# Event-driven Video Frame Synthesis

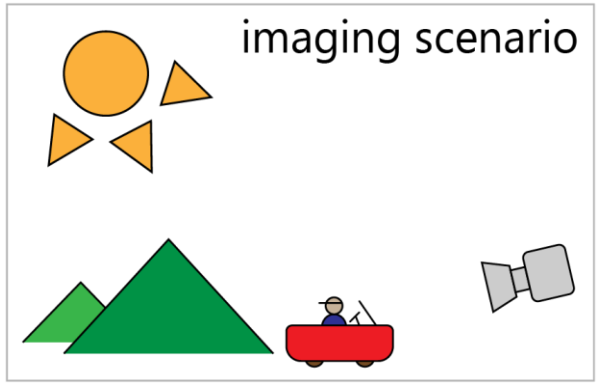
**Zihao Wang**[1], Weixin Jiang[1], Kuan He[1], Boxin Shi[2],
Aggelos Katsaggelos[1], Oliver Cossairt[1]

[1] Northwestern University [2] Peking University

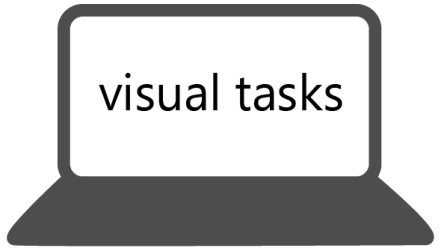


2nd Int'l Workshop on Physics Based Vision meets Deep Learning (PBDL) in conjunction with ICCV 2019 Seoul, Korea

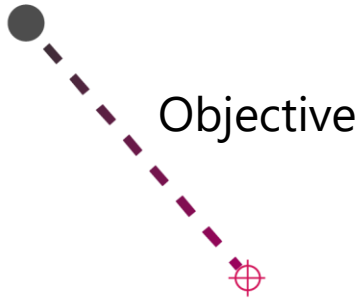
# Physics-based vision meets deep learning




**Physics-based vision**



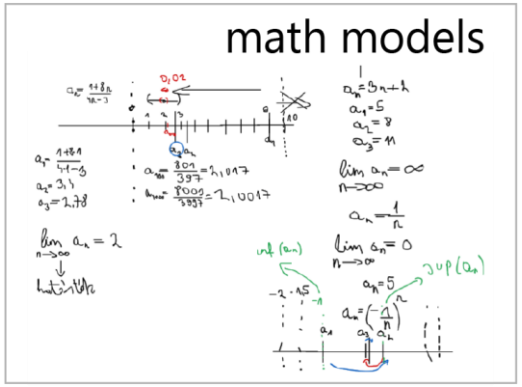

Initial point



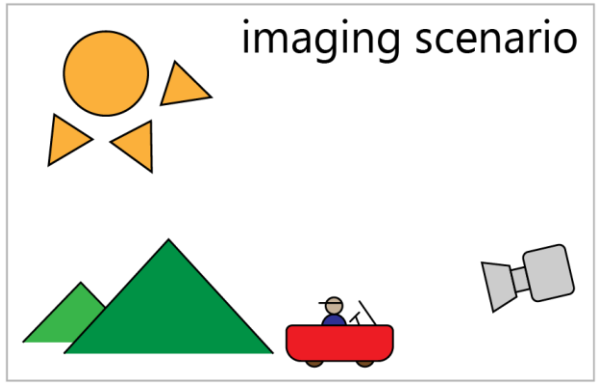
Objective

Target

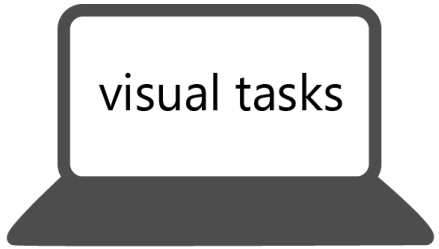
Designed priors
following physics laws



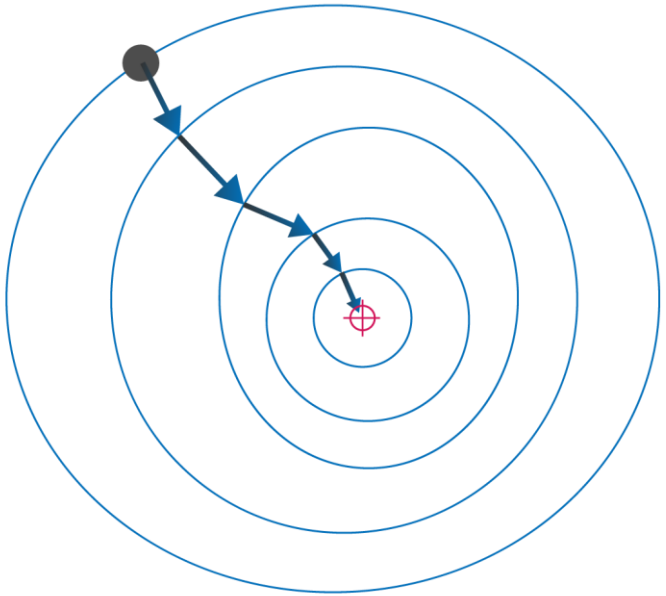
# Physics-based vision meets deep learning




**Physics-based vision**



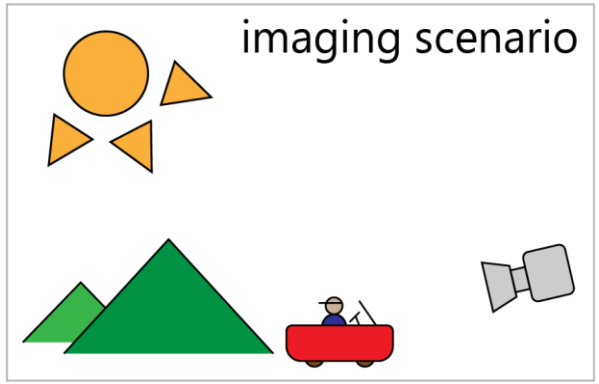

Initial point



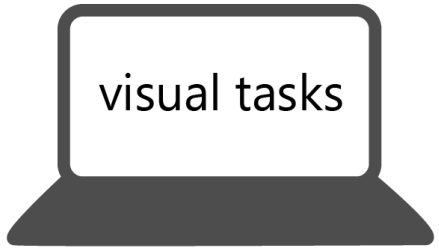
Physics-based optimization

Subject to noise, or
incomplete modeling

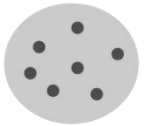
# Physics-based vision meets deep learning




**Physics-based vision**



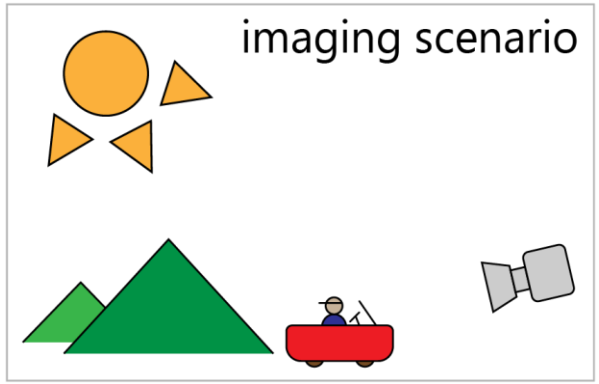

Noisy input



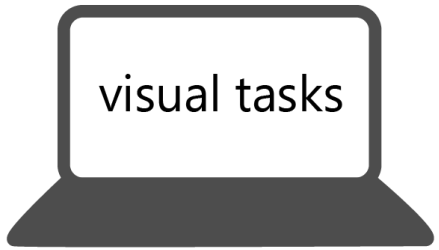
Physics-based optimization

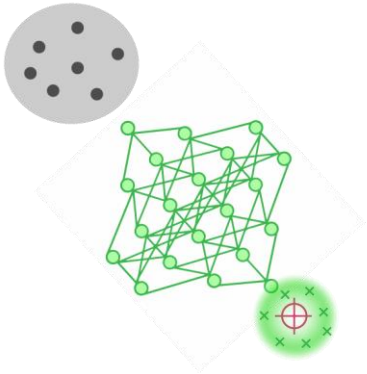
Subject to noise, or incomplete modeling

# Physics-based vision meets deep learning




**Learning-based vision**

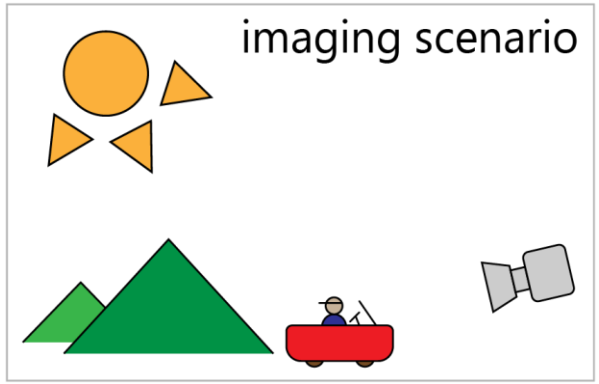


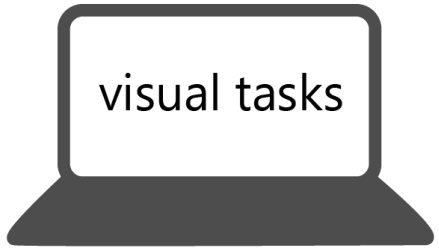
End-to-end non-linear fitting

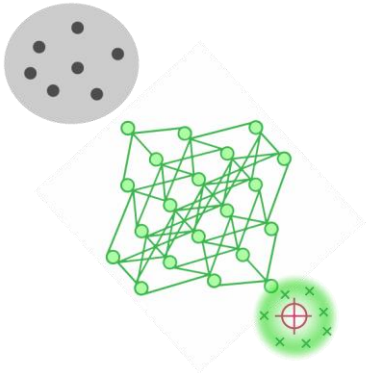
usually have
superior performance
as long as you have
sufficient data and GPUs

# Physics-based vision meets deep learning




**Learning-based vision**

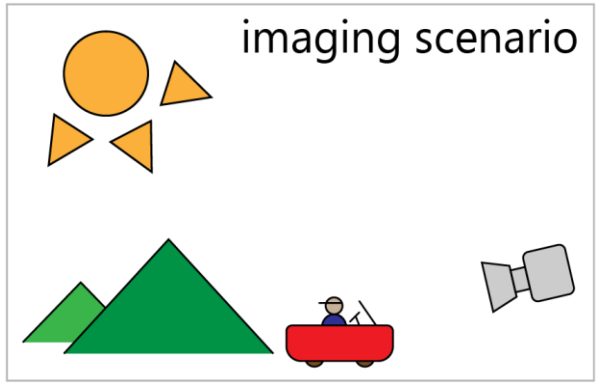


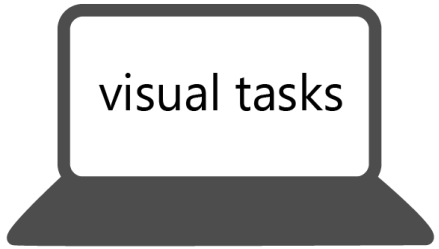
End-to-end non-linear fitting

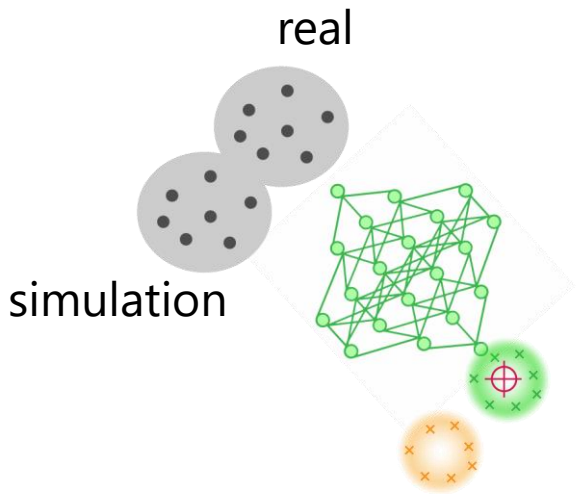
Learning from simulation!

# Physics-based vision meets deep learning




**Learning-based vision**

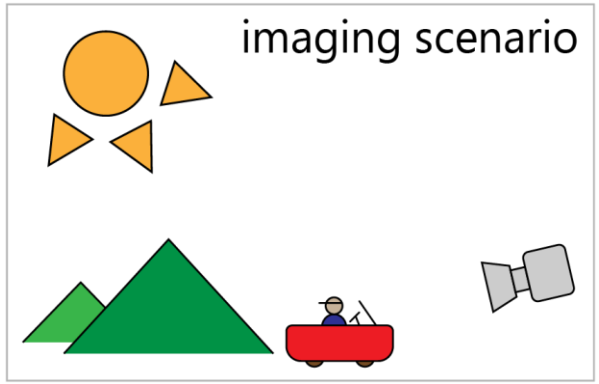


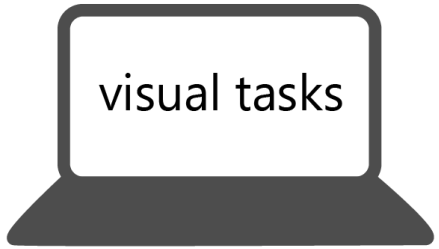

End-to-end non-linear fitting

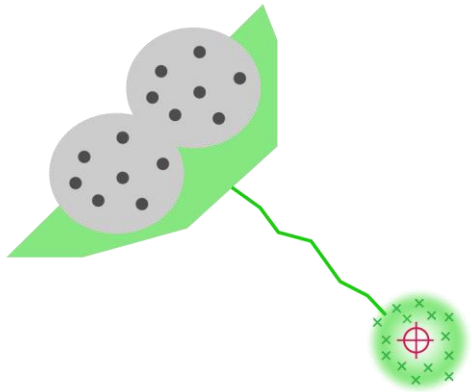
Learning from simulation!
The gap between
simulation and real data

# Physics-based vision meets deep learning




**Learning-based vision**

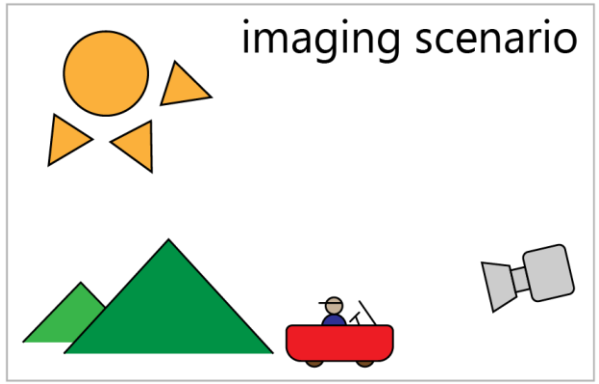


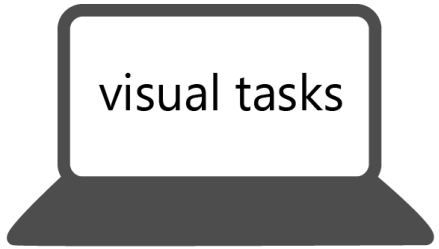
End-to-end non-linear fitting

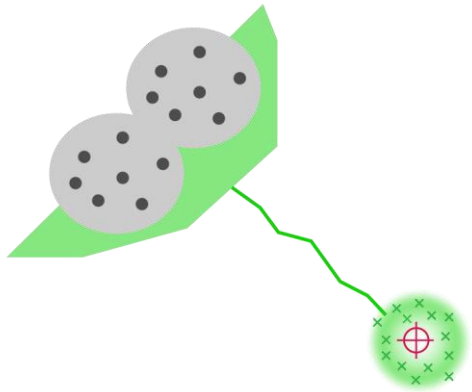
Train with data augmentation

# Physics-based vision meets deep learning




**Learning-based vision**

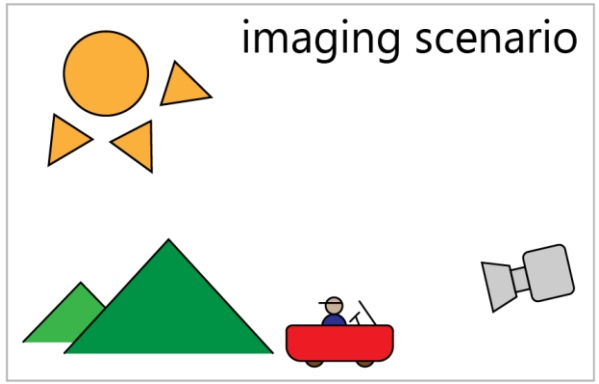


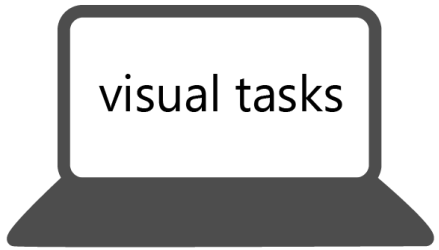
End-to-end non-linear fitting

Require retraining even
for similar tasks.

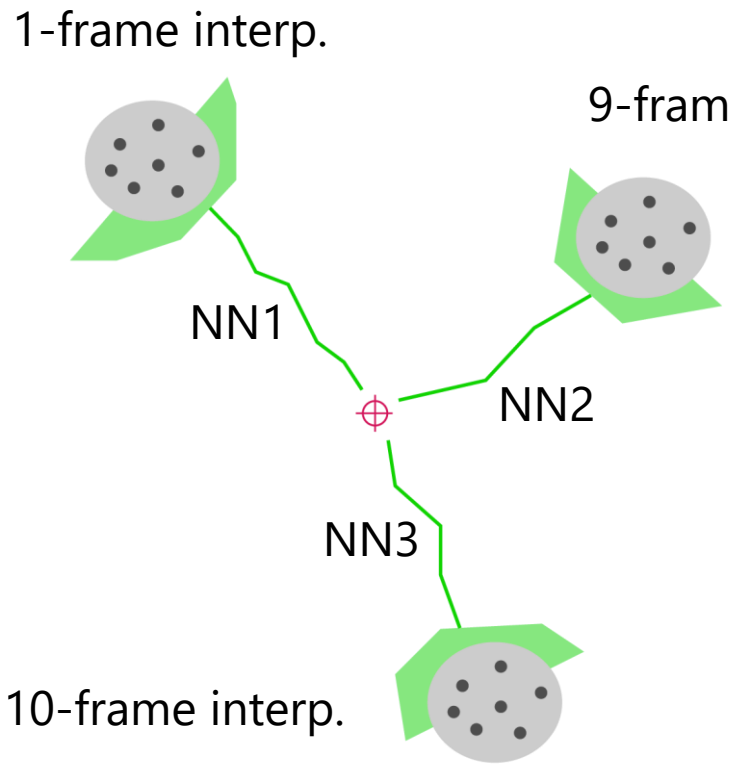
# Physics-based vision meets deep learning




**Learning-based vision**



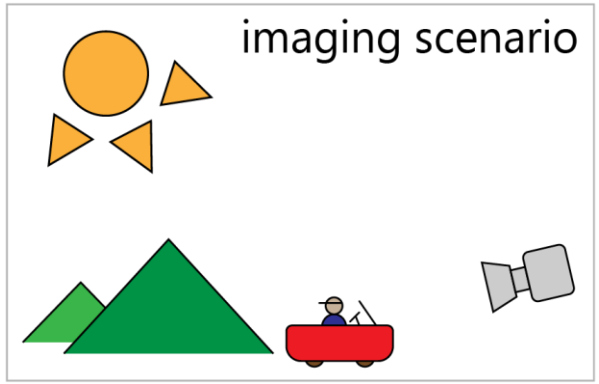

E.g. video frame interpolation



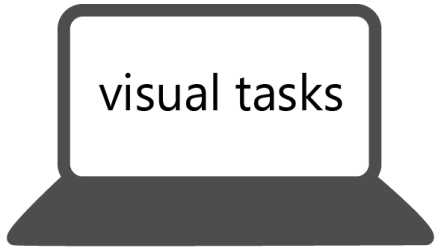

End-to-end non-linear fitting

Require retraining even for similar tasks.

# Physics-based vision meets deep learning

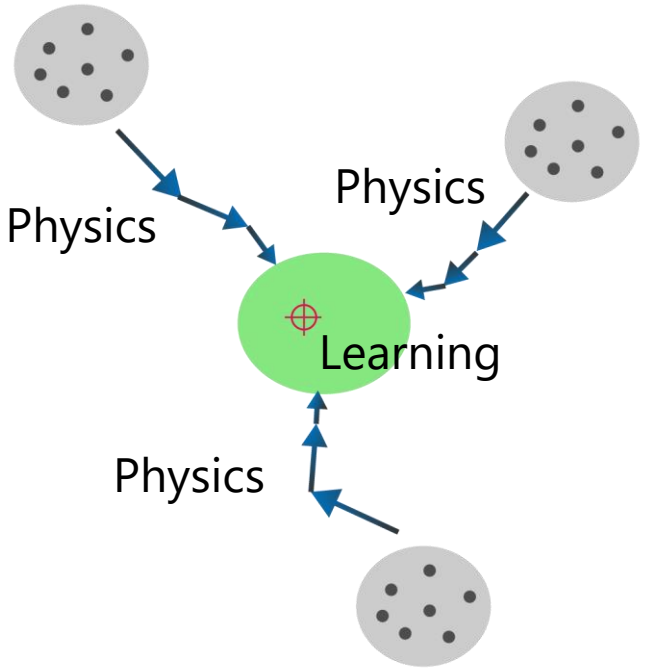



**Physics + learning based vision**



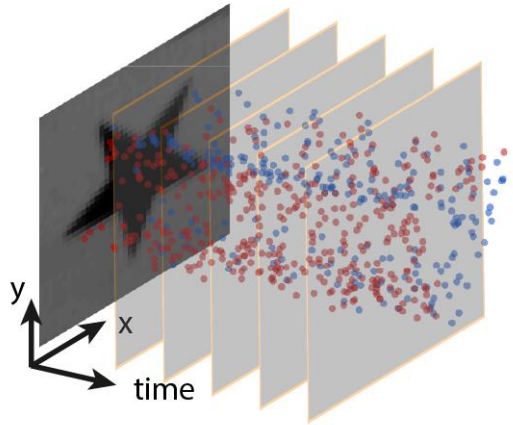

Framework:

- First use a unifying physics-based approach to have rough estimation
- Then use DNN to learn the residual.

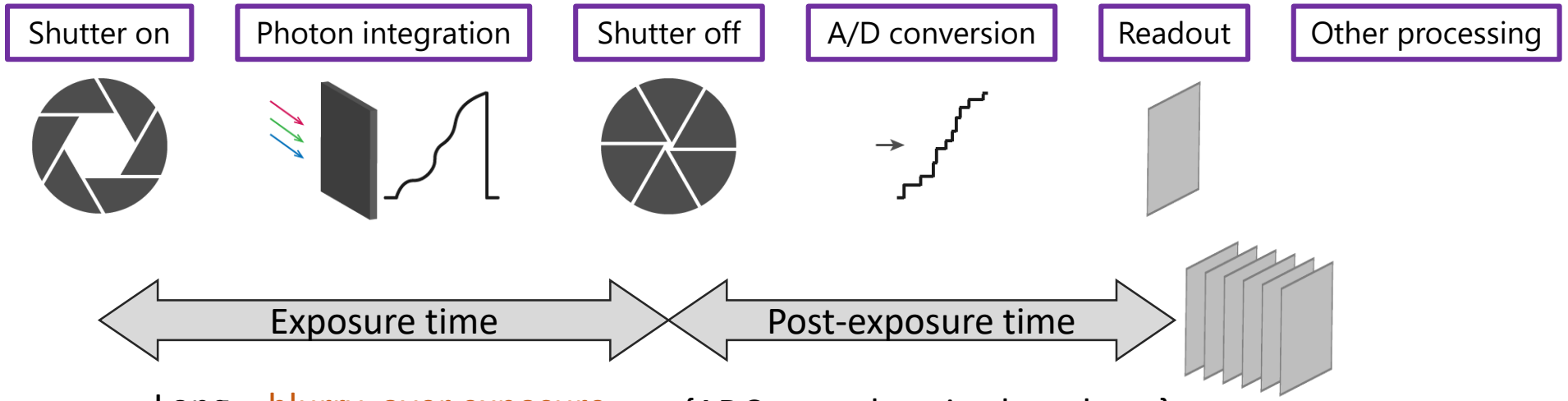



Suitable application:

multi-modal video synthesis



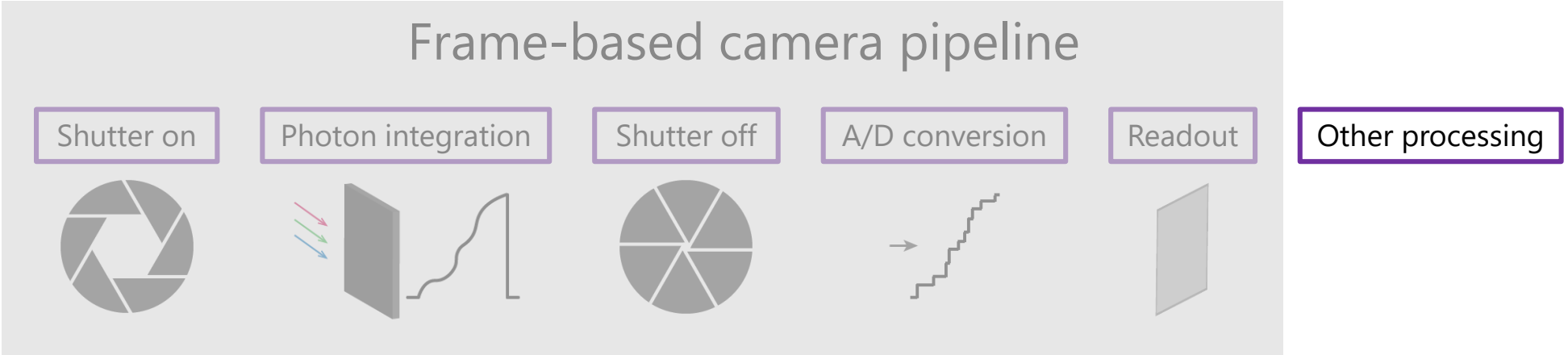
# Background: rethinking frame-based imaging




Long – blurry, over exposure
Short – noisy, under expo.

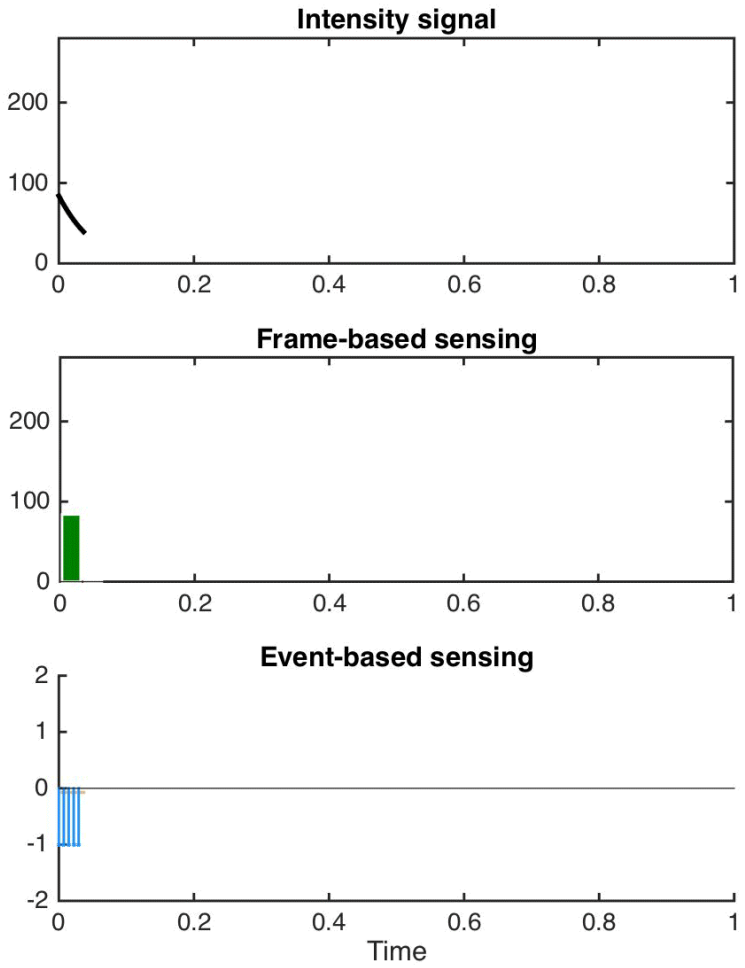
{ADC + synchronized read-out}
– discrepancy between frames
(frame interpolation)

# Motivation

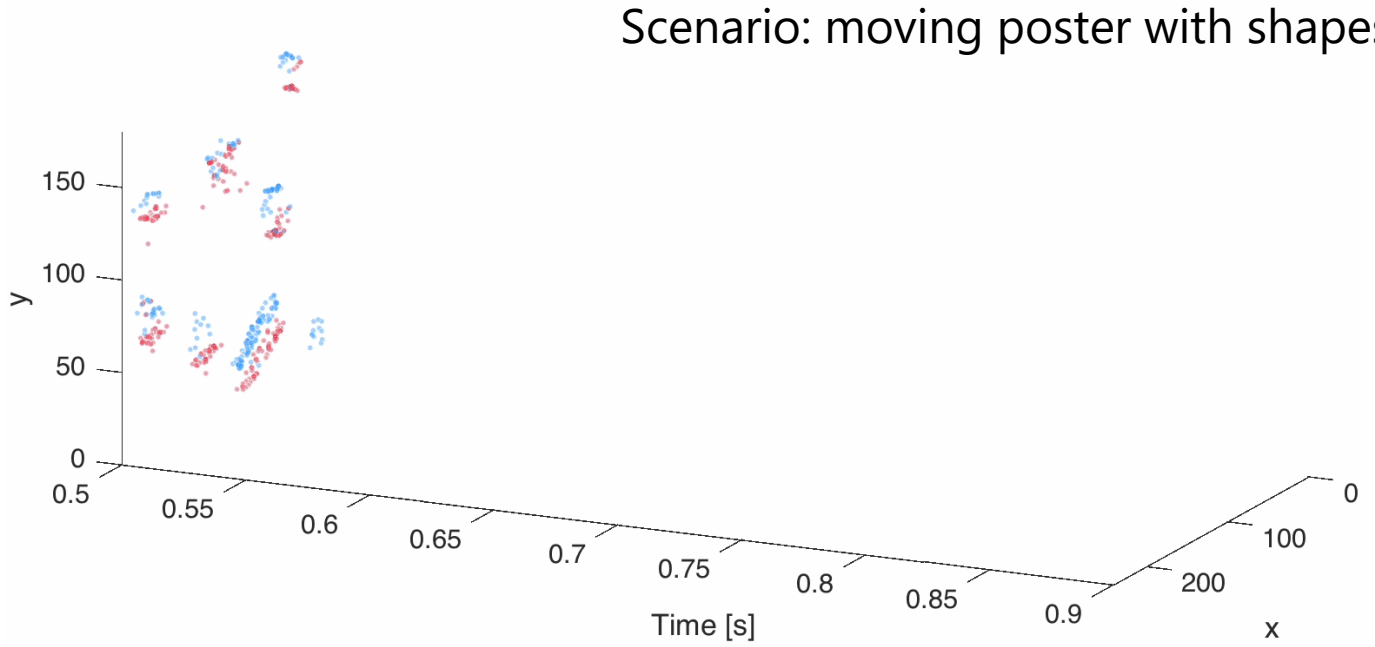



- We need "smart" cameras that:
  - Can respond to high speed motions (eliminate blur)
  - Do not always operate at high speed (less data redundancy)
- Potential solution: event cameras

# What's event camera? Another high-speed camera?




Scenario: moving poster with shapes




Data from DAVIS dataset

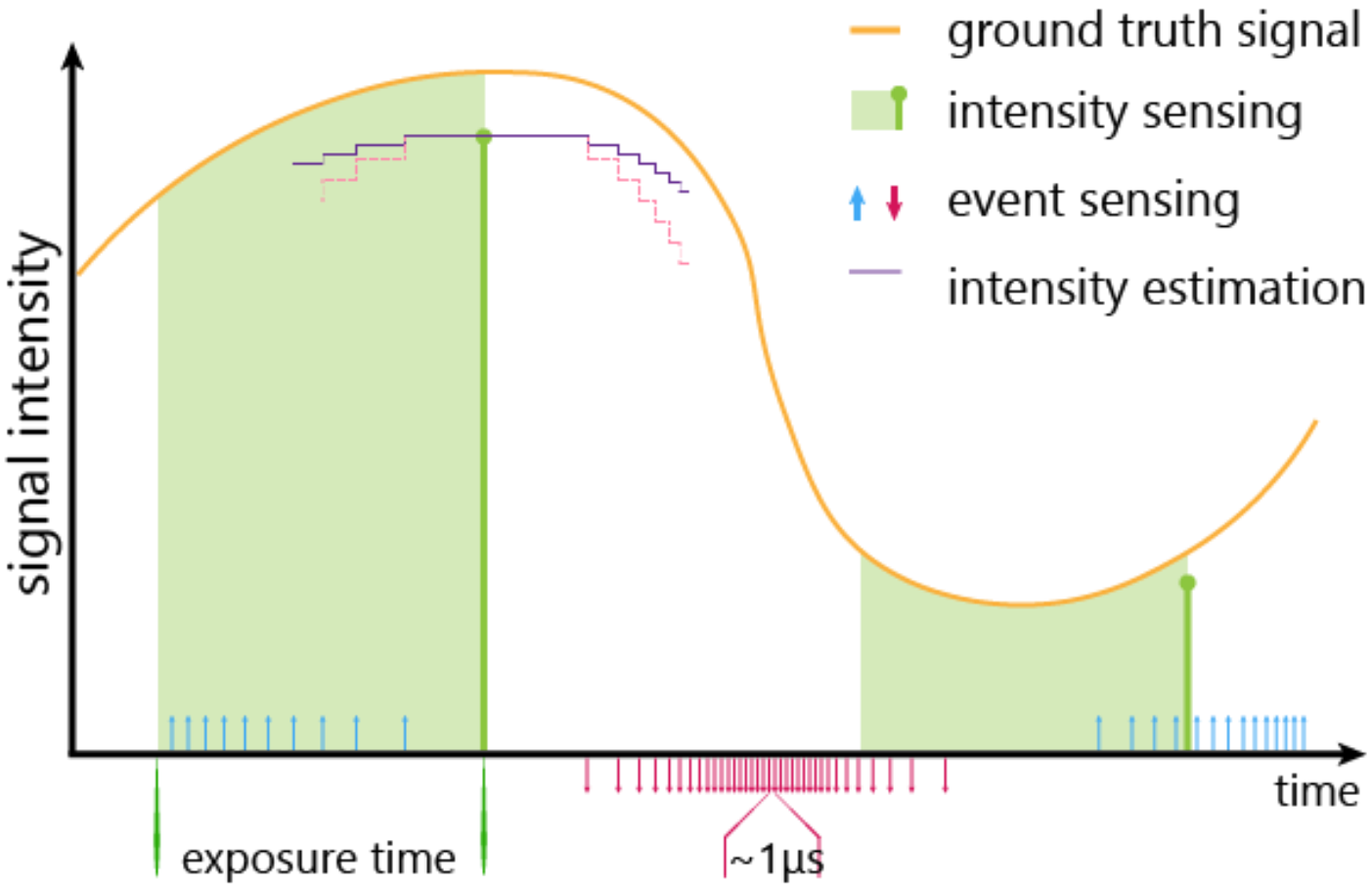
Each pixel:

- Compare brightness variations
  - (blue: increase; red: decrease)
- Small latency (micro-second level)
  - $10^6$ FPS! (at max)
- Works independently (asynchronous)

Capture: 22 FPS  Display: 1.1 FPS

# But…

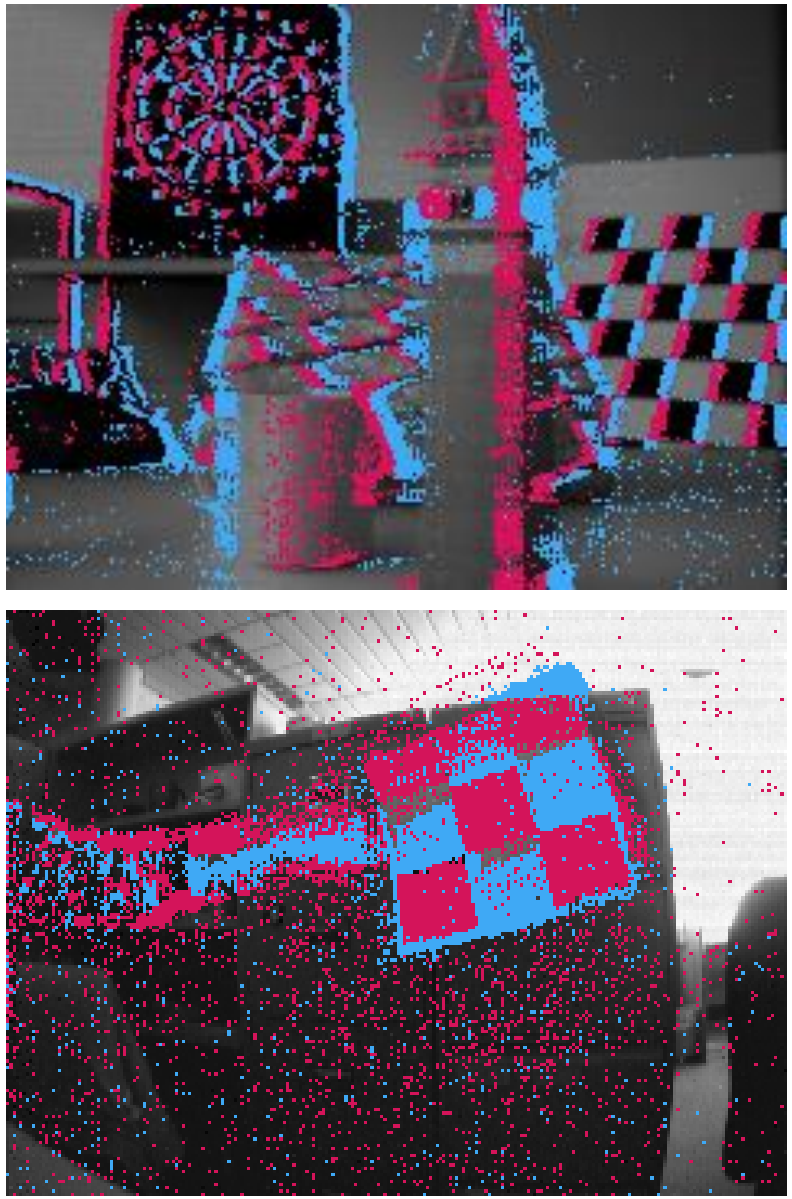
- Events ≠ temporal gradient
  - Infinitely many solutions to infer intensity from events.
  - Cannot capture weak variations



# But…

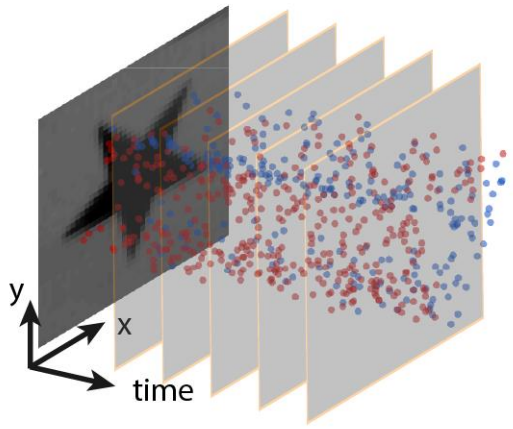
- Events ≠ temporal gradient
  - Infinitely many solutions to infer intensity from events.
  - Cannot capture weak variations

- Events are very noisy
  - Noise model not well understood
    - Gaussian on threshold
  - Event denoisers not advanced
    - Able to cancel isolated events (correlation)
    - Cannot handle complex scenarios, e.g. illumination change

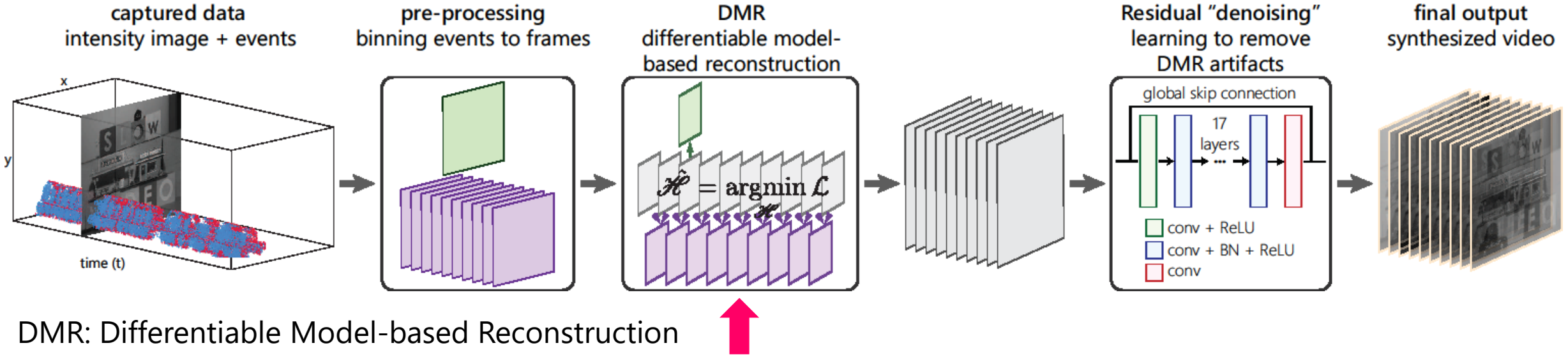


Example images overlaid with neighbor events data from DAVIS dataset and Pan et al. CVPR'19

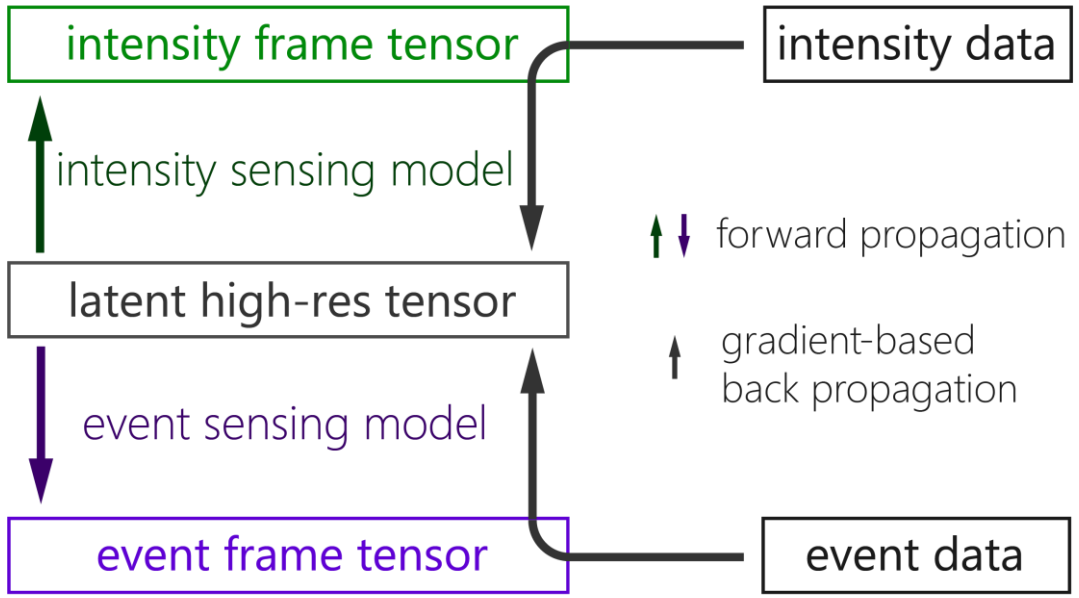
We propose intensity frame + events
for high frame-rate video synthesis



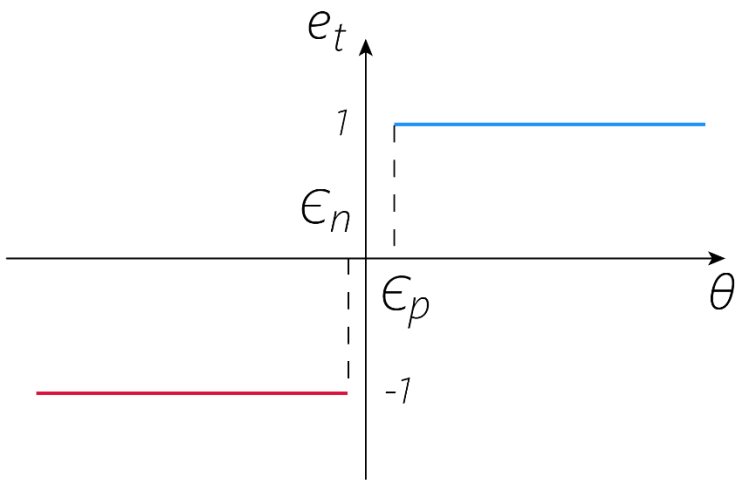
# Our approach: fusion of intensity frame + events



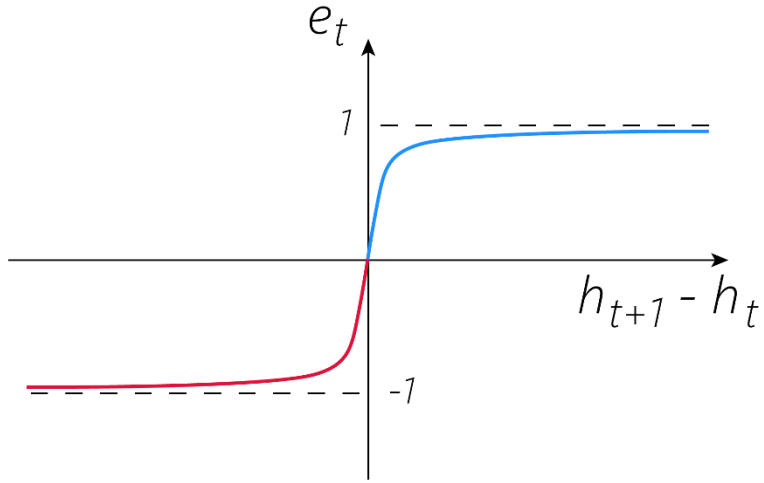

DMR: Differentiable Model-based Reconstruction



# Differentiable model (event sensing)




Per-pixel sensing model




Differentiable model (approx.)

$$\theta = \log(I_t + b) - \log(I_0 + b)$$

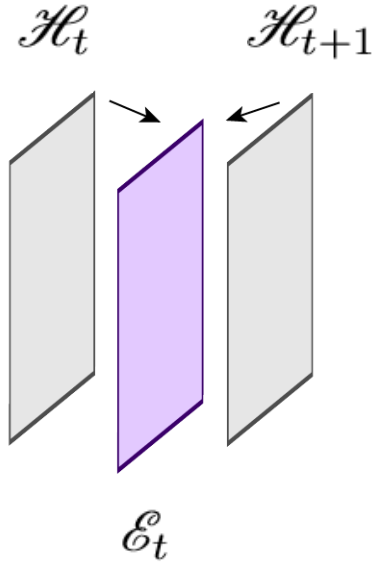
$$e_t = \begin{cases} 1 & \theta > \epsilon_p \\ -1 & \theta < -\epsilon_n \\ 0 & otherwise \end{cases}$$

$h_t$ denotes one pixel of $\mathscr{H}_t$

$\mathscr{H}_t$ is t[th] frame of $\mathscr{H} \in \mathbb{R}^{h \times w \times d}$

$$\mathscr{E}_t = \tanh\left\{\alpha[\mathscr{H}_{t+1} - \mathscr{H}_t]\right\}$$

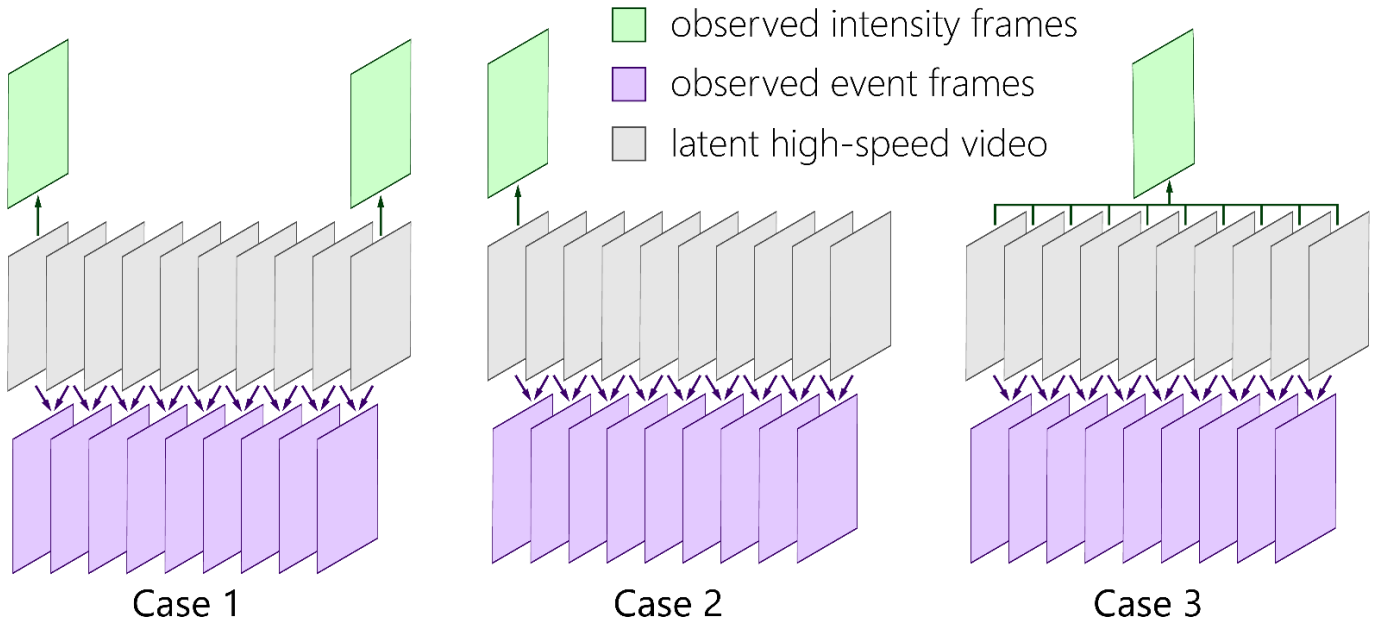
t[th] event frame



# Differentiable model (frame sensing)

We consider 3 temporal settings: interpolation, prediction and motion deblur.

| Case (intensity tensor notation) | Model |
|---|---|
| Interpolation ($\mathscr{F}^i$) | $\mathscr{F}^i_1 = \mathscr{H}_1, \mathscr{F}^i_2 = \mathscr{H}_d$ |
| Prediction ($\mathscr{F}^p$) | $\mathscr{F}^p = \mathscr{H}_1$ |
| Motion deblur ($\mathscr{F}^m$) | $\mathscr{F}^m = \frac{1}{d}\sum_{t=1}^{d} \mathscr{H}_t$ |



# Reconstruction loss and optimization

Objective $$\hat{\mathscr{H}} = \underset{\mathscr{H}}{\operatorname{argmin}}\, \mathcal{L}_{pix}(\mathscr{H}, \mathcal{F}, \mathcal{E}) + \mathcal{L}_{TV}(\mathscr{H})$$

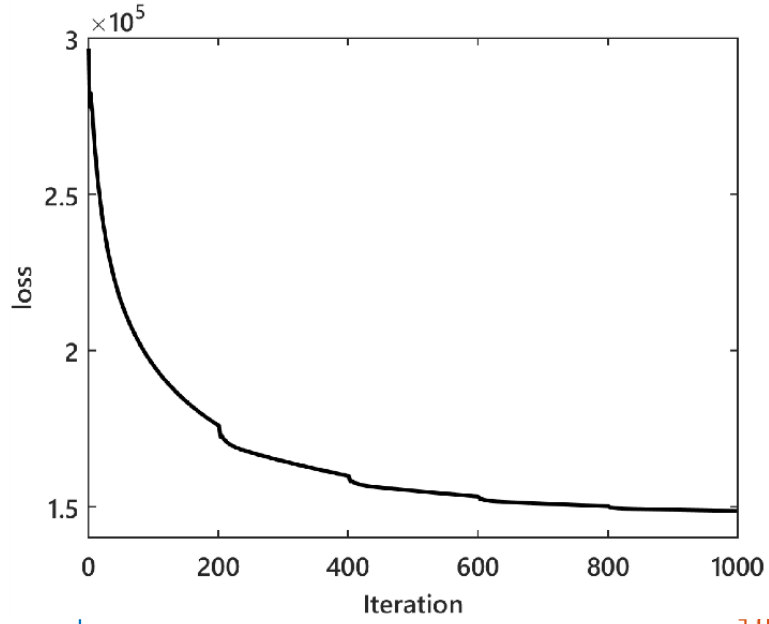
Frame pixel error

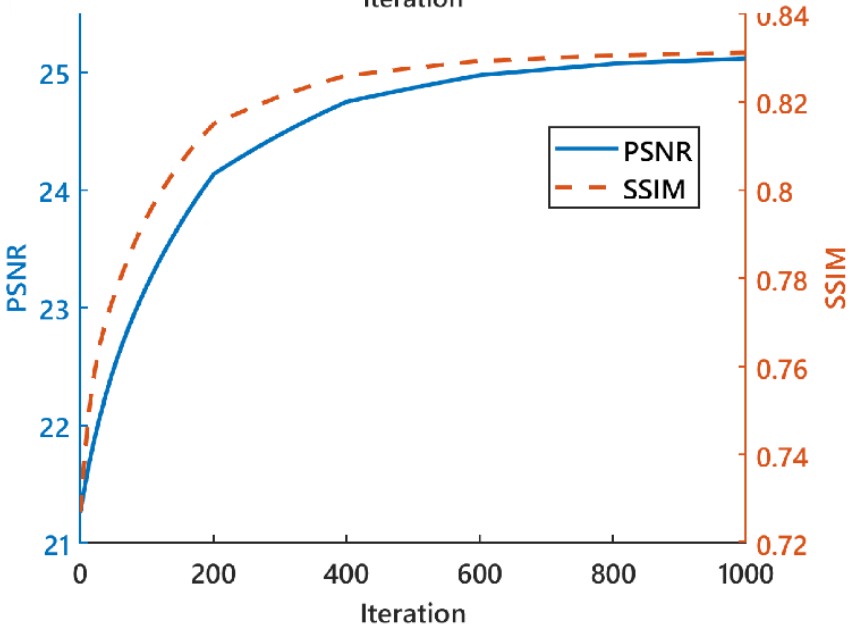
Pixel loss $$\mathcal{L}_{pix}(\mathscr{H}, \mathcal{F}, \mathcal{E}) = \mathbb{E}_{fpix}[\|\mathcal{F} - \mathcal{A}(\mathscr{H})\|_1] + \lambda_e \mathbb{E}_{epix}[\|\mathcal{E} - \mathcal{B}(\mathscr{H})\|_1]$$

Event pixel error

Sparsity loss $$\mathcal{L}_{TV}(\mathscr{H}) = \lambda_{xy}\mathbb{E}_{hpix}\left[\left\|\dot{\mathscr{H}}_{xy}\right\|_1\right] + \lambda_t \mathbb{E}_{hpix}\left[\left\|\dot{\mathscr{H}}_t\right\|_1\right]$$

$$\dot{\mathscr{H}}_{xy} = \frac{\partial \mathscr{H}}{\partial x} + \frac{\partial \mathscr{H}}{\partial y} \qquad \dot{\mathscr{H}}_t = \frac{\partial \mathscr{H}}{\partial t}$$




Use stochastic gradient descent (SGD) to minimize the loss.
As loss decreases, results get closer to ground truth.

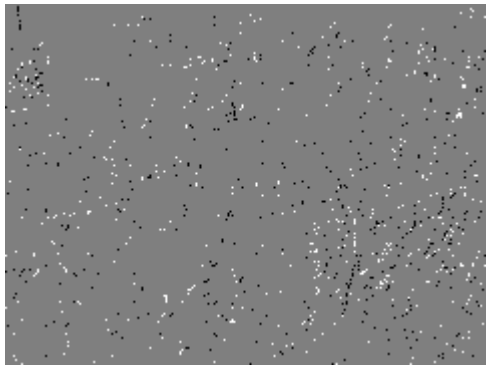
# Results (DMR)

- Interpolation case
  - Given start & end frames + events in-between, recover intermediate frames

Low-speed intensity frames (2 frames)

The middle frame is withheld for evaluation

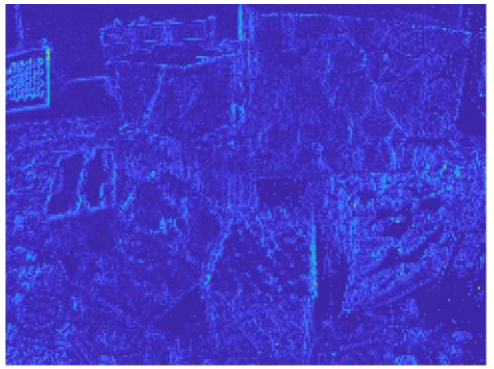
Event frames (20 frames)



High-speed video (21 frames)

PSNR: 33.41 SSIM: .955

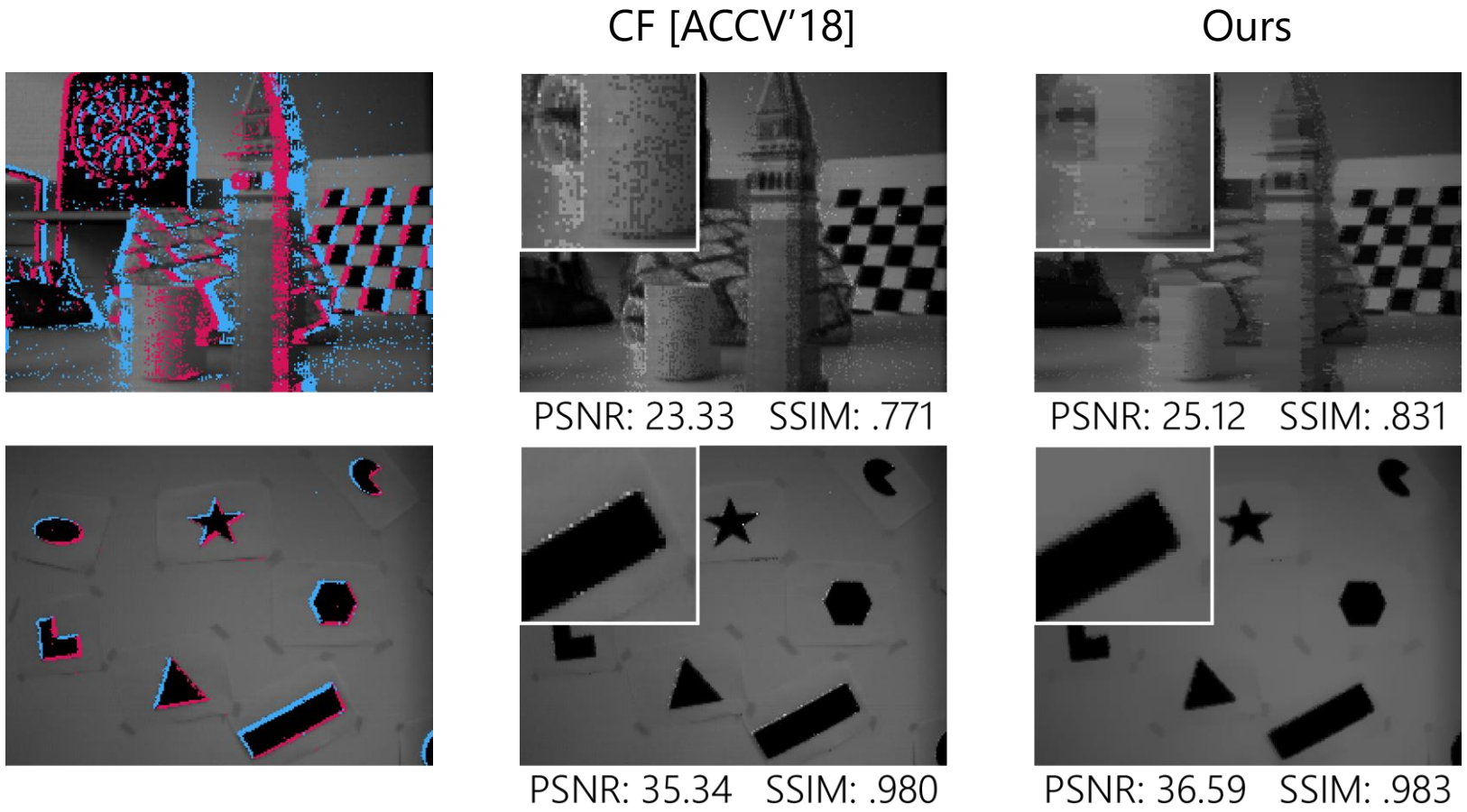
Frame #10




Error map of Frame #10

# Results (DMR)

- Prediction case
  - Given start frame and future events, recover future frames

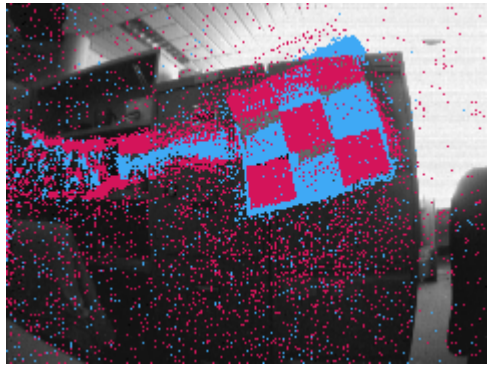


# Results (DMR)

- Motion deblur case
  - Given a blurry image + events in-exposure, recover intermediate sharp frames.

Blurry images

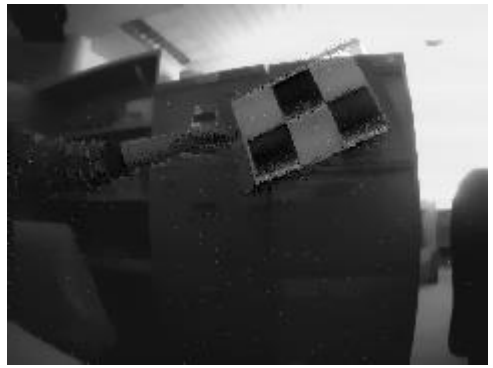
Events during exposure



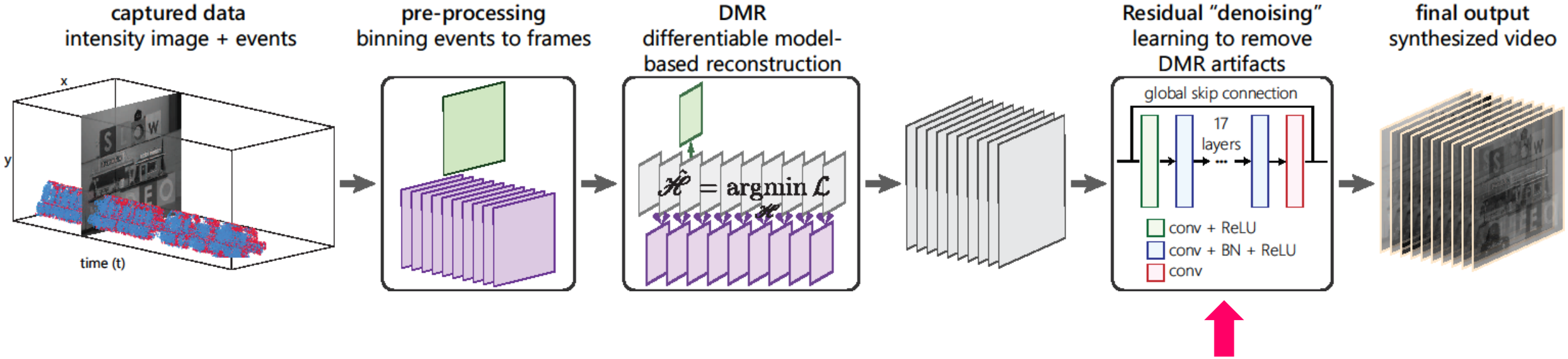
EDI [CVPR'19]

Ours

Video recovery

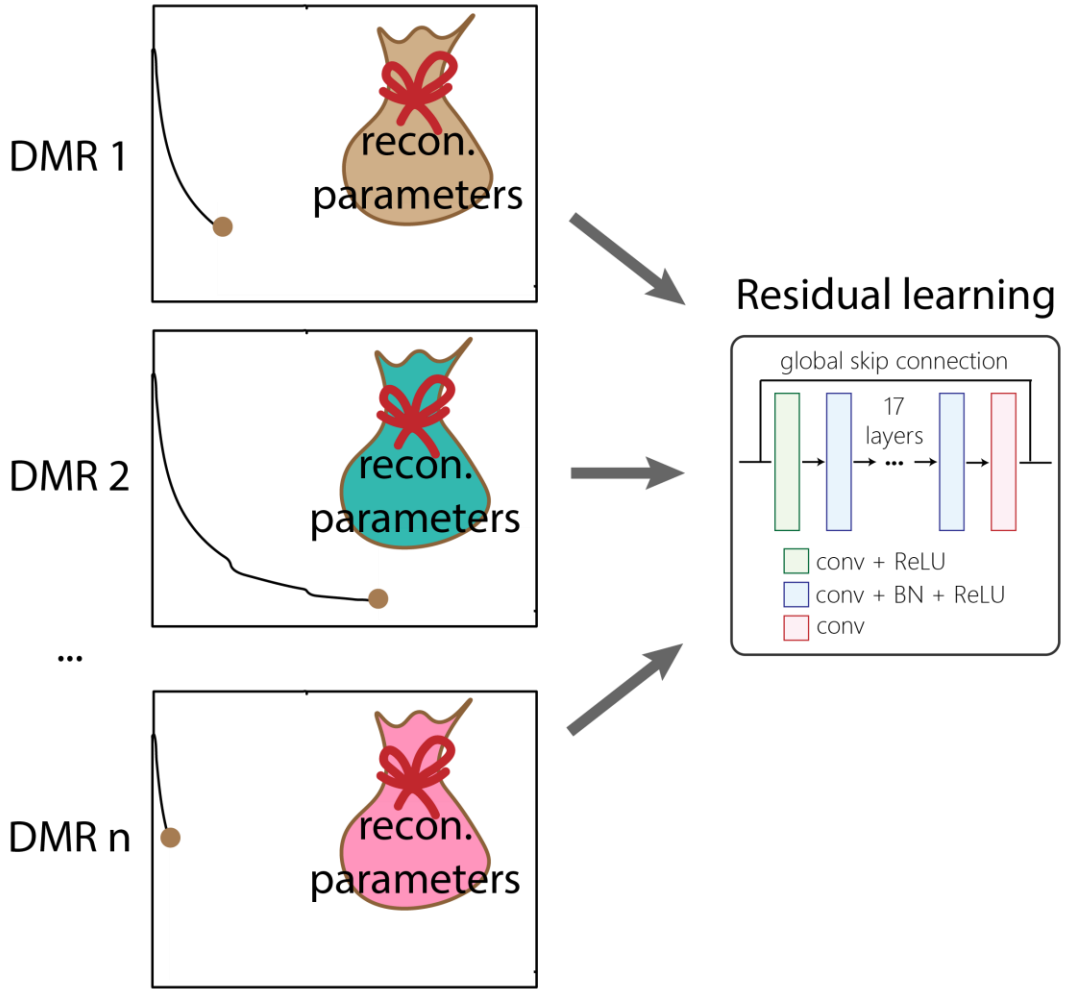


# Overview of our approach

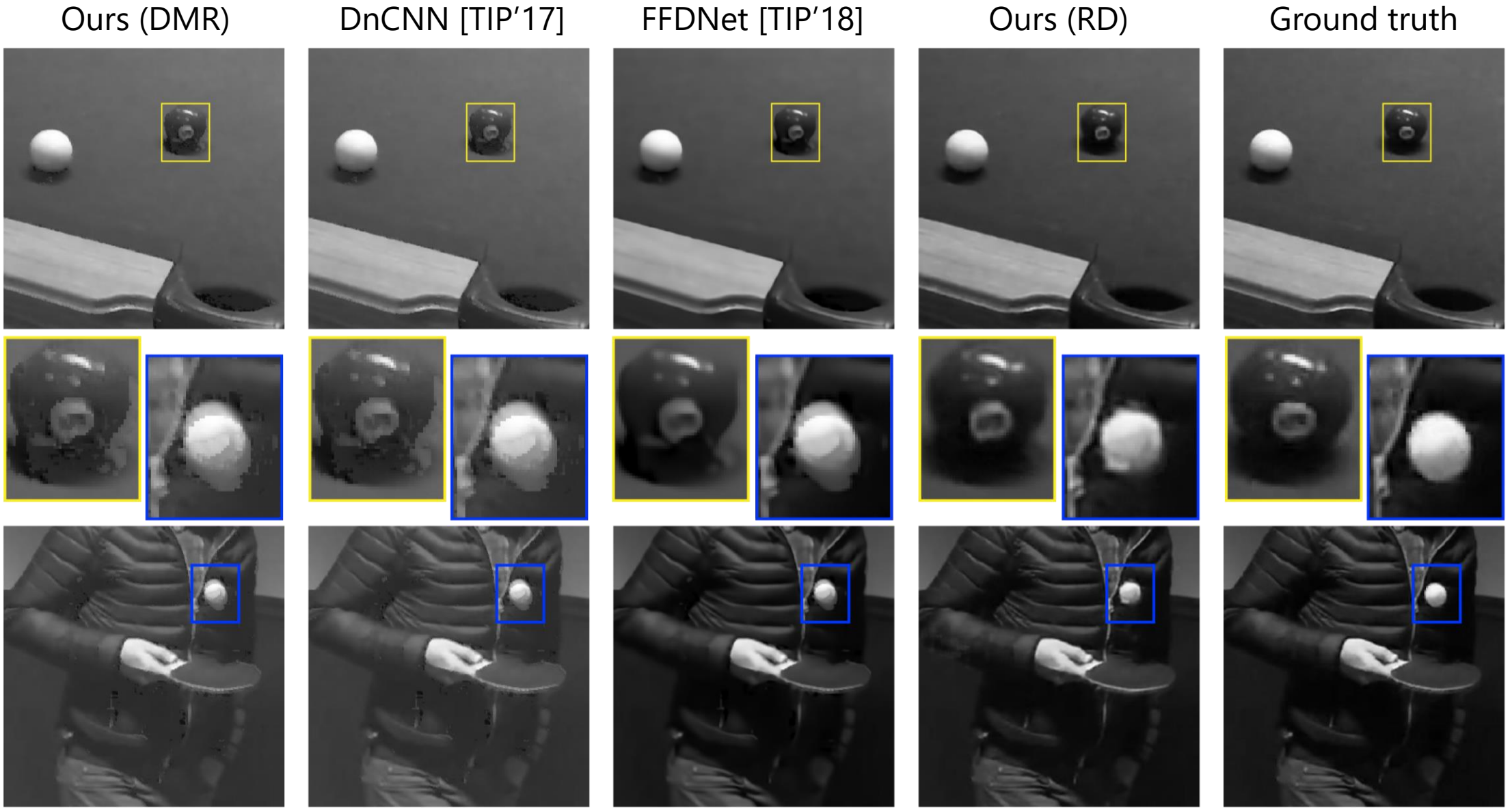


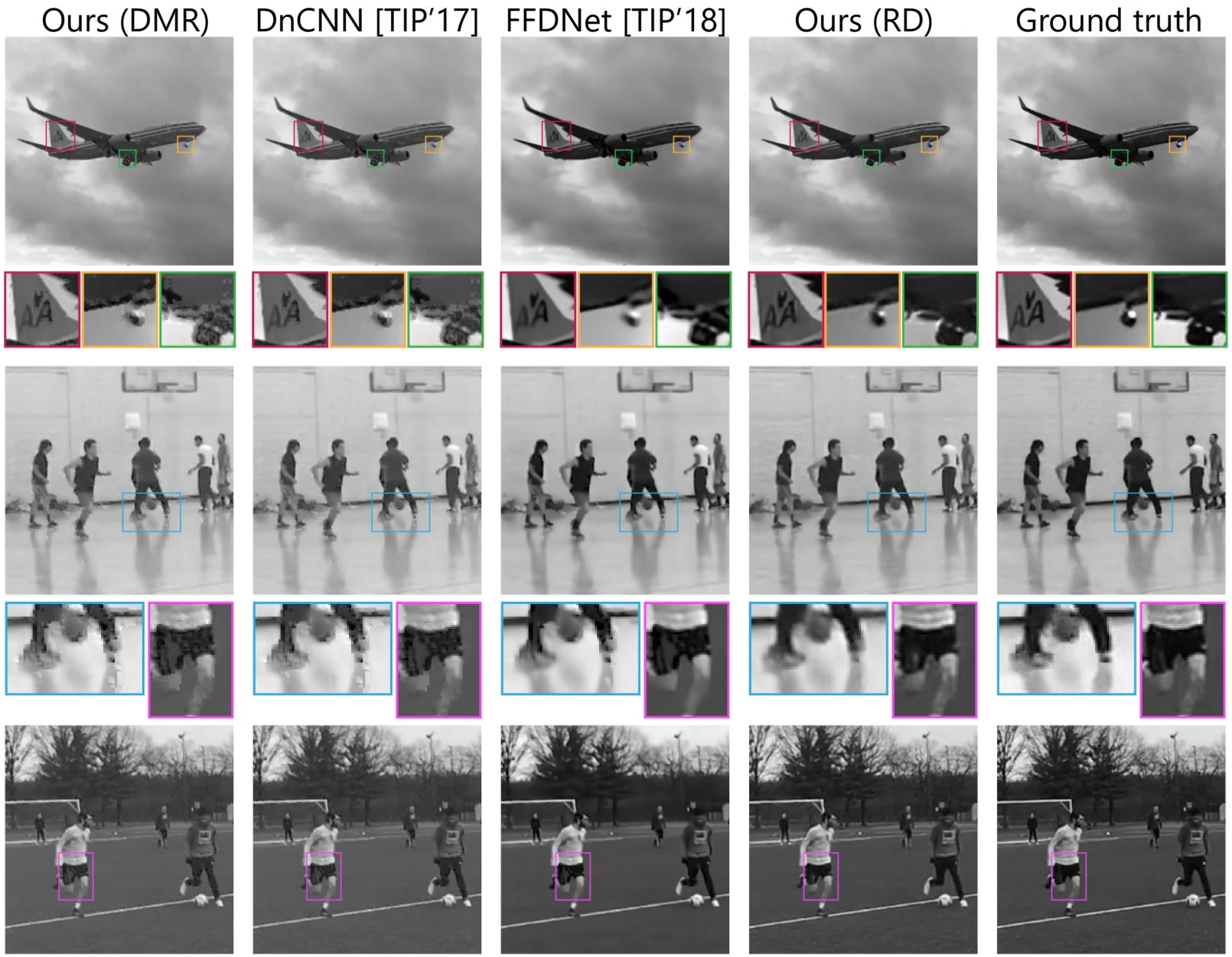
# Residual “denoiser”

- Use CNN to learn the residual of DMR output w.r.t. ground truth
  - Designed to enhance DMR results
- Easy to train
  - Model DMR artifacts as residual “noise”
  - Actually beyond Gaussian denoising
- Single frame based
  - Interface well with DMR



# Results (residual denoiser)






| clip name | metric | DMR | DnCNN | FFDNet | Ours |
|---|---|---|---|---|---|
| airplane | PSNR | 30.91 | 31.10 | 30.92 | **31.38** |
| | SSIM | 0.975 | 0.982 | 0.976 | **0.982** |
| basketball | PSNR | 23.55 | 24.05 | 23.47 | **24.06** |
| | SSIM | 0.963 | 0.971 | 0.964 | **0.972** |
| soccer | PSNR | 29.96 | 31.08 | 30.13 | **31.29** |
| | SSIM | 0.961 | 0.974 | 0.962 | **0.975** |
| billiard | PSNR | 36.46 | 35.42 | **36.48** | 36.46 |
| | SSIM | 0.982 | 0.986 | 0.983 | **0.987** |
| ping pong | PSNR | 32.46 | 32.26 | **32.50** | 32.24 |
| | SSIM | 0.974 | 0.978 | 0.975 | **0.979** |

# Results

- Comparison with non-event-based frame interpolation approach
  - Events can provide additional information which is useful for challenging motions.

SepConv [CVPR'17]

Ground truth

Ours (DMR + RD)

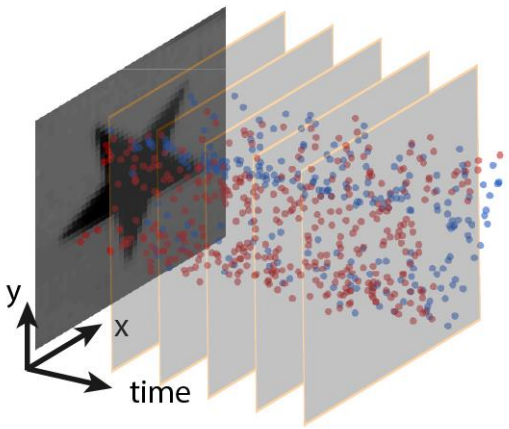
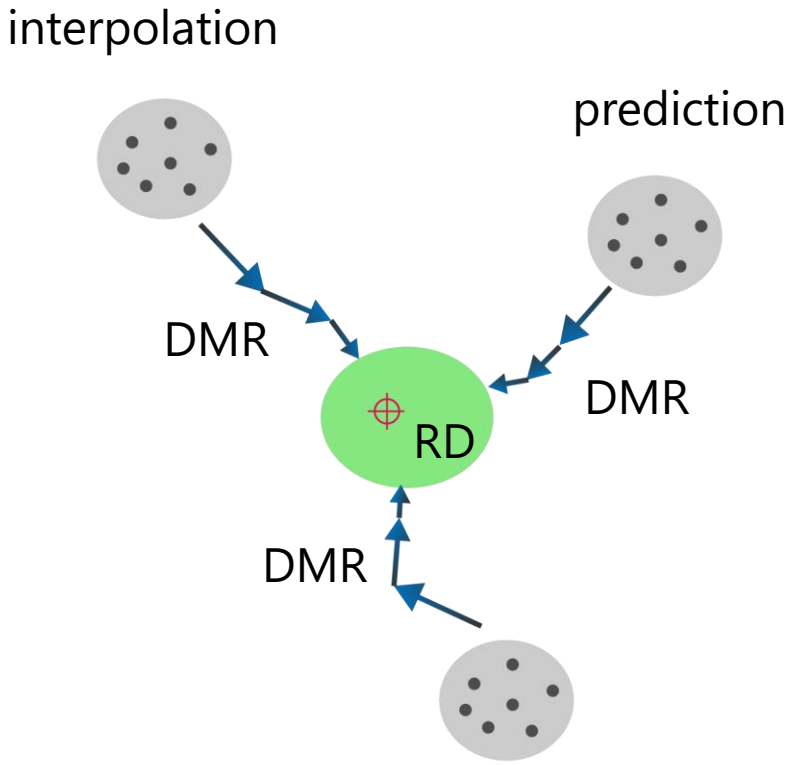



image + events




motion deblur

Thank you!
Zihao (Winston) Wang
zwinswang@gmail.com