Bandwidth Limited Imaging via Sparse Scanning and Diffusion Priors

02/25/2026

Presenter: Peiran Li
Supervisor: Adithya Pediredla
Collaborators: Rohit Banerjee, Sotiris Nousias

Introduction

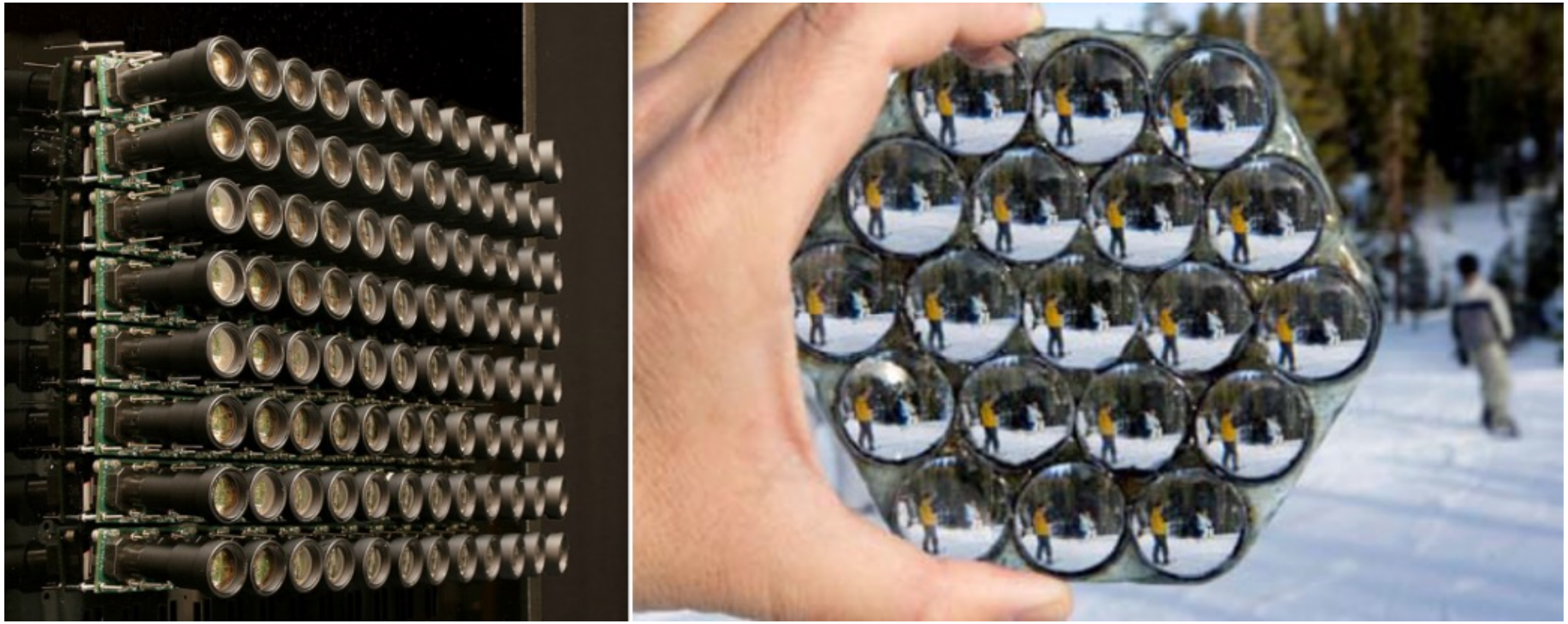
Capturing the Invisible——High-speed Cameras

- **200,000 frames per second**
- **10TB per minute**

$(10\text{TB} \approx \mathbf{2500}\ high\ resolution\ movies)$

Introduction

Increasing the dimension —— Light Field Cameras



- **Conventional image:** L(x, y) —— 2D
- **Light field:** L(x, y, u, v) —— 4D

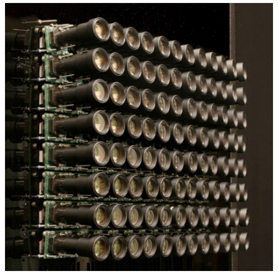
Problem

Why don't we capture everything in the world?

Information Lost

Modern Imaging system

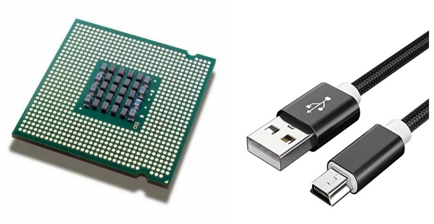
High Speed Camera



Light Field Camera

Raw data

Limited Bandwidth



Readout Device

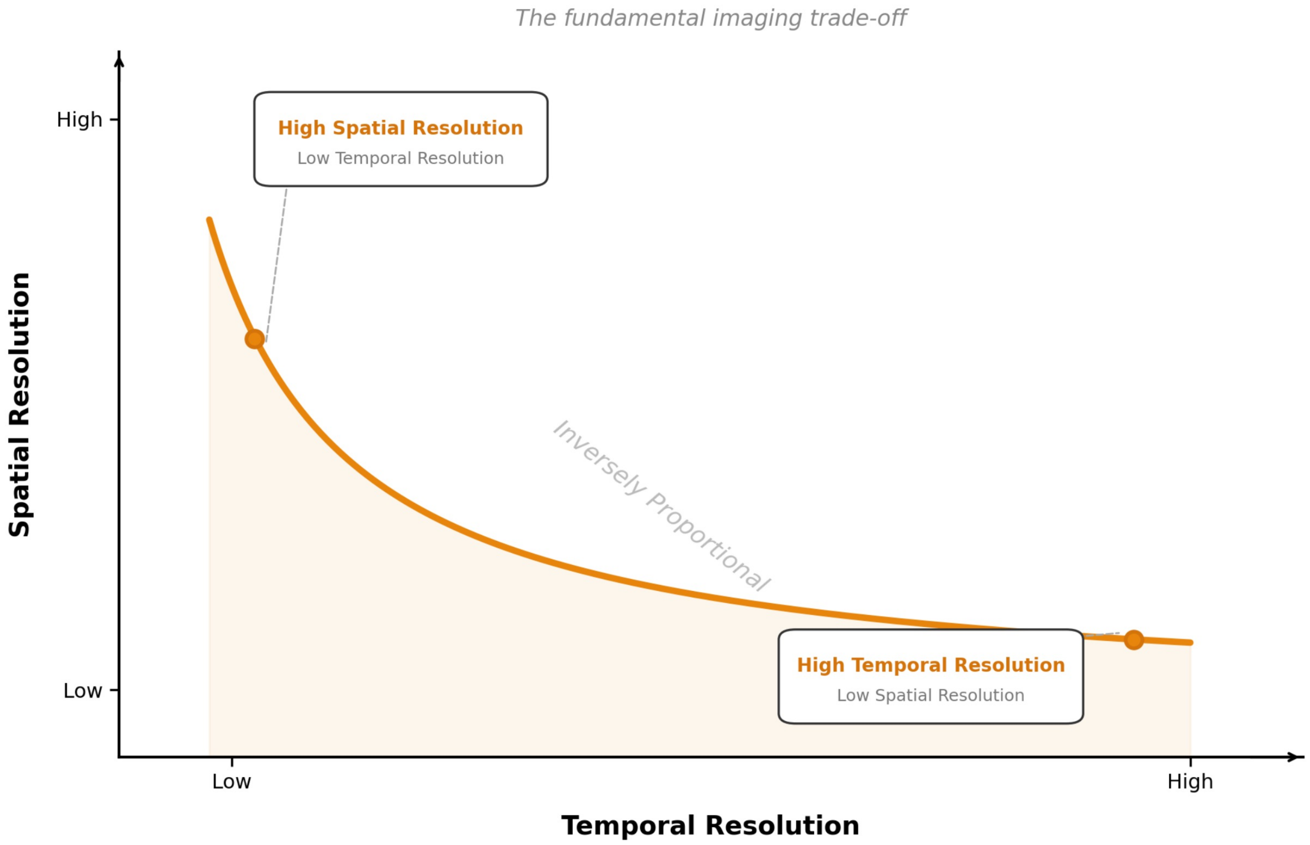
Output data

Bandwidth: How much data your sensor can output per second

Problem Statement

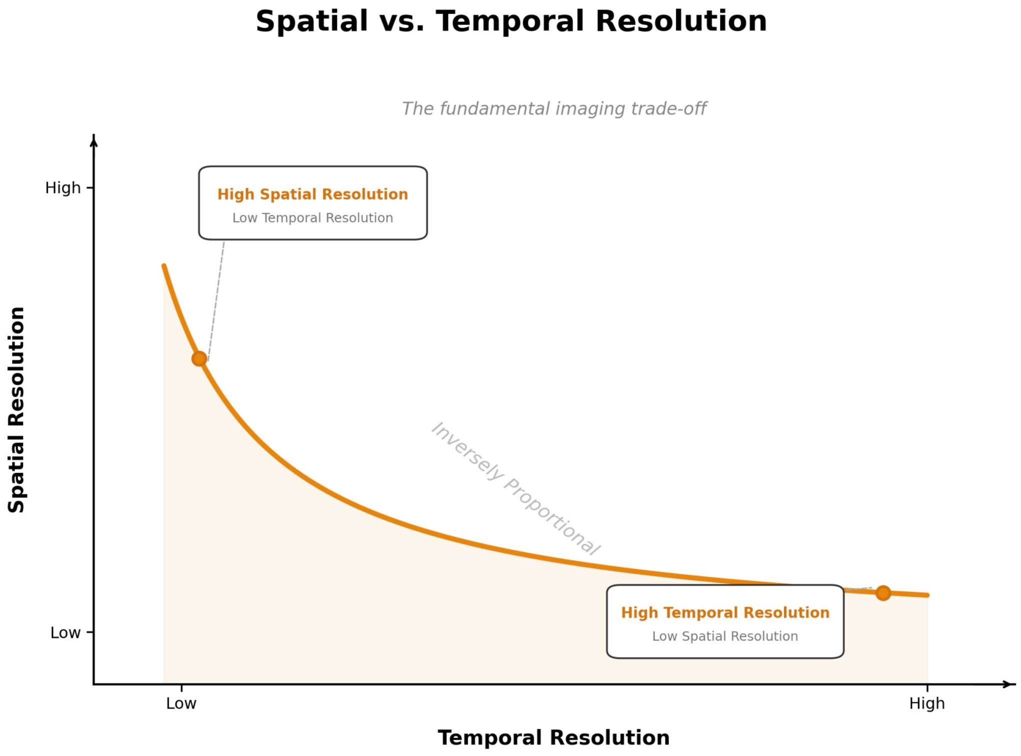
Modern imaging systems are “**bandwidth limited**”

$$Bandwidth \propto \text{Temporal Resolution} \times \text{Spatial Resolution}$$



Problem Statement

Modern imaging systems are “**bandwidth limited**”.



Imaging Systems with High Temporal Resolution

SPAD

Phantom Cameras

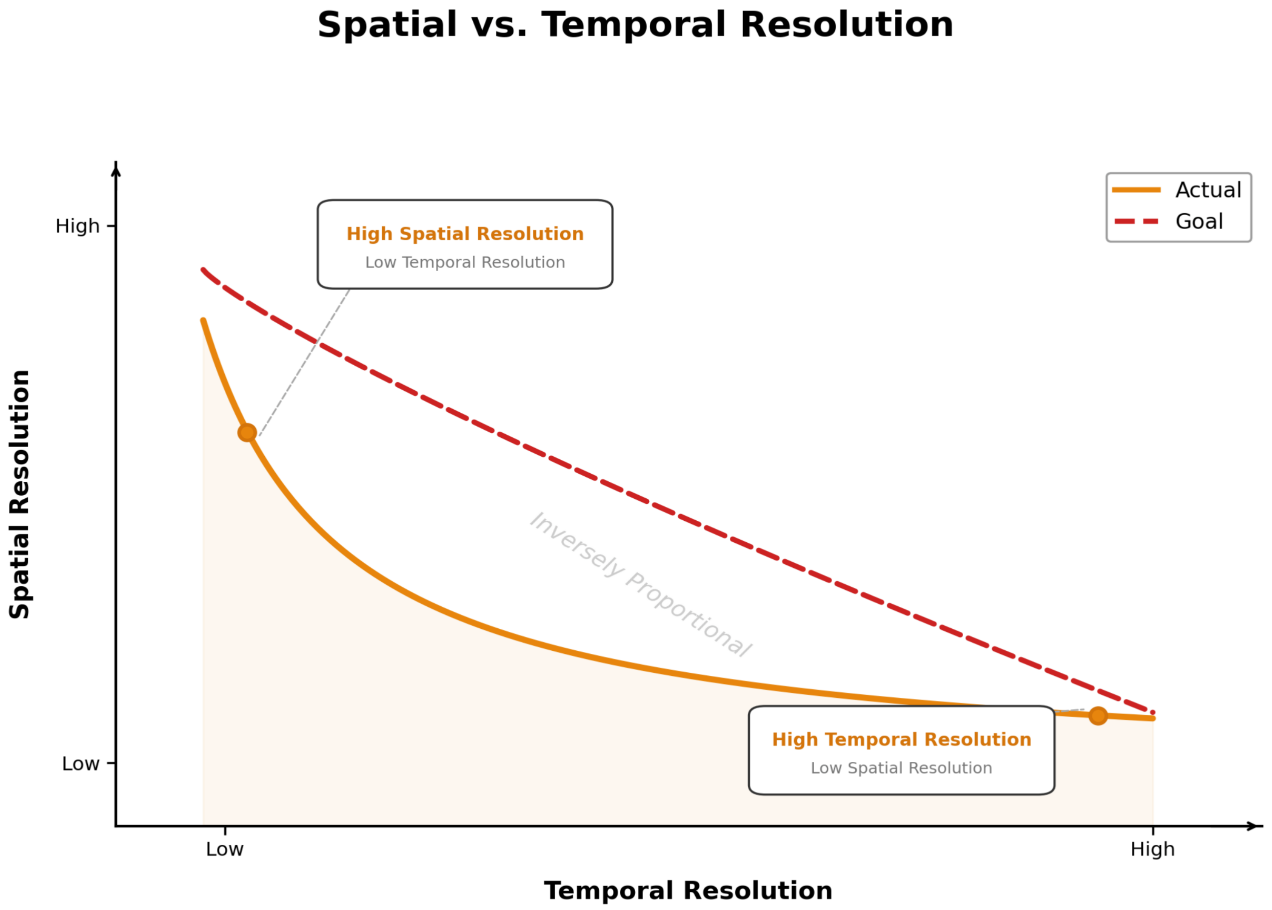
Imaging Systems with High Spatial Resolution

Sky Survey Camera

Microscope

Problem Statement

Modern imaging systems are "**bandwidth limited**".



Phantom Camera:
Temporal Resolution: 10,000 FPS
Spatial Resolution: 64× 64

Our Goal:

Break the curve!

How to Solve the Problem?

Hardware Perspective

- **Faster readout → more circuits**

1,000 fps high-speed camera **$15,000**
10,000 fps high-speed camera **$50,000**

- **Hard physical limit**

❌

Algorithm Perspective

Recover high-dimensional content from incomplete or compressed measurements

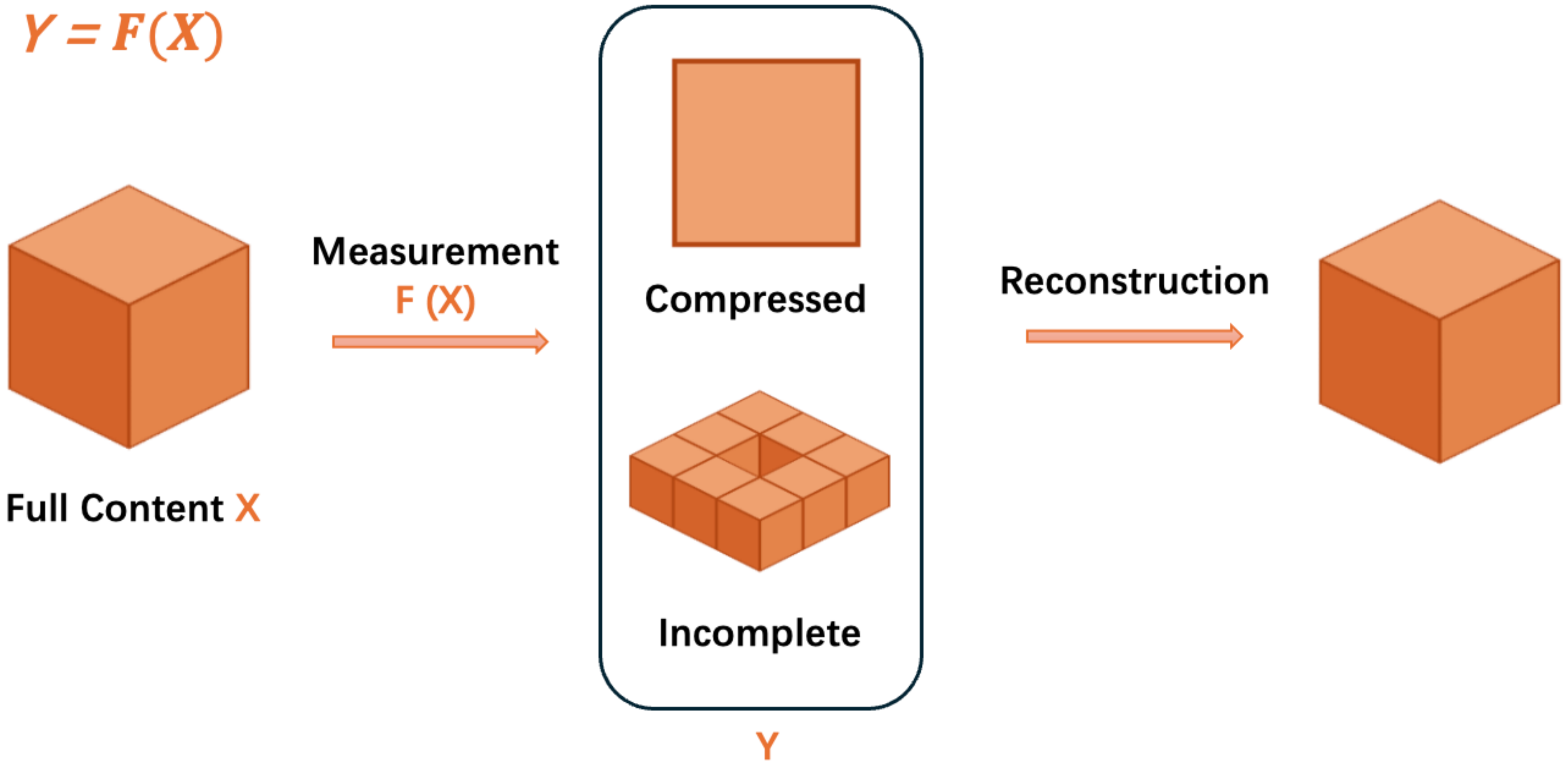
$\boldsymbol{Y = F(X)}$

- X: full content
- Y: incomplete or compressed content
- F: any measurement / compression process

Reconstruct full X from Y

✅

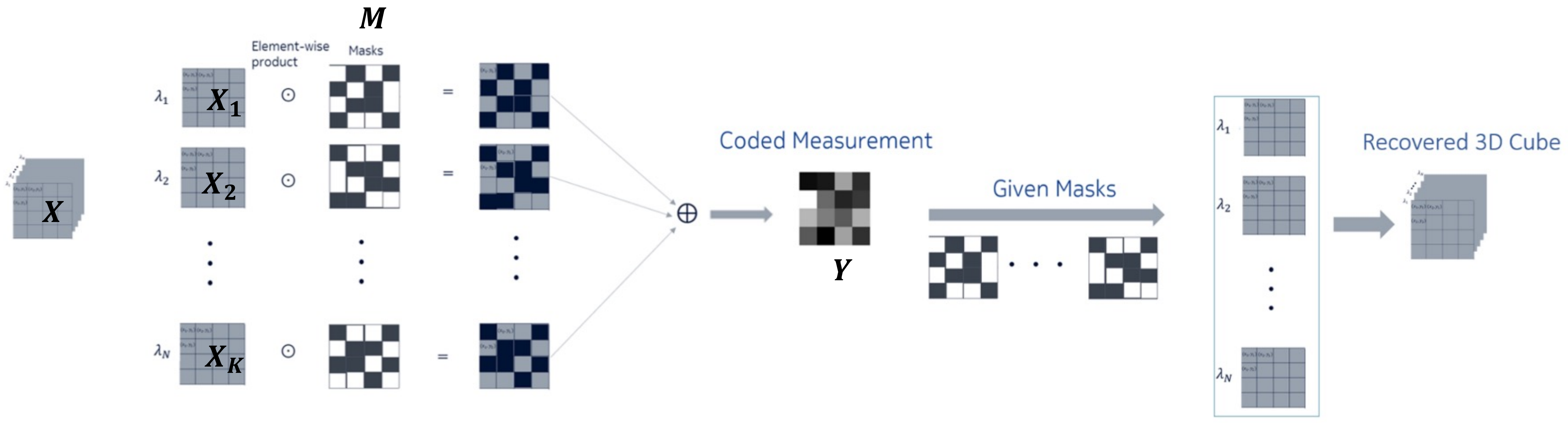
How do we solve the problem?



Reconstruct full X from the incomplete Y

Related work

Compressive Imaging (Yuan et al., 2021)

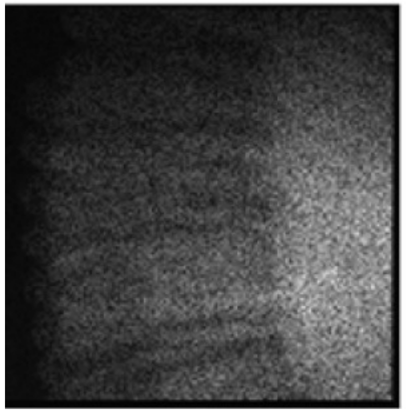


$$Y = \sum_{k=1}^{K} (X_k \odot M_k)$$

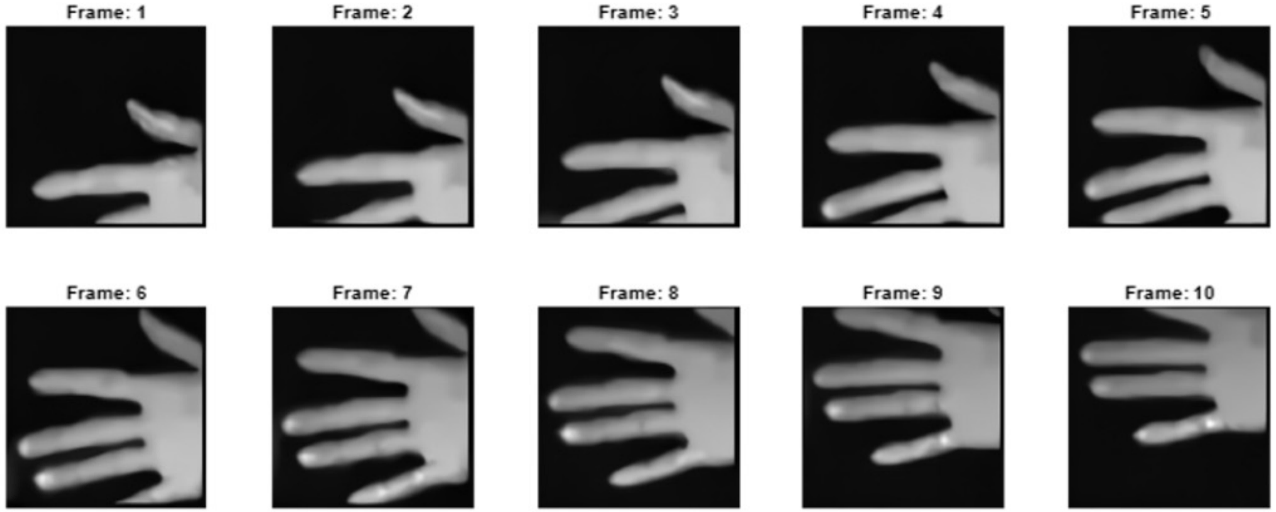
- $X = \{X_1, X_2 ...... X_K\}$: high−dimensional 3D cube
- M_k: coding mask applied to slice X_K
- Y: 2D coded measurement
- Reconstruct 3D cube X from 2D Y

Related work

Compressive Imaging (Yuan et al., 2021)



2D measurement



3D reconstruction video frame

Related work

Traditional Inpainting —— TV (Total Variation) Inpainting

Ground Truth X

Incomplete Measurement Y

Reconstructed X

$$Y = M \odot X$$

- Reconstruction: $\hat{X} = \arg\min_{X} \|M \odot X - Y\|_2^2 + \lambda \mathrm{TV}(X)$
- TV prior: works for smooth areas and edges

Current Algorithm Limitation

Method	Measurement Operator F(X)	Reconstruction Method
Compressive Imaging	$Y = \sum_{k=1}^{K} (X_k \odot M_k)$	Mask decoding
Traditional Inpainting	$Y = M \odot X$	Traditional priors

- Compressive imaging: Still requires full data acquisition — the sensor bandwidth bottleneck remains unresolved
- Traditional Inpainting (TV, PDE, patch-based…) **:** These priors fail to recover complex textures and high-frequency details in dynamic scene

Our Method

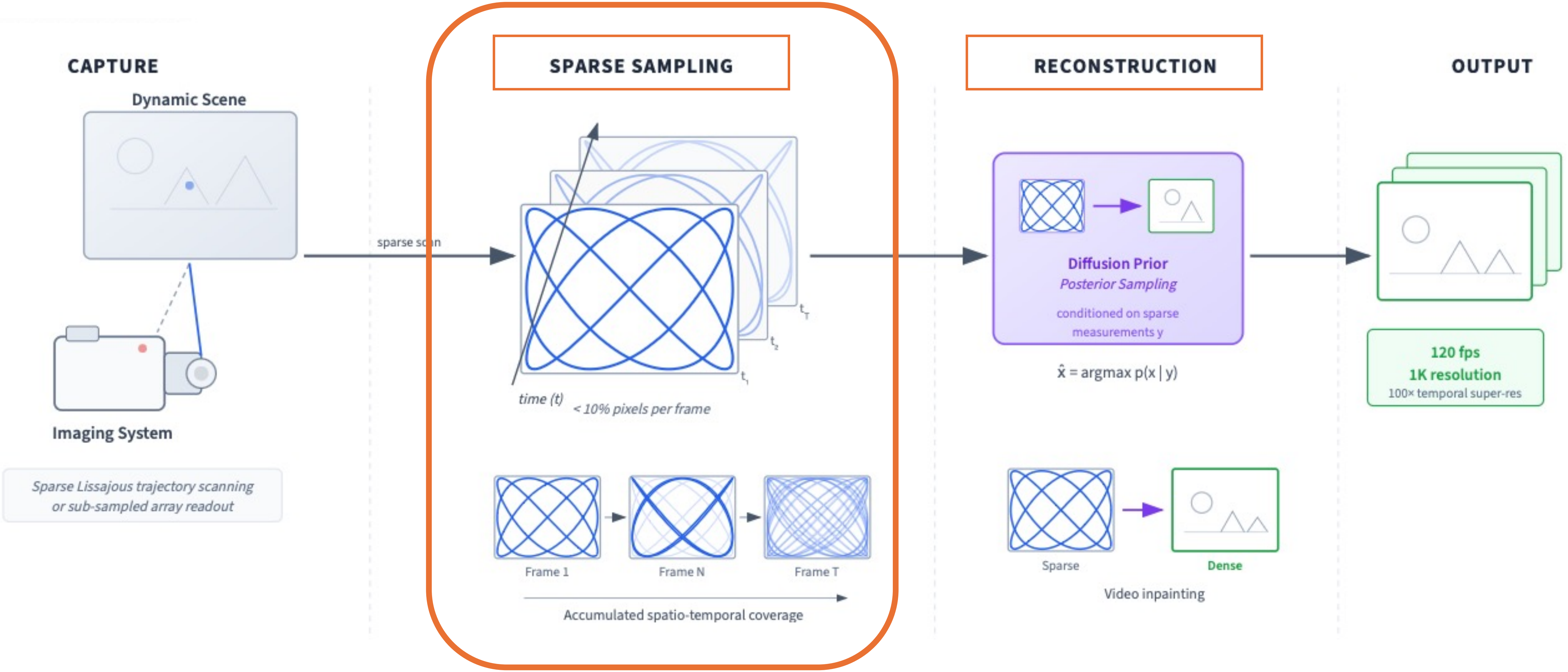
Sparse Scanning and Diffusion Priors

Method	Measurement Operator F(X)	Reconstruction Method
Compressive Imaging	$Y = \sum_{k=1}^{K} (X_k \odot M_k)$	Mask decoding
Traditional Inpainting	$Y = M \odot X$	Traditional priors
Ours	Sparse Scanning	Generative prior

Our Method

Sparse Scanning and Diffusion Priors

Our Method


CAPTURE
Dynamic Scene
Imaging System
Sparse Lissajous trajectory scanning
or sub-sampled array readout
sparse scan
SPARSE SAMPLING
time (t)
< 10% pixels per frame
Frame 1
Frame N
Frame T
Accumulated spatio-temporal coverage
RECONSTRUCTION
Diffusion Prior
Posterior Sampling
conditioned on sparse
measurements y
x̂ = argmax p(x | y)
Sparse
Dense
Video inpainting
OUTPUT
120 fps
1K resolution
100× temporal super-res

Sparse Sampling

Ideal: A sensor that only reads specified pixels

Such a camera does not exist today

- Research prototype
- Custom chip tape-out: $500K–$1M

Sparse Sampling

Simulation:

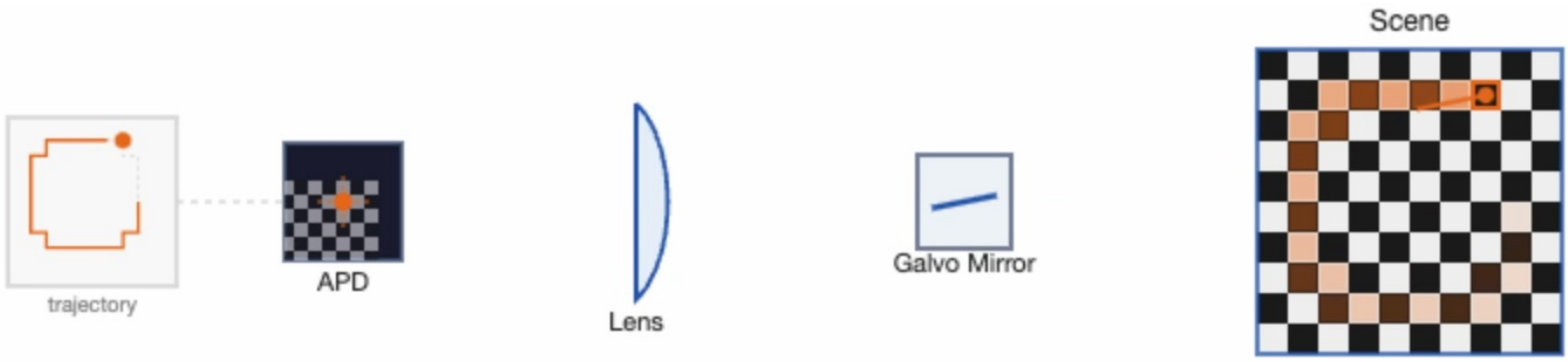
- Randomly drop pixels
- No real hardware noise/imperfection

Physical emulation:

Pixel-Level Imaging System

Sparse Sampling

Pixel-Level Imaging System



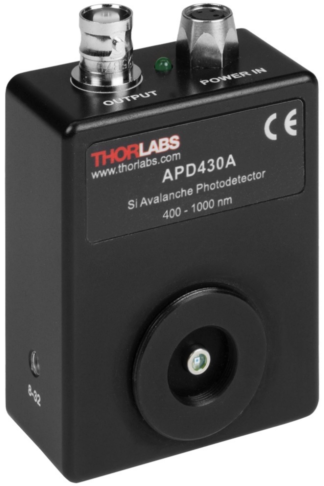
Scene — Checkerboard illuminated by bright light
Galvo Mirrors — Steers the light path by rotating at a programmed frequency
Lens — Focuses the reflected light onto the APD
APD — Single-pixel detector, measures light intensity at one fixed point at a time

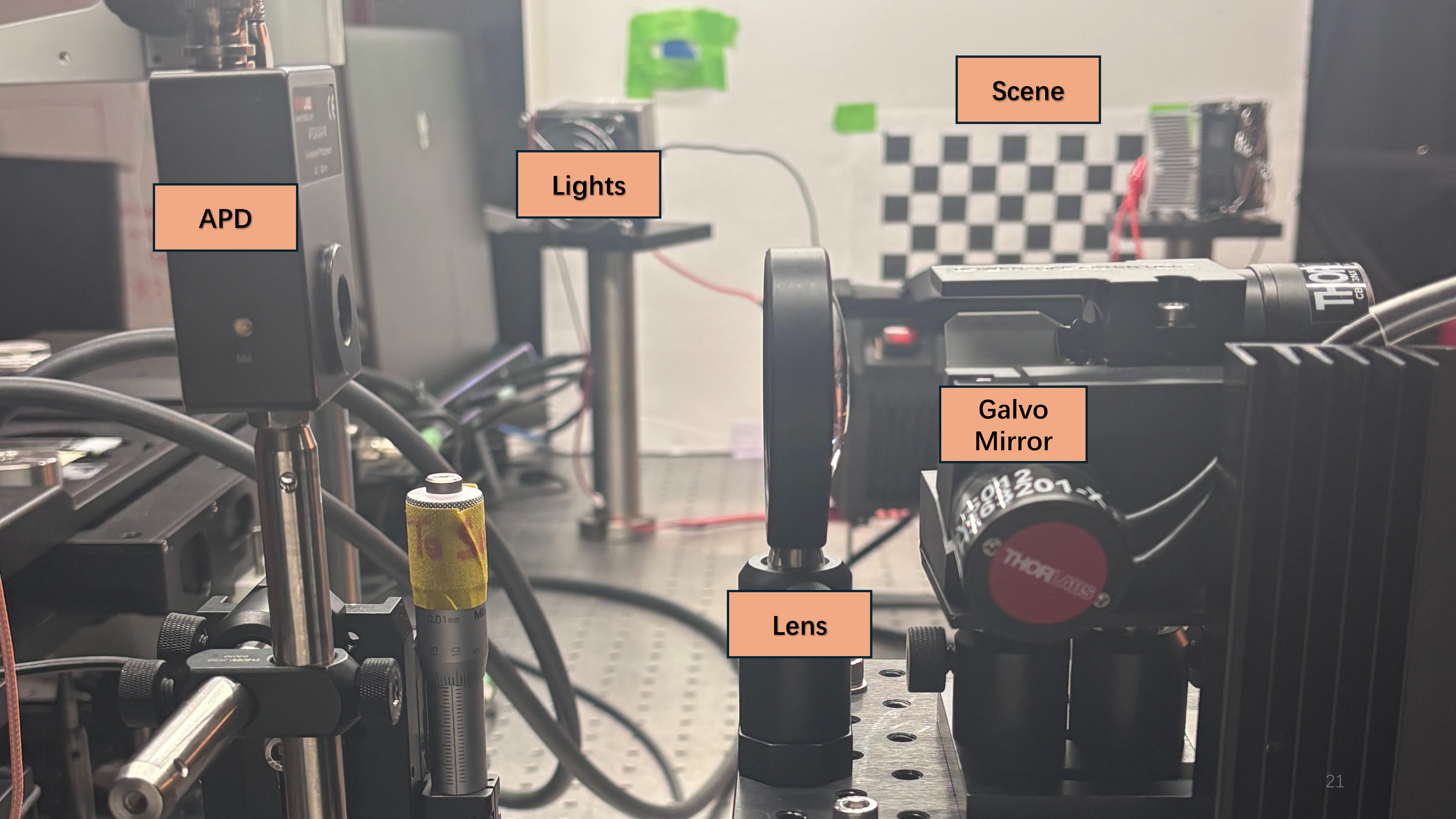
Sparse Sampling

Pixel-Level Imaging System

The imaging systems consists of:

- Avalanche Photo Diodes (APD)
- Dual-axis Scanning Galvanometer
- Singlet Convex Lens
- Very bright Lights




Scene
Lights
APD
Galvo
Mirror
Lens

Captured Scene

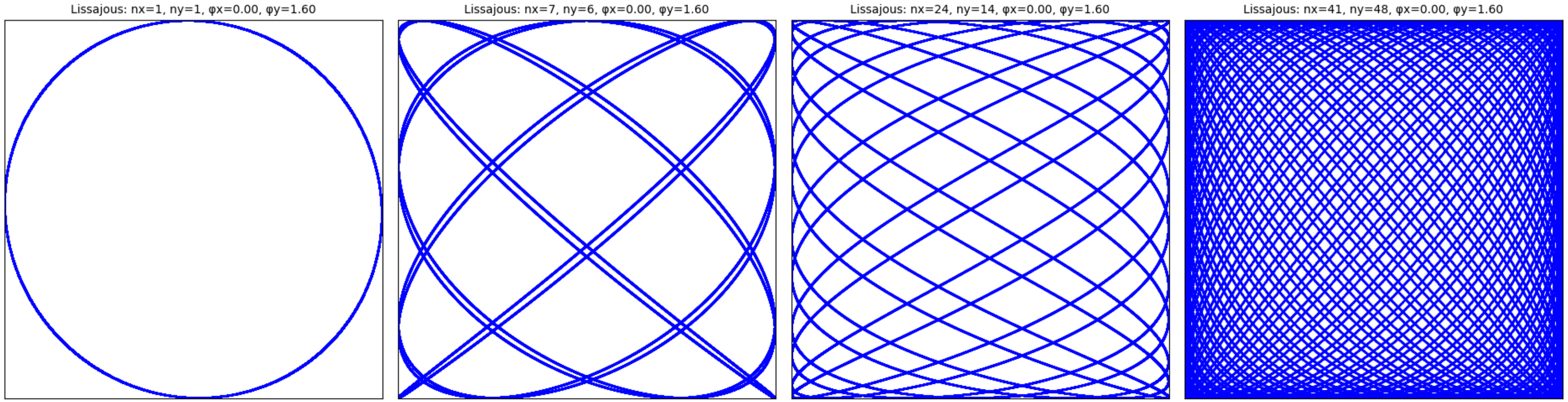
Sparse Sampling

Scan Pattern

We use **Lissajous patterns** as our scanning trajectory, where f_{max} = 500 Hz

$$x(t) = A\sin(2\pi n_x f_0 t + \delta_1)$$
$$y(t) = B\cos(2\pi n_y f_0 t + \delta_2)$$

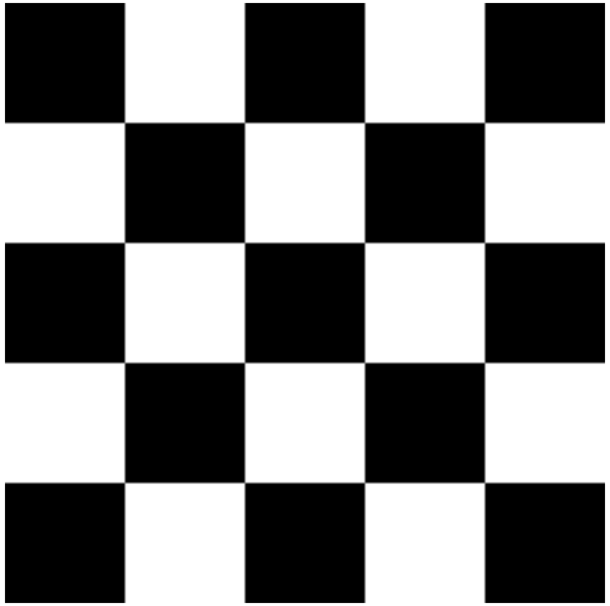
Galvo Limit: $\boldsymbol{n_x \times f_0 \leq f_{max}}$ **and** $\boldsymbol{n_y \times f_0 \leq f_{max}}$



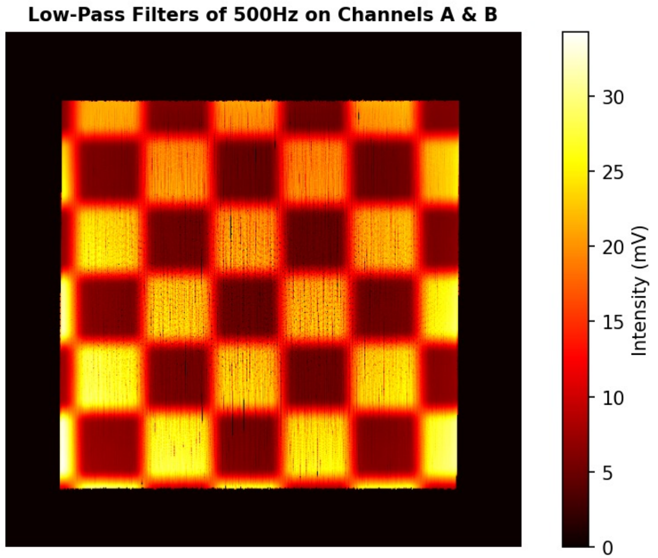
Larger $\boldsymbol{n_x}$ and $\boldsymbol{n_y}$ - denser scan pattern

Sparse Sampling

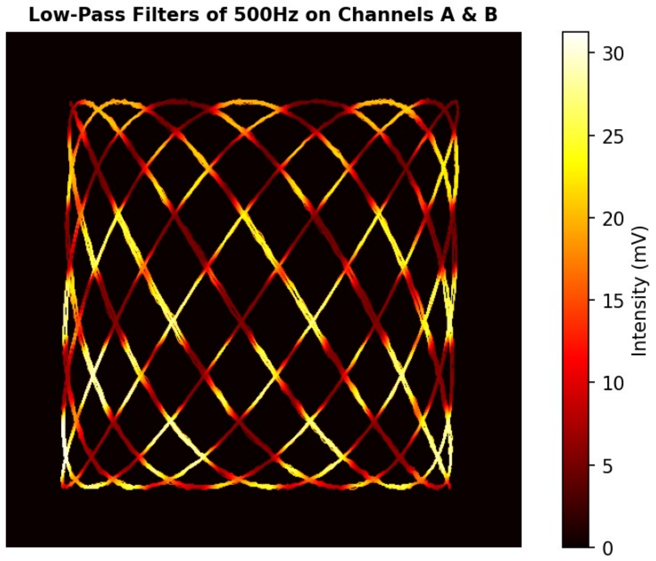
Static Scene Scan Result



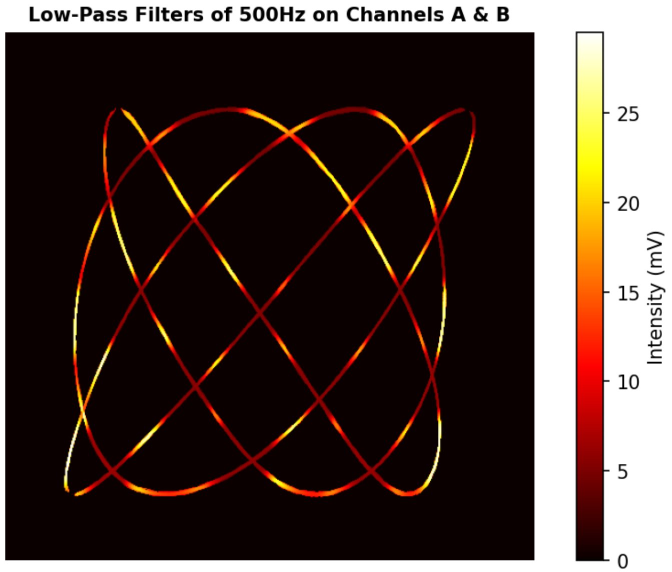
Checkerboard Scene



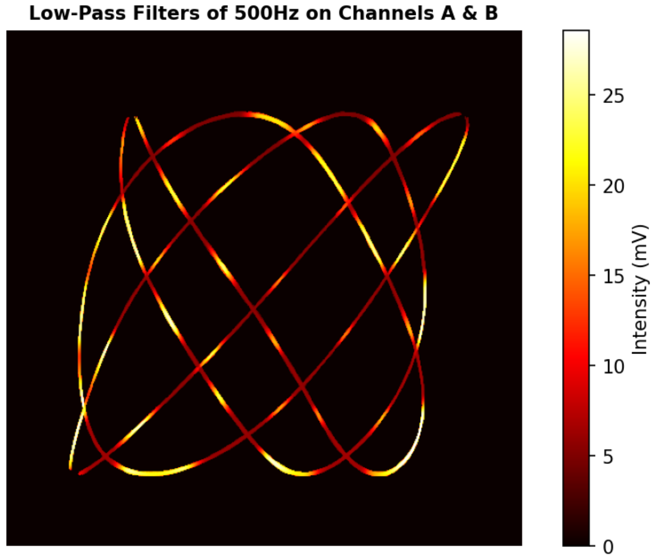
100-0.1Hz



110-170Hz



300-400Hz

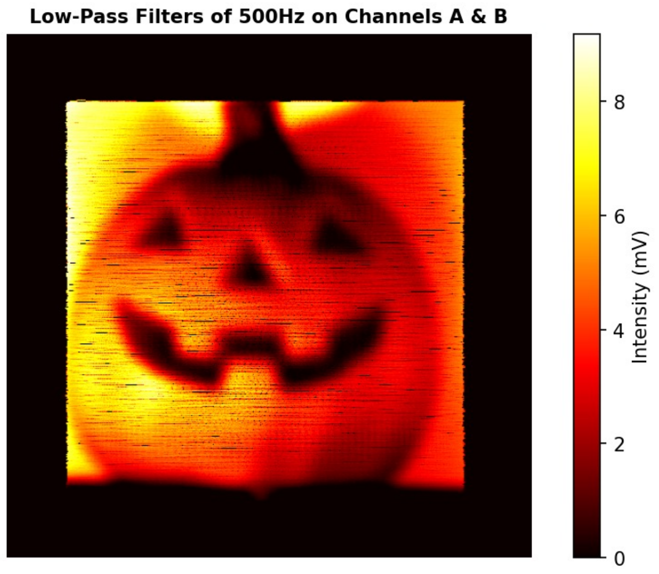


360-480Hz

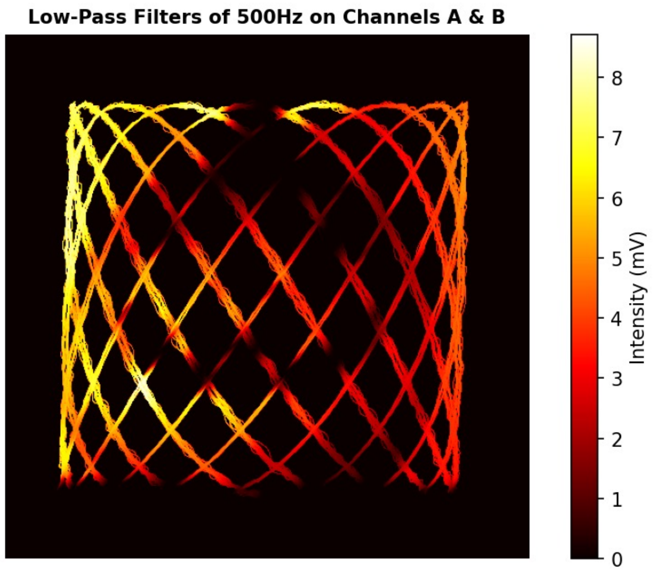
Sparse Sampling

Static Scene Scan Result

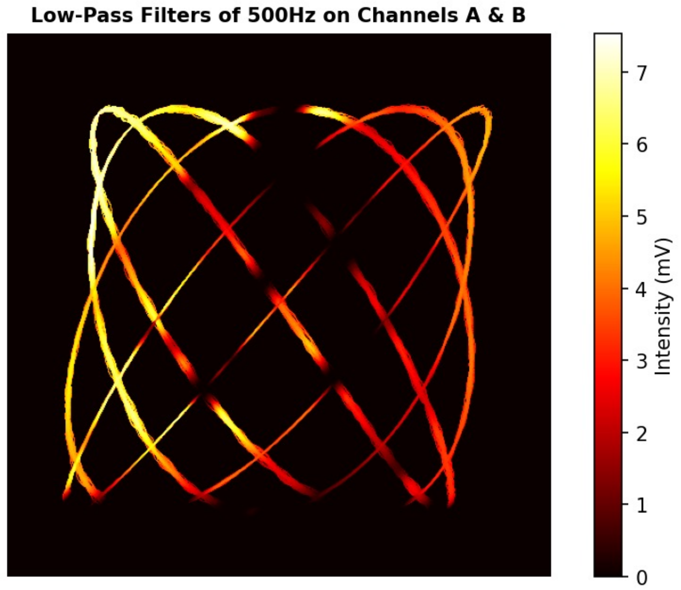
Pumpkin Scene



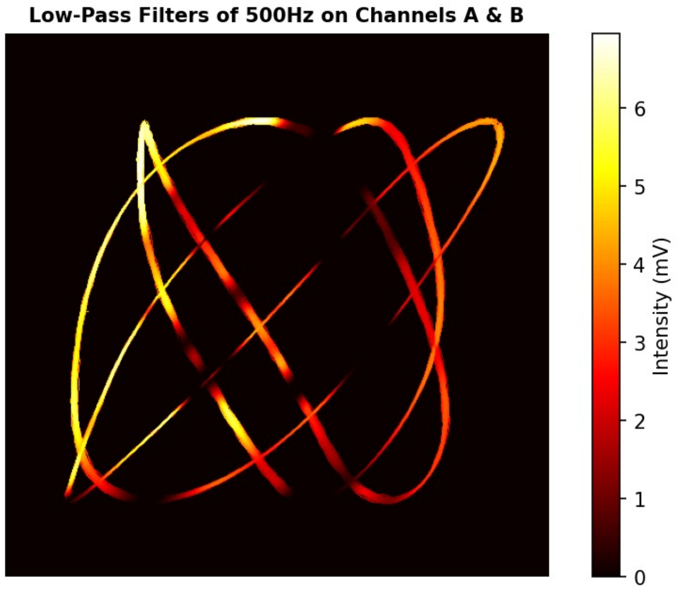
100-0.1Hz



110-170Hz



300-400Hz

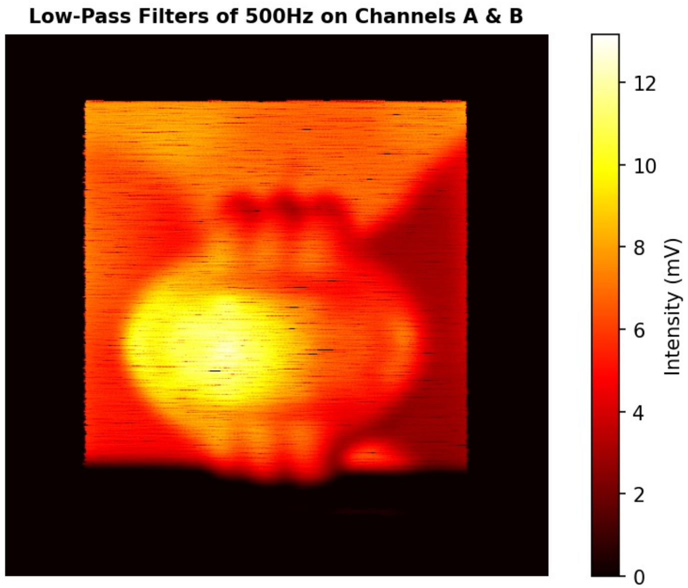


360-480Hz

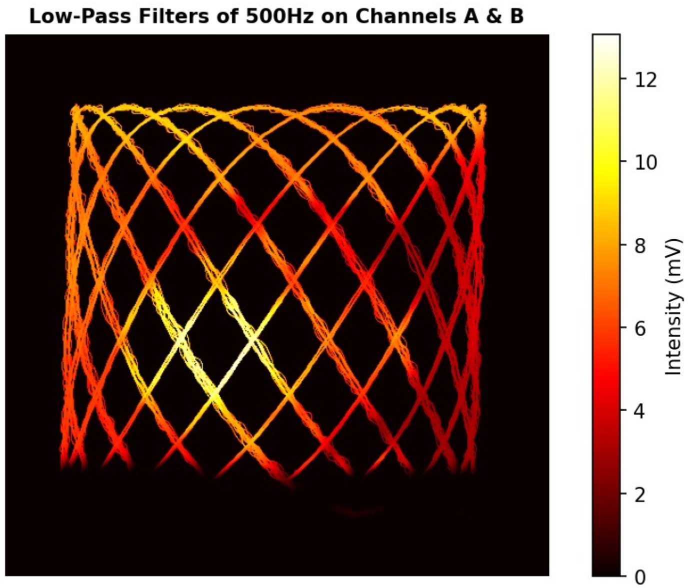
Sparse Sampling

Static Scene Scan Result

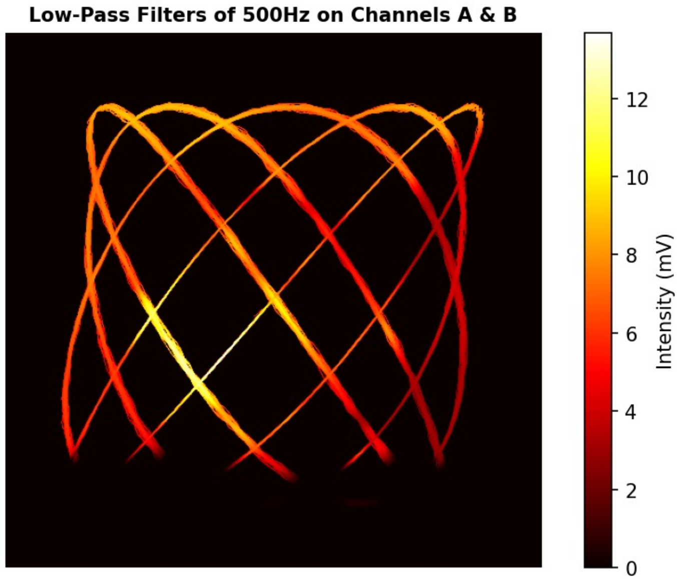
Candle Scene



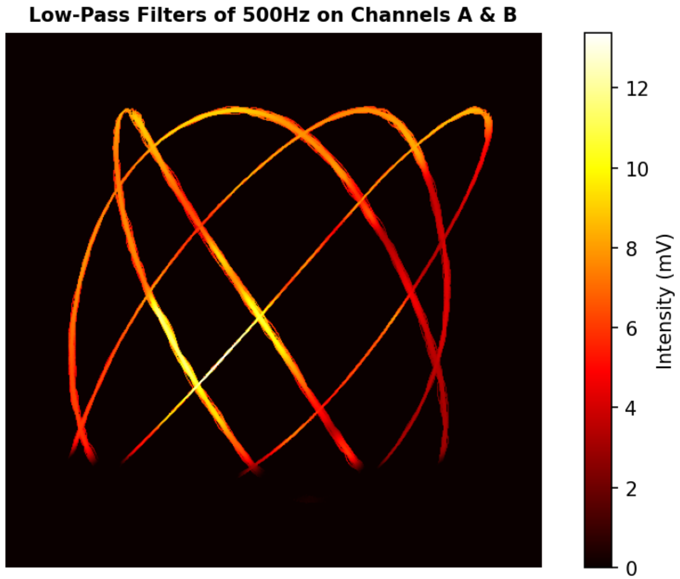
100-0.1Hz



110-170Hz

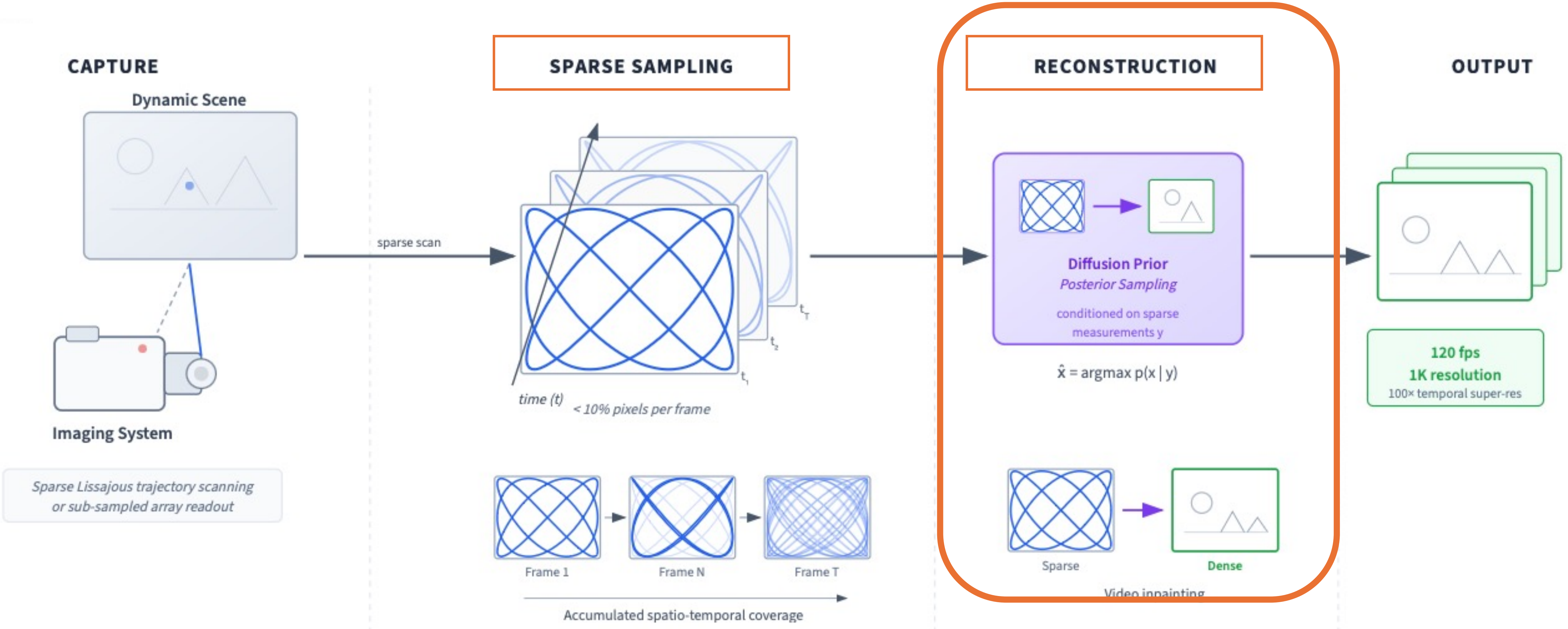


300-400Hz



360-480Hz

Our Method

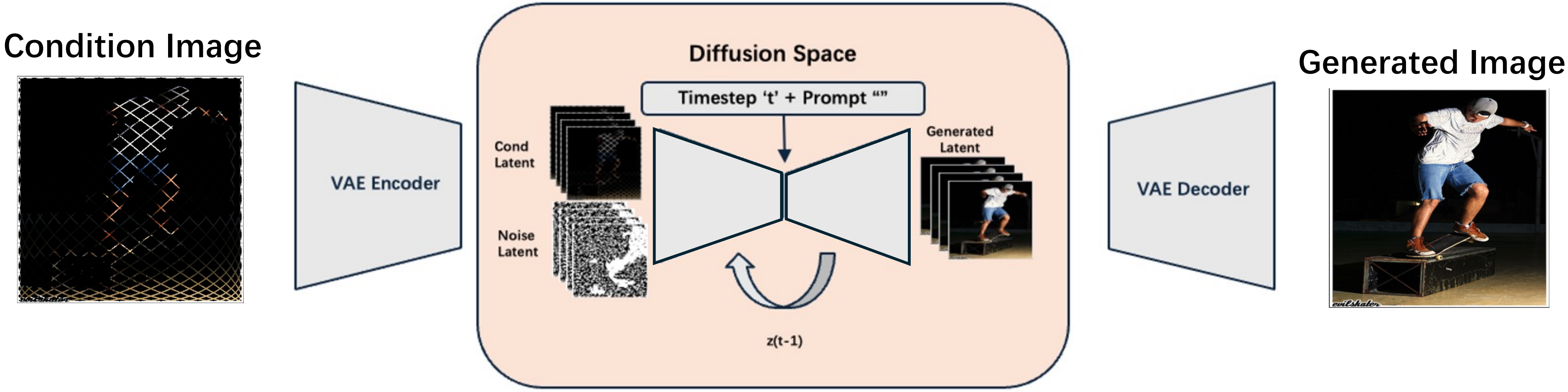


CAPTURE
Dynamic Scene
Imaging System
Sparse Lissajous trajectory scanning
or sub-sampled array readout
sparse scan
SPARSE SAMPLING
time (t)
< 10% pixels per frame
Frame 1
Frame N
Frame T
Accumulated spatio-temporal coverage
RECONSTRUCTION
Diffusion Prior
Posterior Sampling
conditioned on sparse
measurements y
x̂ = argmax p(x | y)
Sparse
Dense
Video inpainting
OUTPUT
120 fps
1K resolution
100× temporal super-res

Diffusion Priors

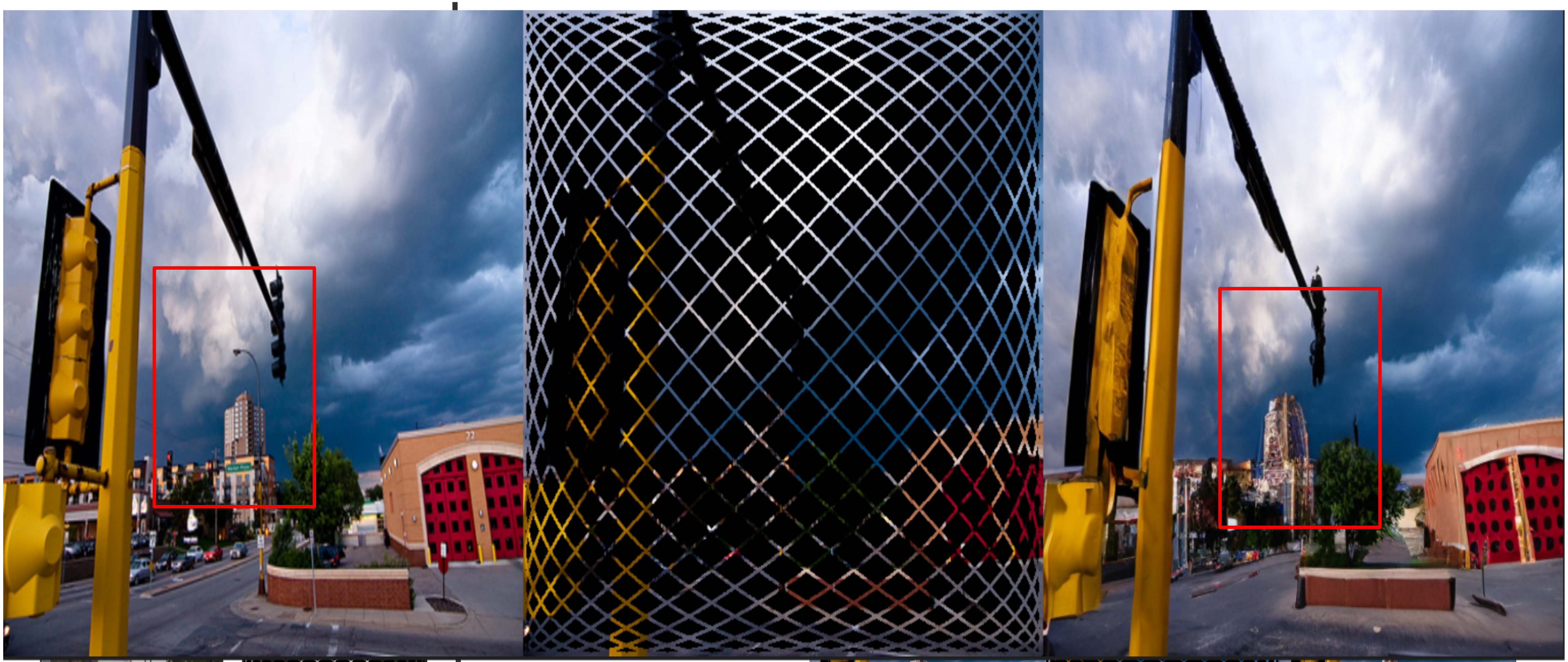
Image to image Reconstruction
Video Reconstruction

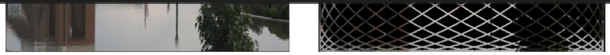
Image-to-Image Diffusion Stable Diffusion 2



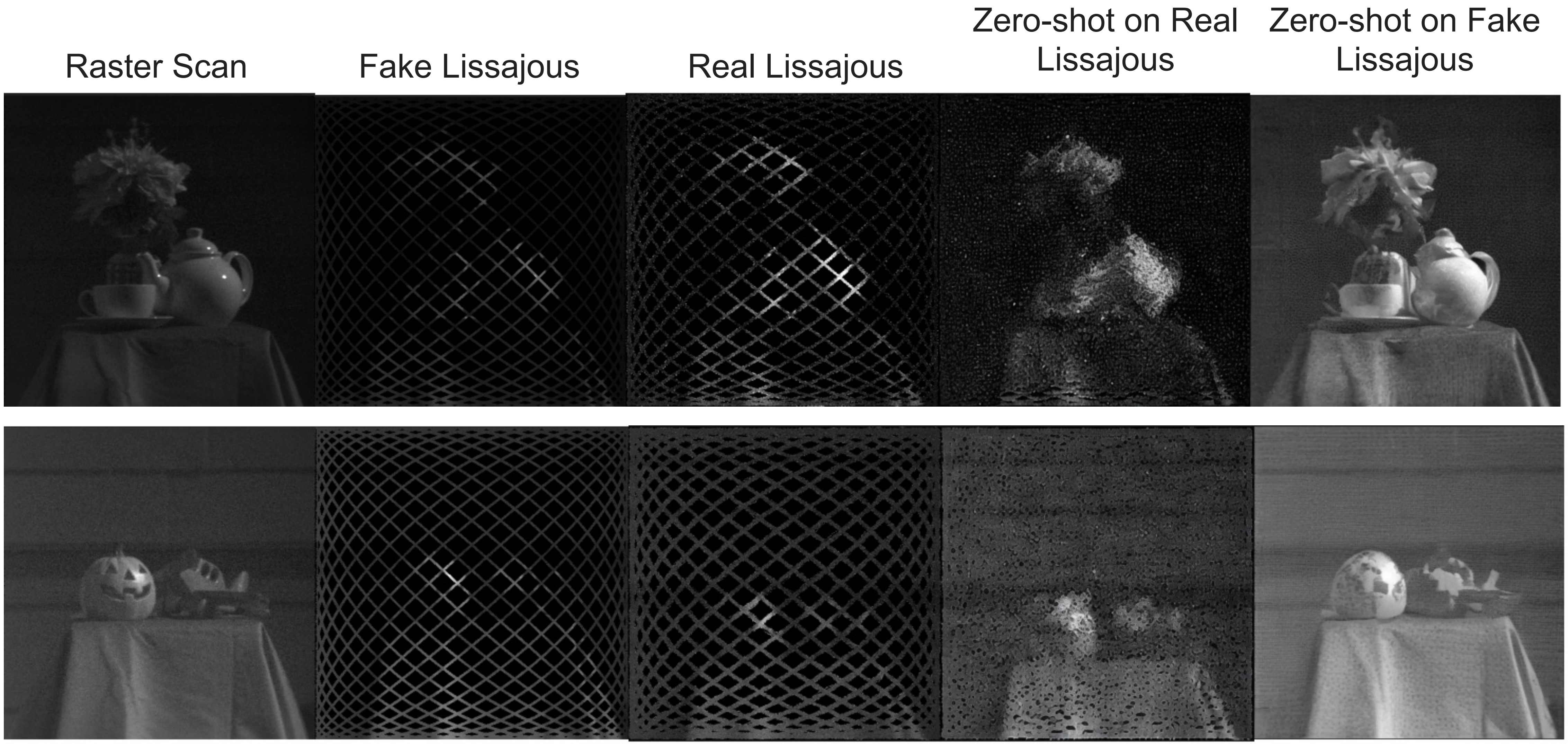
- Condition Latent: Carries original video information
- Noise Latent: Pure Gaussian noise in latent space with same size
- All UNet parameters (Diffusion Space) are trained on the simulation data

Simulation Results – COCO Dataset



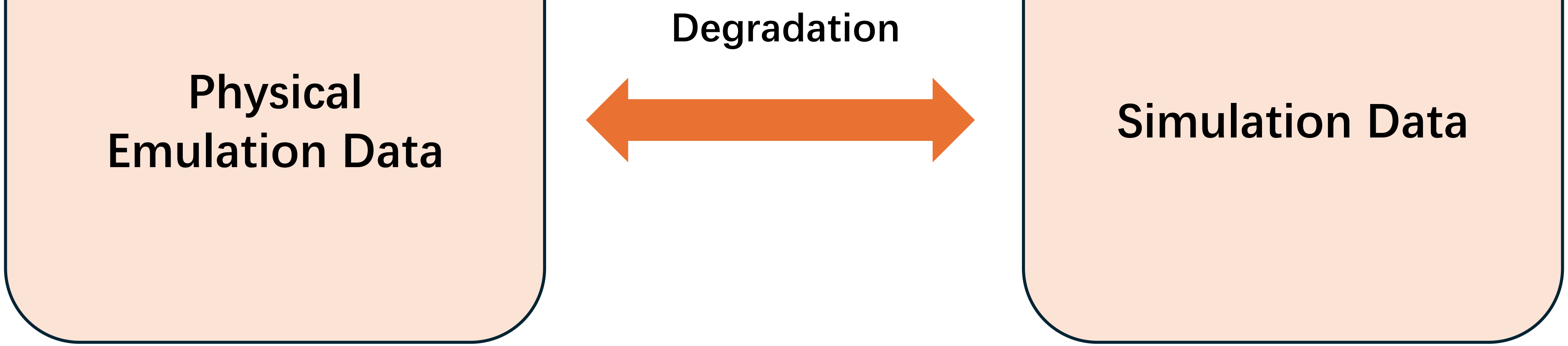


Physical Emulation Results



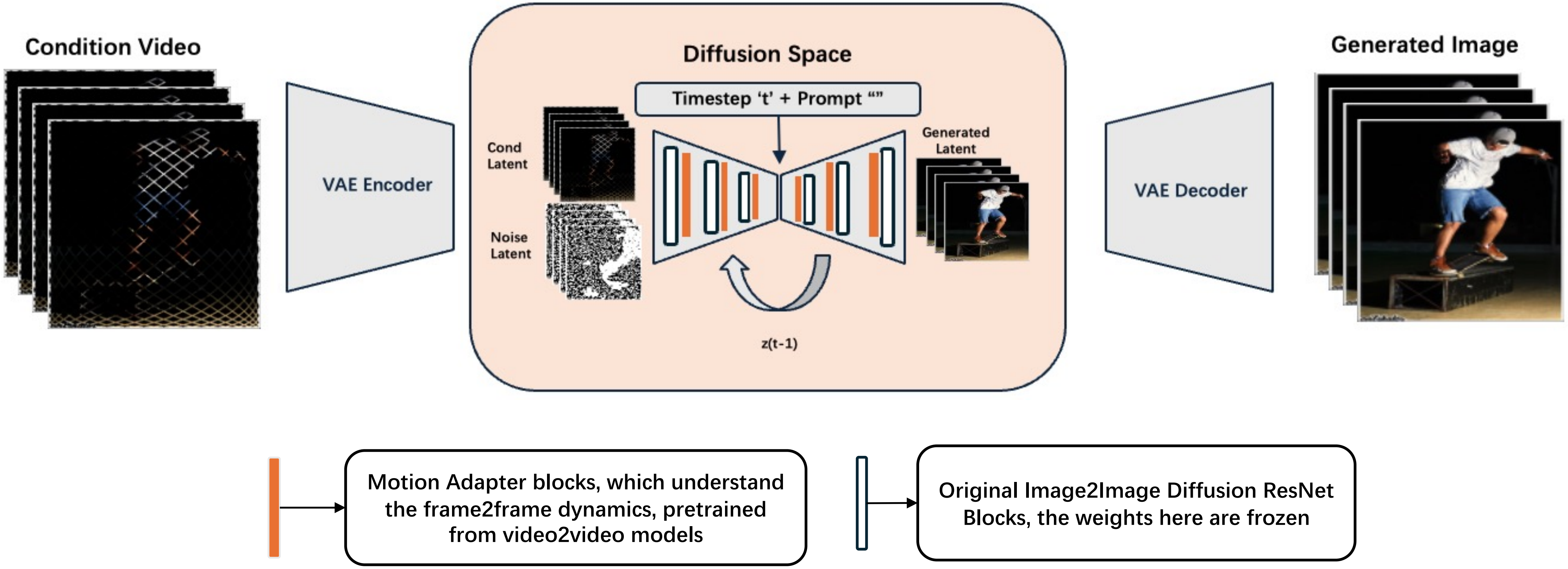
Summary: Image-to-Image Model

Bridge between real-world and generative



Future Work: Learning a degradation model on the conditioning end to bridge the sim-to-real gap.

Video Diffusion Pipeline



Simulation Results

Zero-shot Results are from videos of uCO3D Dataset, which contain scenes of everyday items. Validated on a small set of 2 videos (each video contains ~1000 frames).

	SSIM	PSNR	Abs Err
Video	0.701	19.594	0.076

Problem

We use **Lissajous patterns** as our scanning trajectory, where f_{max} = 500 Hz

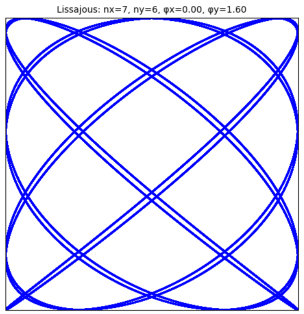
$$x(t) = A\sin(2\pi n_x f_0 t + \delta_1)$$
$$y(t) = B\cos(2\pi n_y f_0 t + \delta_2)$$

Galvo Limit: $n_x \times f_0 \leq f_{max}$ and $n_y \times f_0 \leq f_{max}$

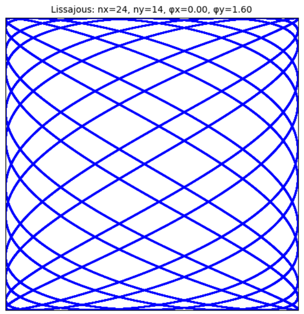
Larger n_x and n_y - Denser

Smaller n_x and n_y - Sparser

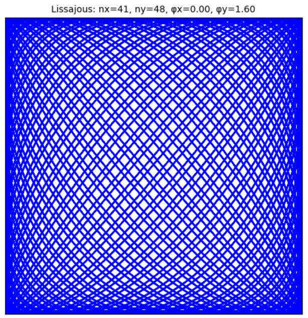
f_0 — which controls the video frame rate



n_x: n_y 7: 6



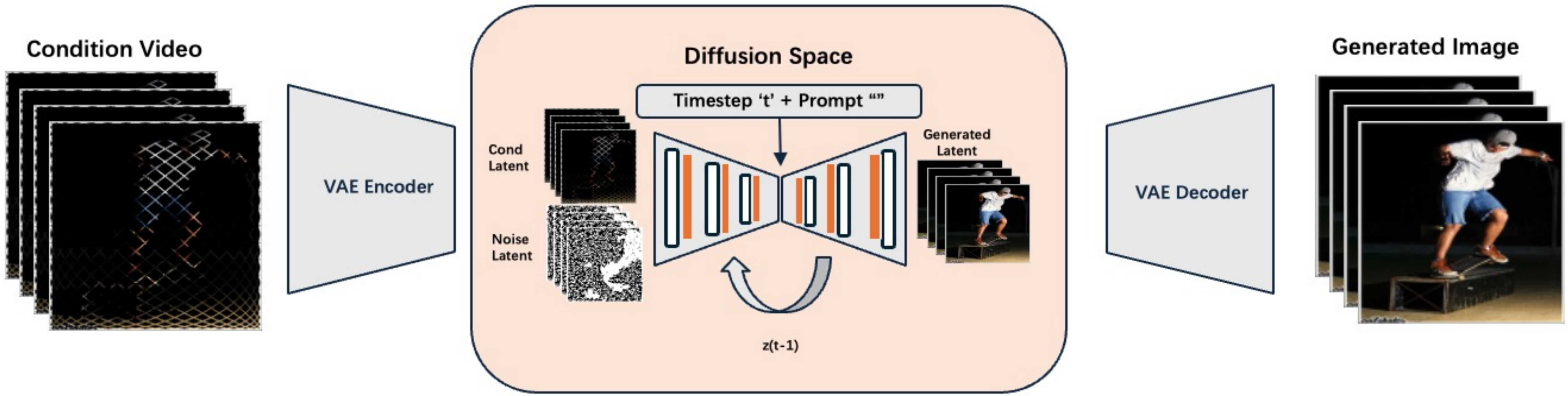
n_x: n_y 24: 14



n_x: n_y 41: 48

$f_0 \uparrow \rightarrow$ smaller n_x, n_y $\rightarrow$ sparser scan on each frame

Video Diffusion Pipeline – Future Work



In-place of trying to generate all possible frames – we only generate keyframes

Keyframes → Interpolation Model

Future Work

Sparse Sampling:

- Capture the video using our single pixel imaging system
- Capture more static scene to build our own sparse scan dataset

Generative Priors:

- Train our model use degradation condition to bridge the sim-real gap
- Develop the keyframe interpolation method

Thank you!