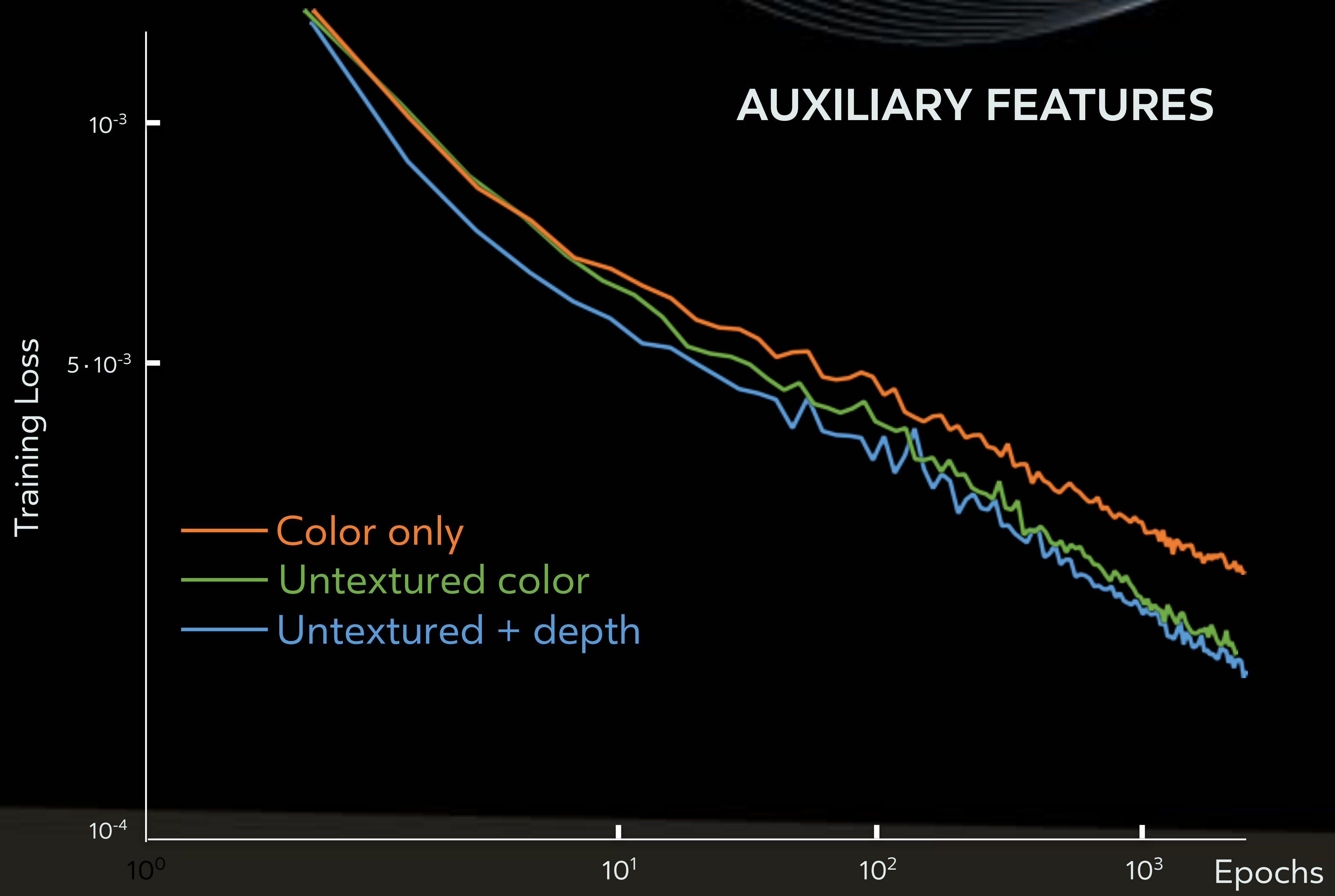
AUXILIARY FEATURES



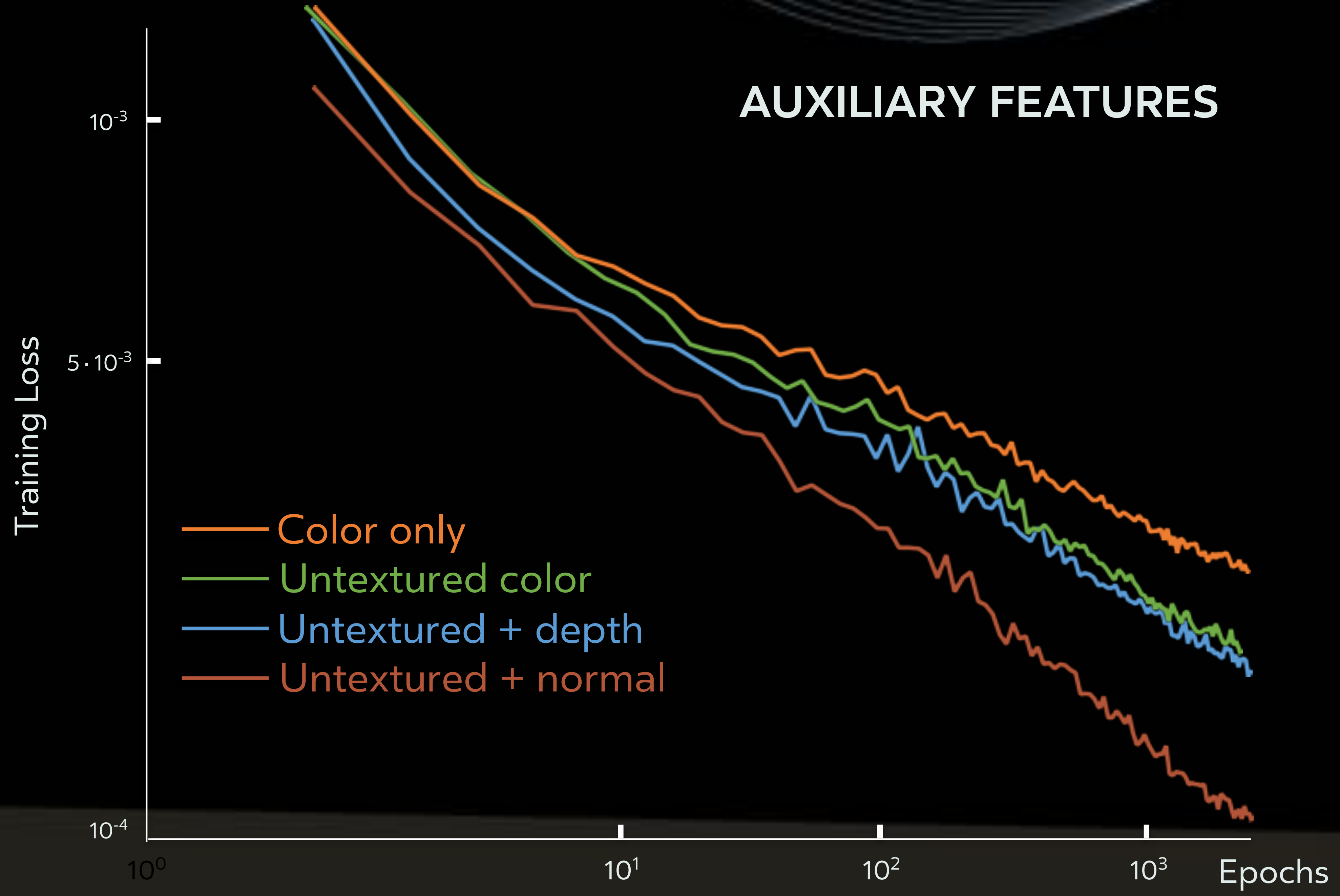
AUXILIARY FEATURES



AUXILIARY FEATURES

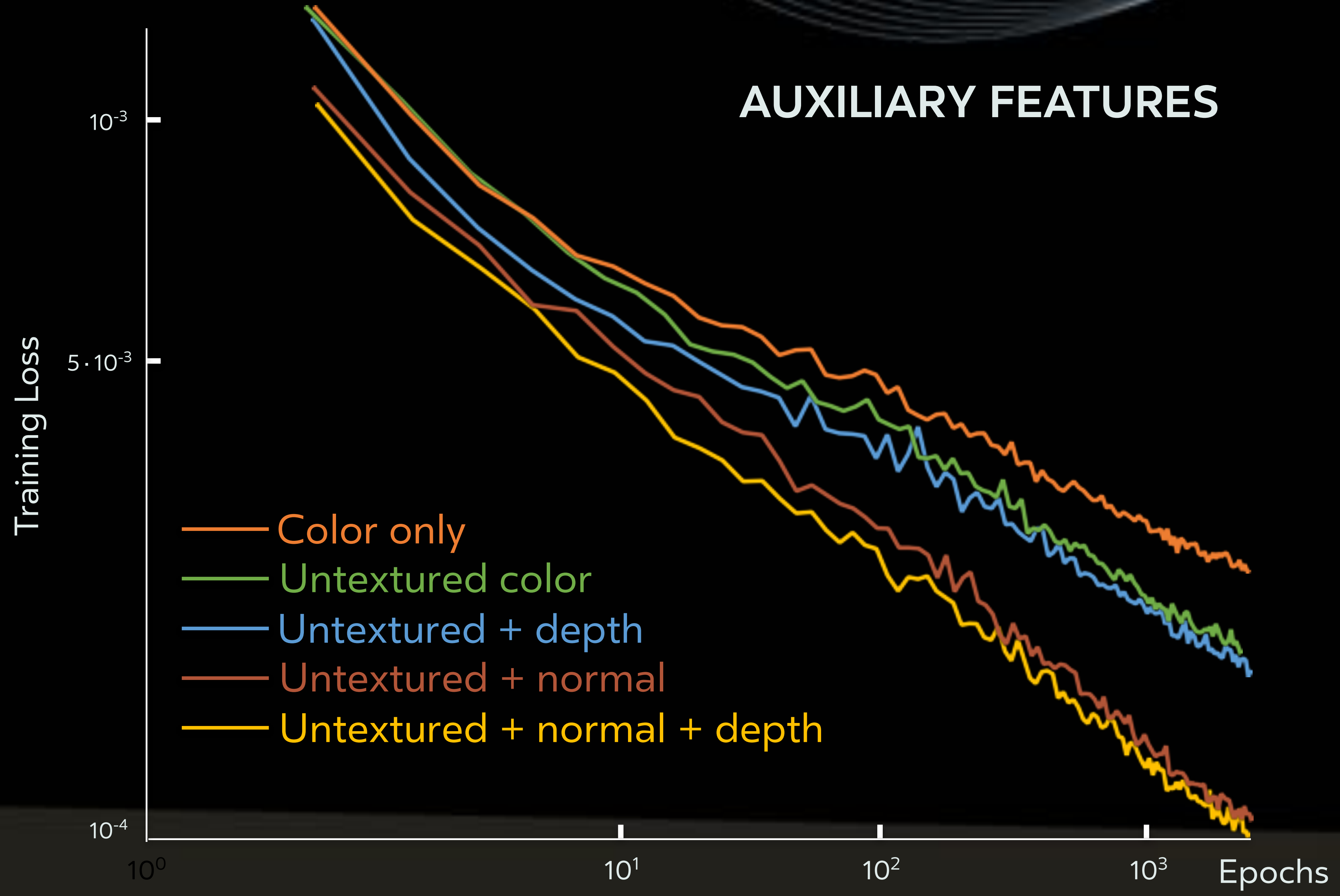


AUXILIARY FEATURES



NVIDIA

AUXILIARY FEATURES



AUXILIARY FEATURES