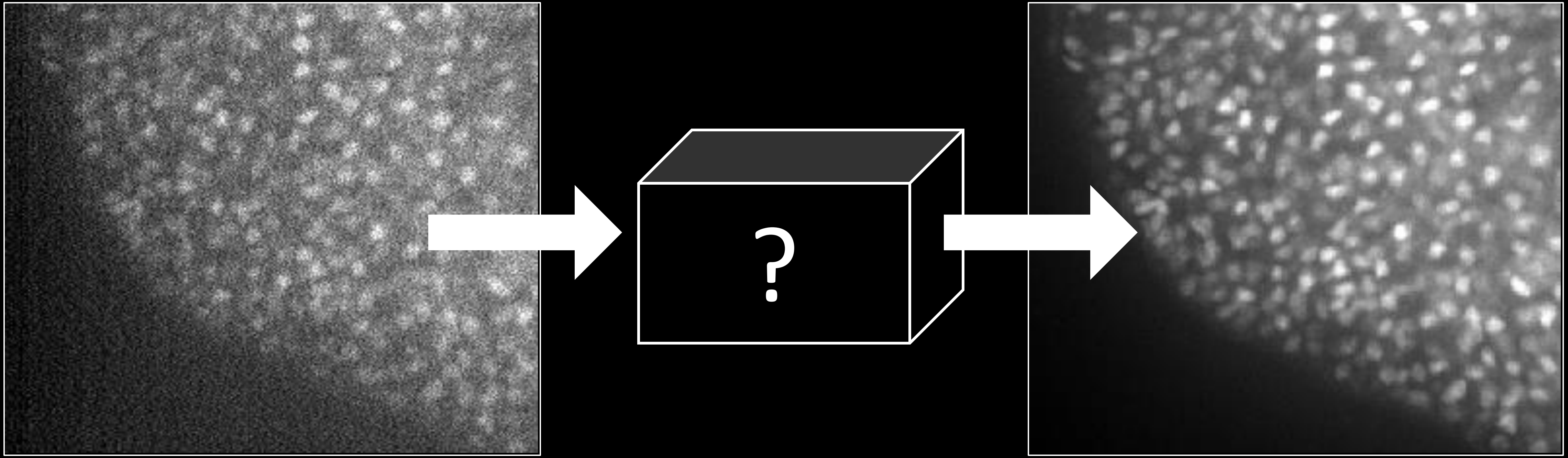
UNIVERSITY OF BIRMINGHAM

Noise and Denoising: Supervised, Self-Supervised and Unsupervised

Alexander Krull

The Problem of Noise



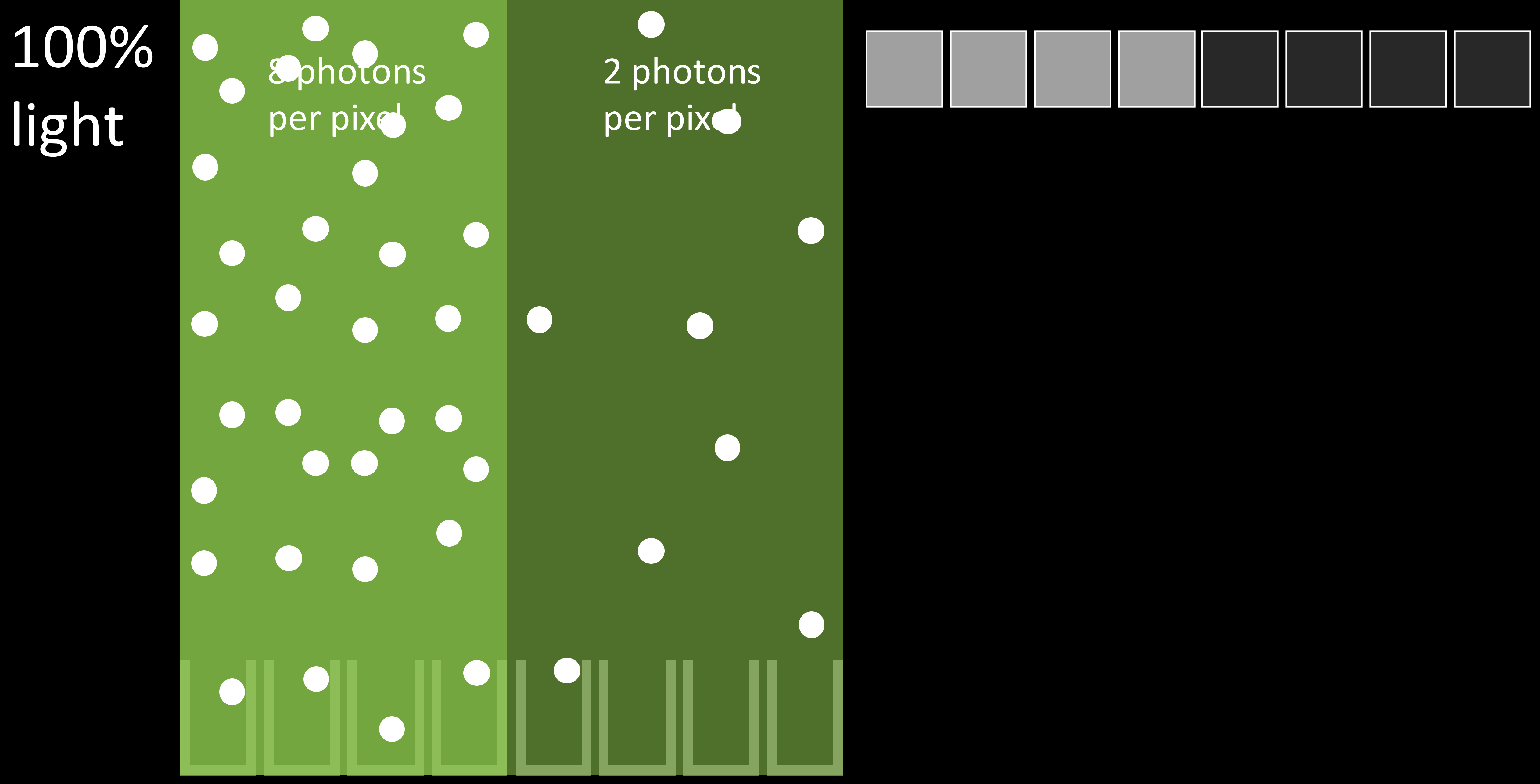
Low exposure:

- Gentle ☺
- Noisy ☹

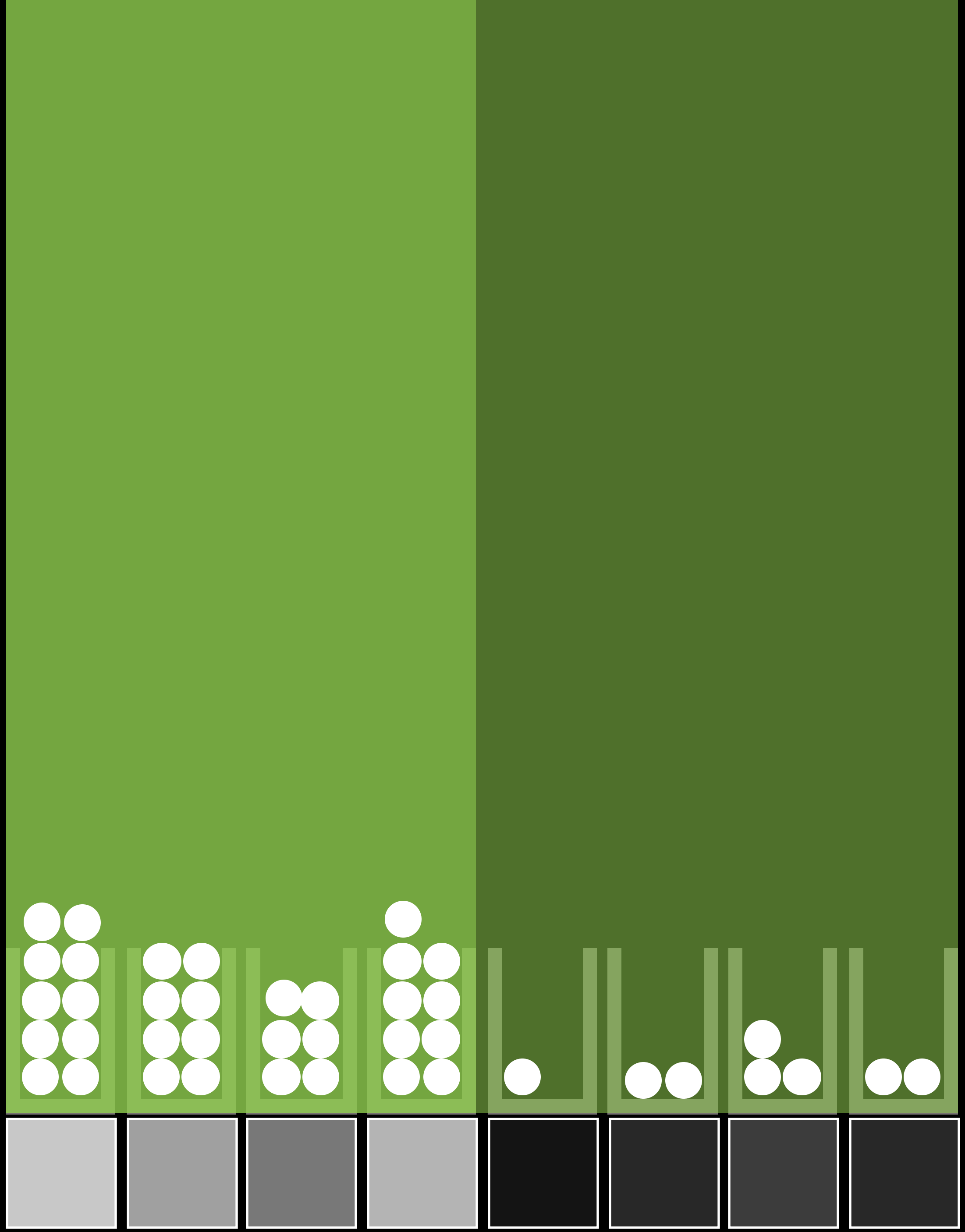
High exposure:

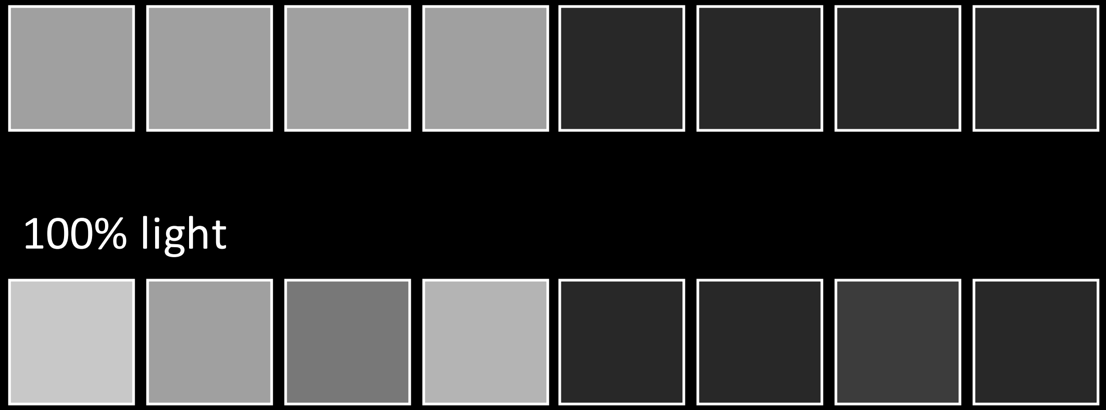
- Damaging ☹
- Clean ☺

Why does Low Light Lead to Noisy Images?

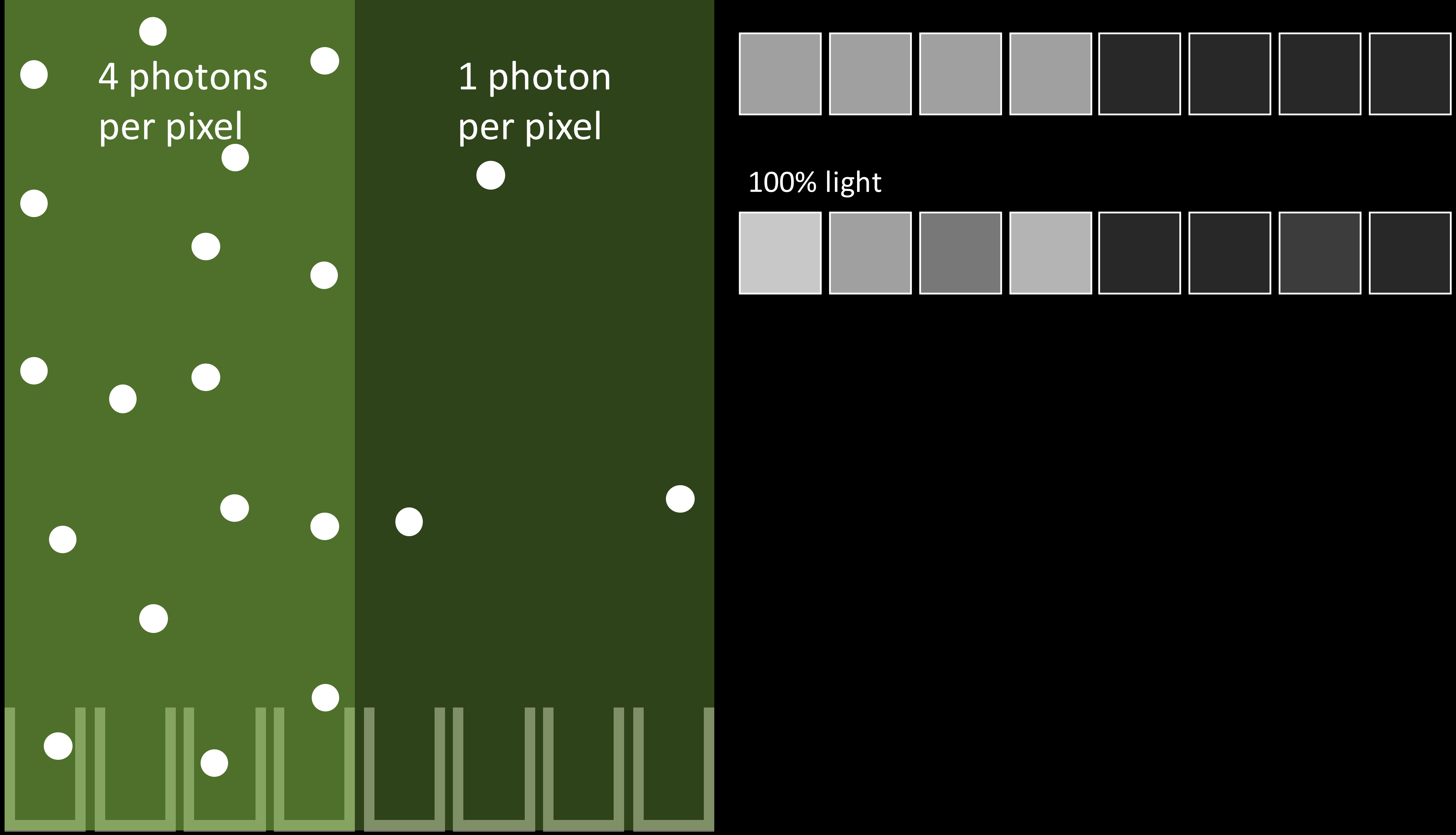

100% light
8 photons per pixel
2 photons per pixel

100% light

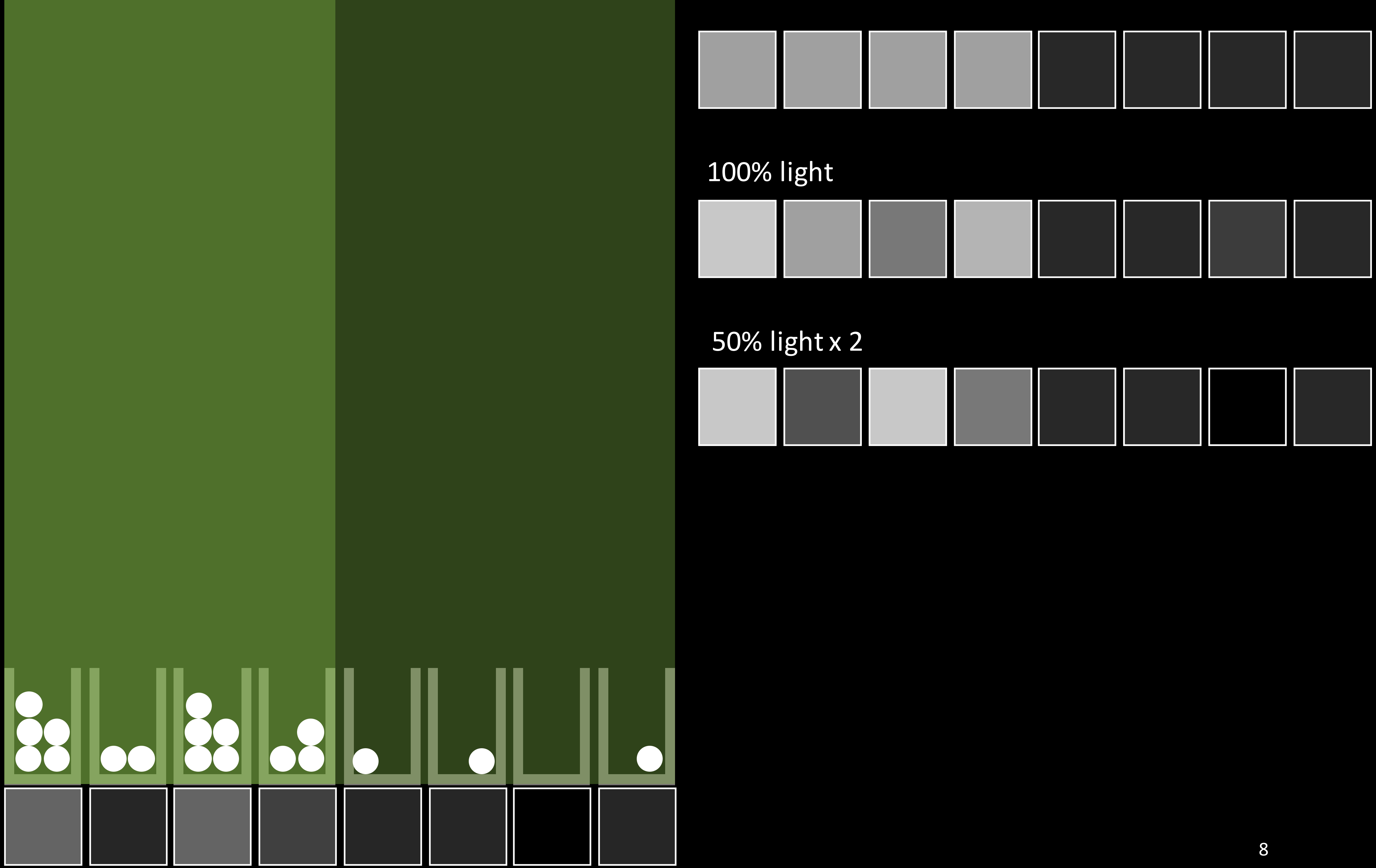




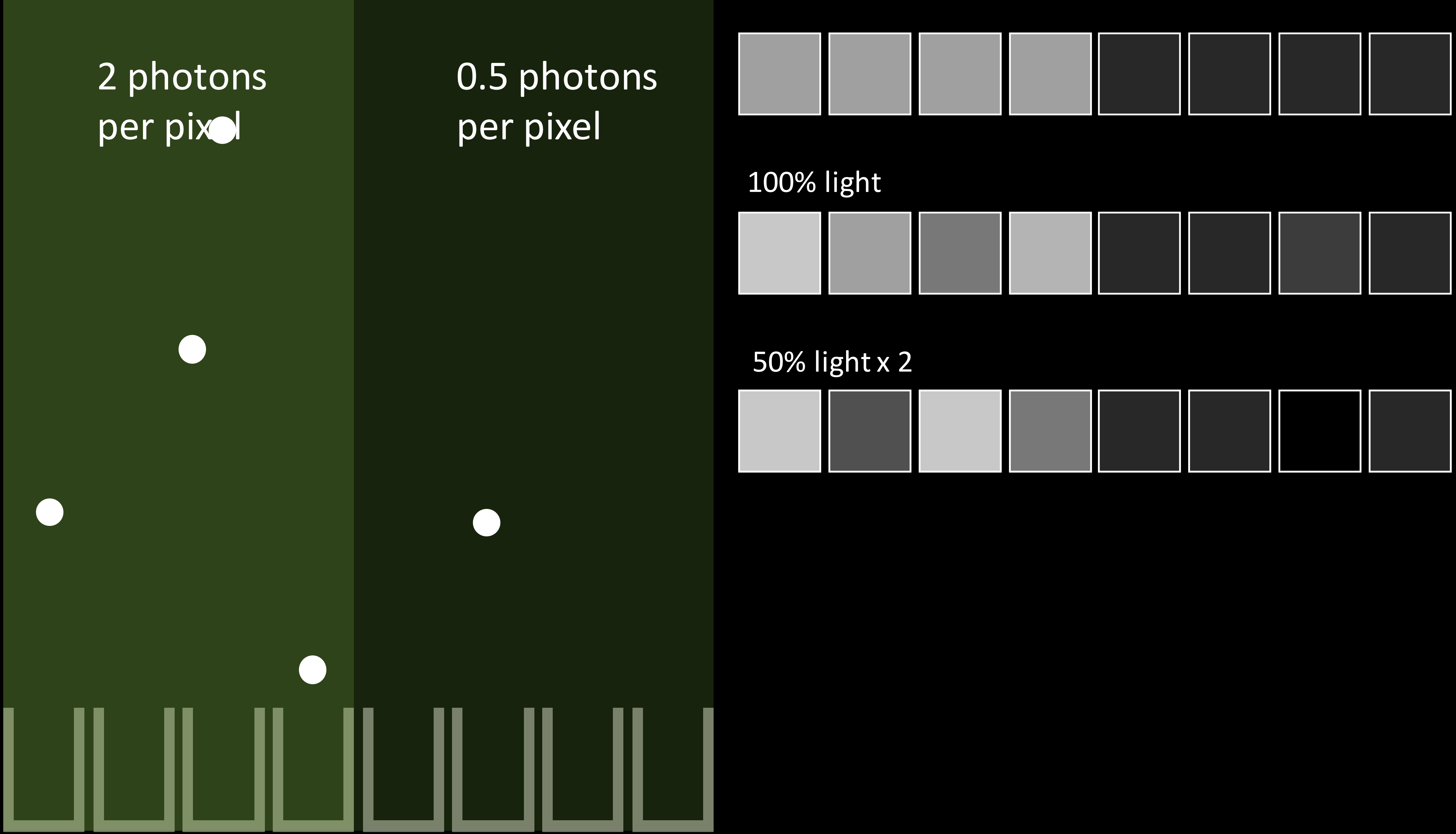
50% light



50% light

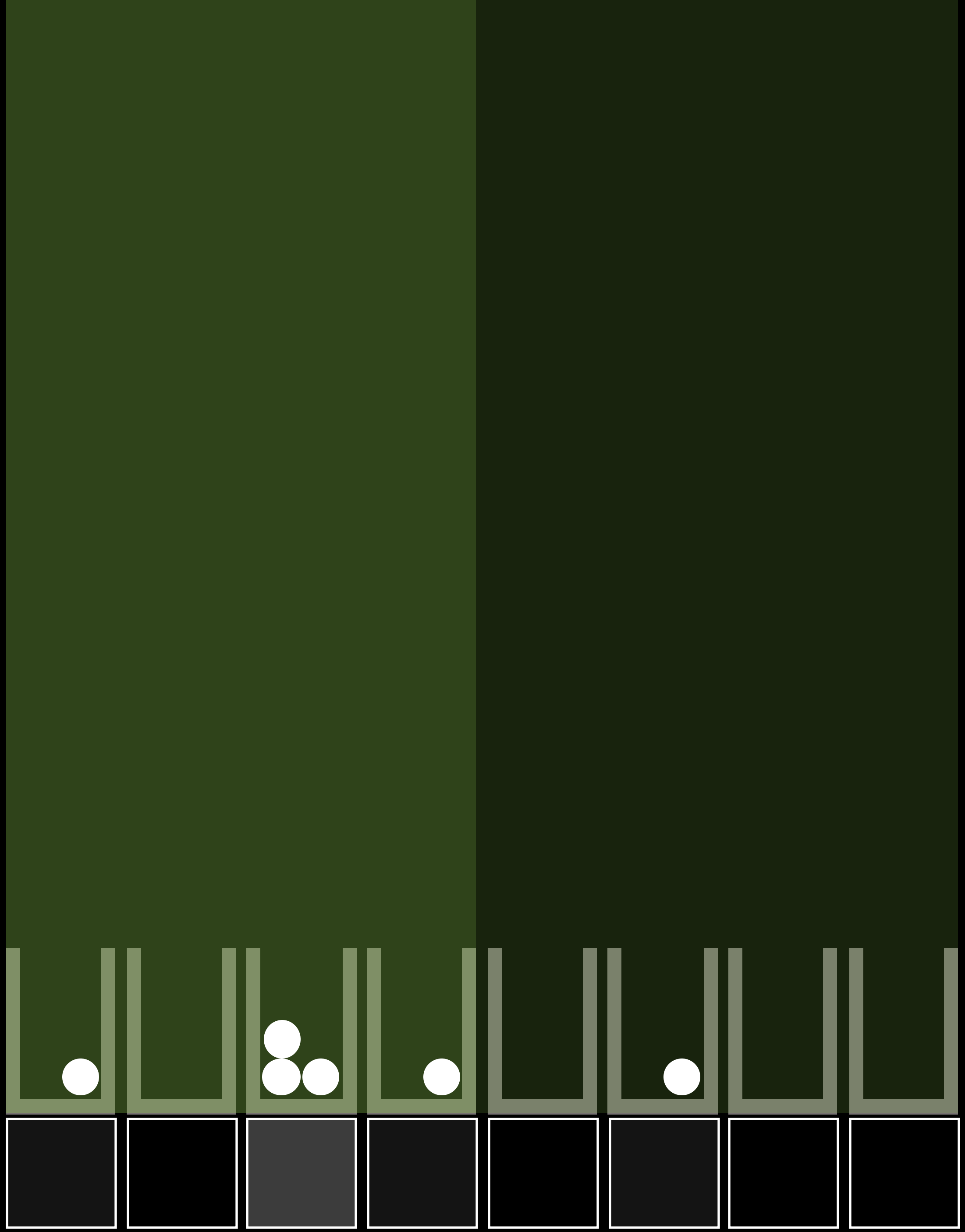


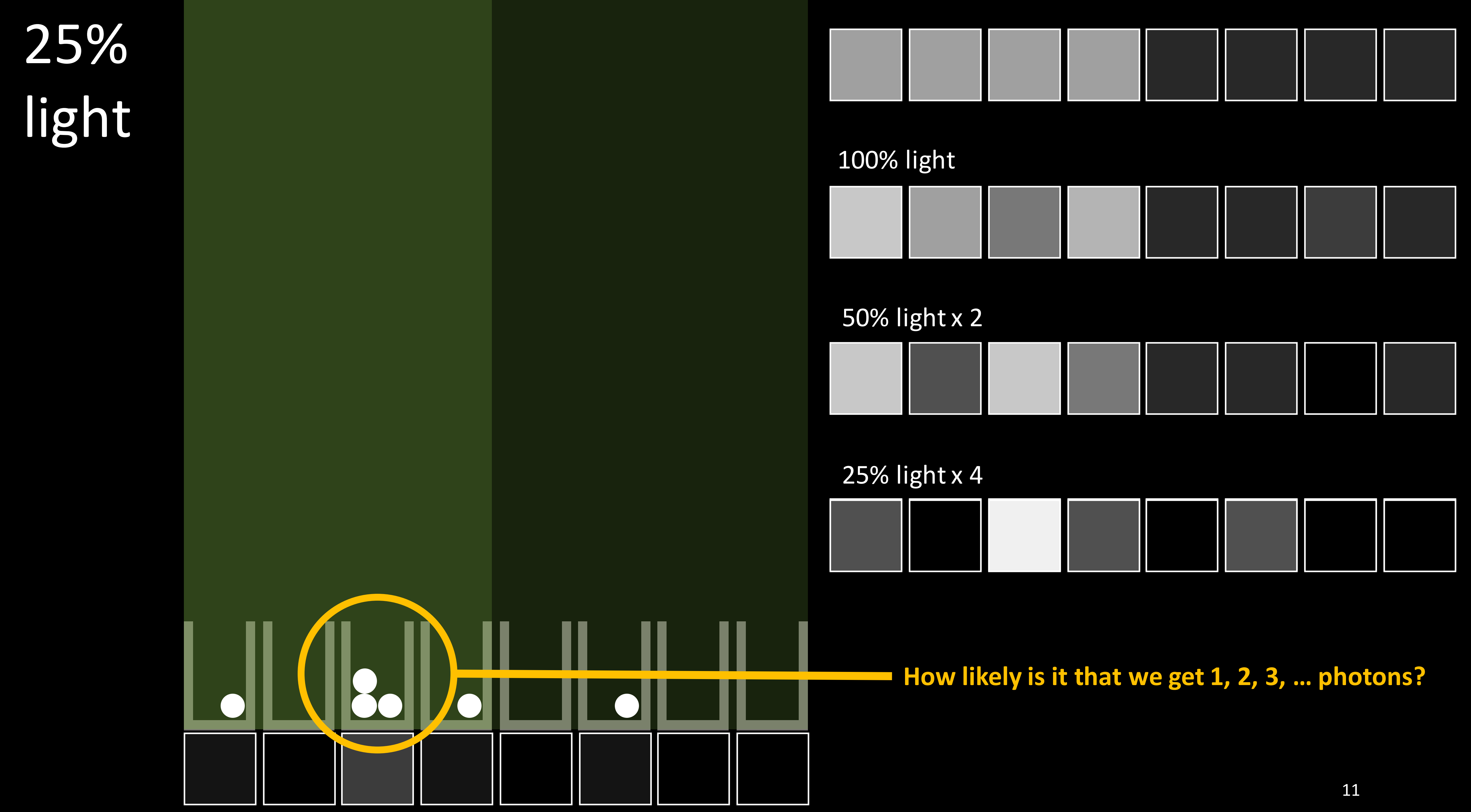
25% light



2 photons per pixel
0.5 photons per pixel
100% light
50% light x 2

25% light




25%
light
100% light
50% light x 2
25% light x 4
How likely is it that we get 1, 2, 3, … photons?

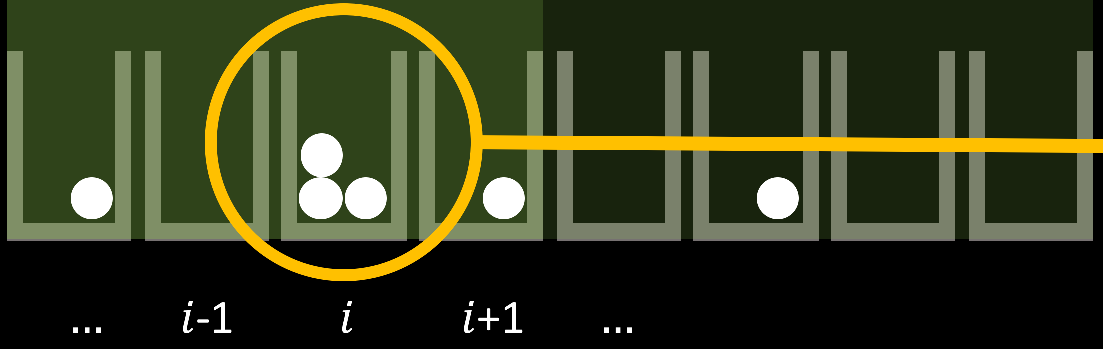
Poisson Shot Noise

Counting independent events occurring at fixed rate.

Light intensity
avg. photons per exposure
(signal)

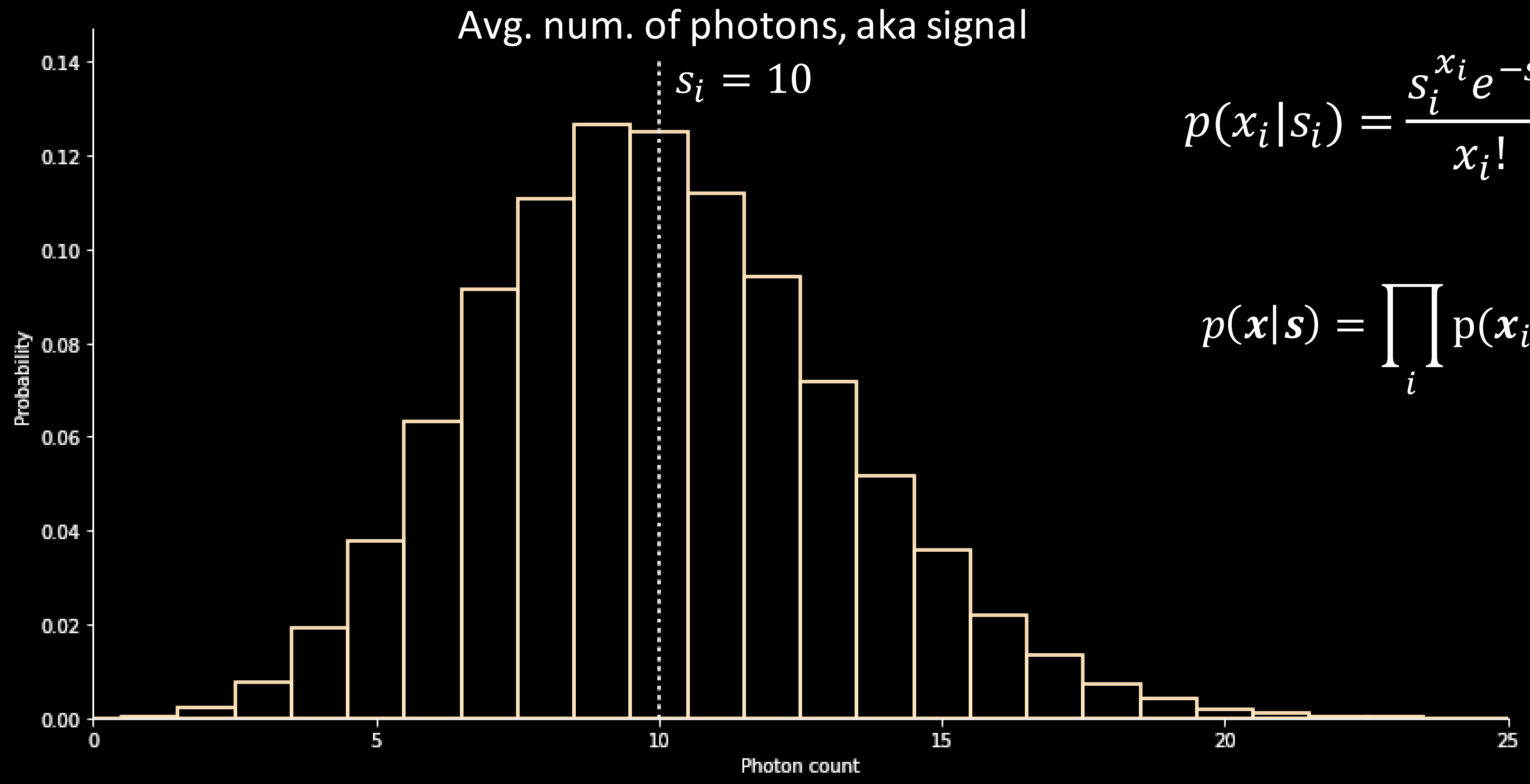
$$p(x_i|s_i) = \frac{s_i^{x_i} e^{-s_i}}{x_i!}$$

Photon count
(noisy value)



How likely is it that we get 1, 2, 3, … photons?

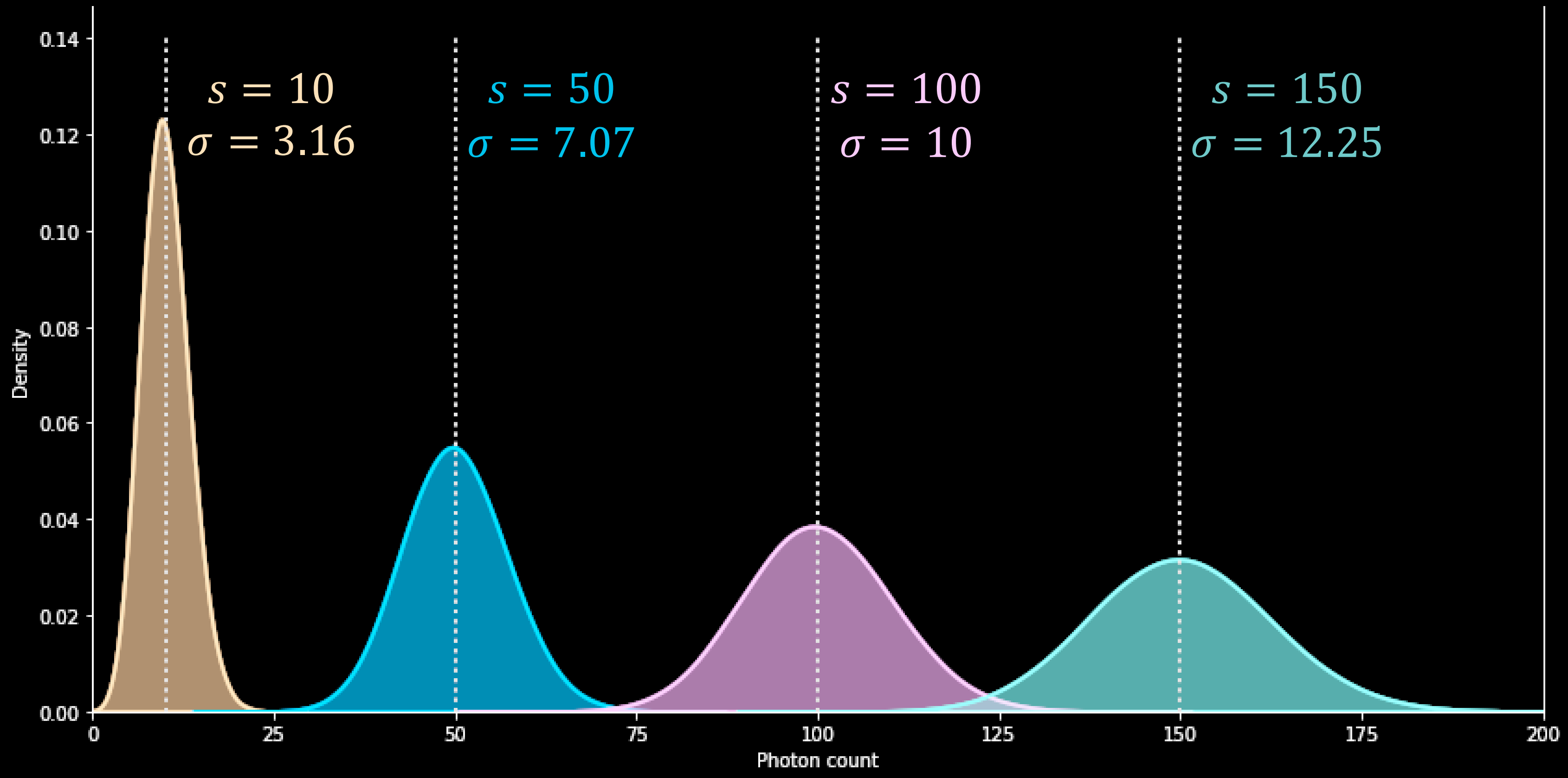
Poisson Shot Noise



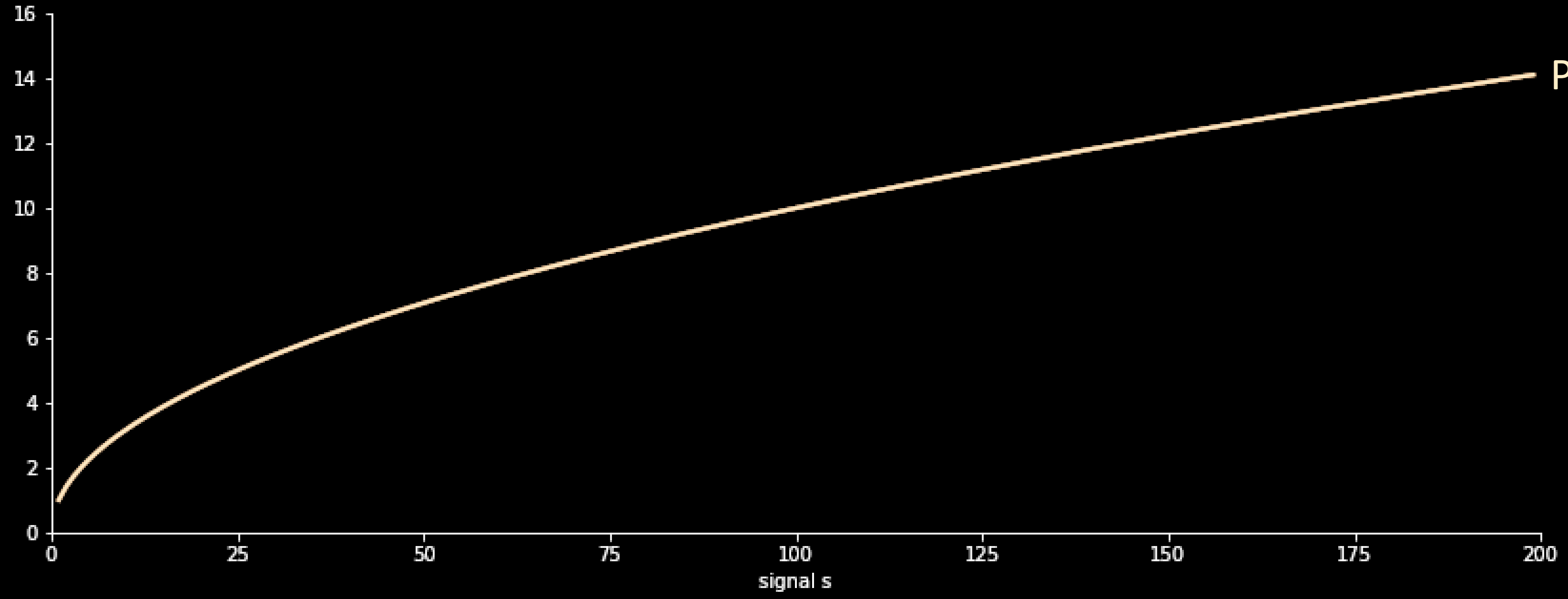
$$p(x_i|s_i) = \frac{s_i^{x_i} e^{-s_i}}{x_i!}$$

$$p(\boldsymbol{x}|\boldsymbol{s}) = \prod_i \mathrm{p}(\boldsymbol{x}_i|s_i)$$

Poisson Shot Noise

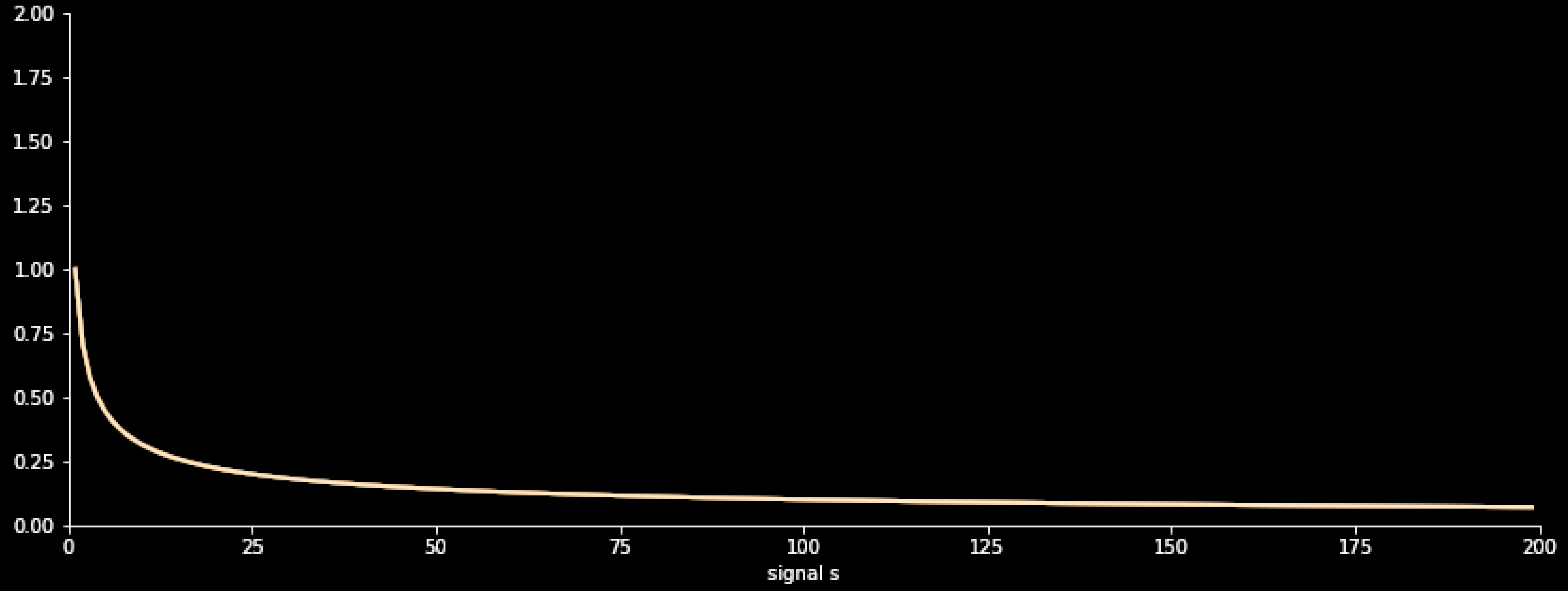


Noise std.
σ



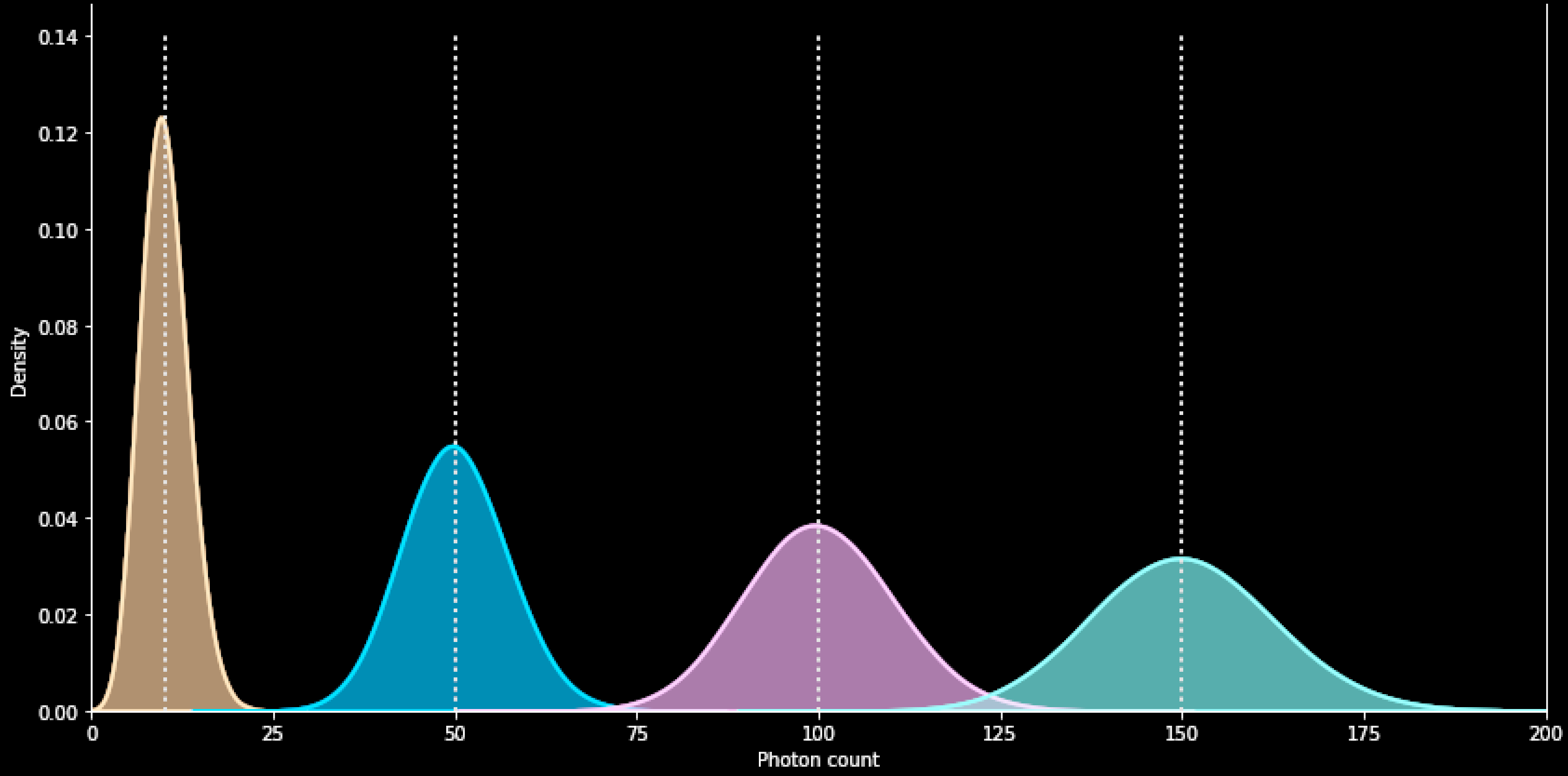
Poisson noise

Noise std.
per signal
σ/s



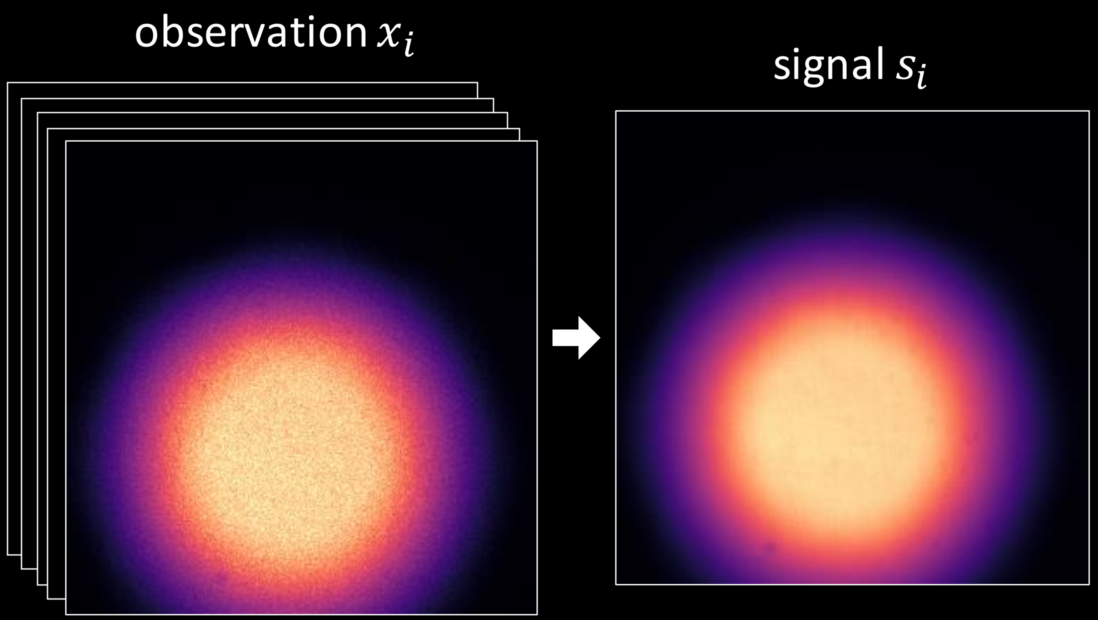
Real Noise

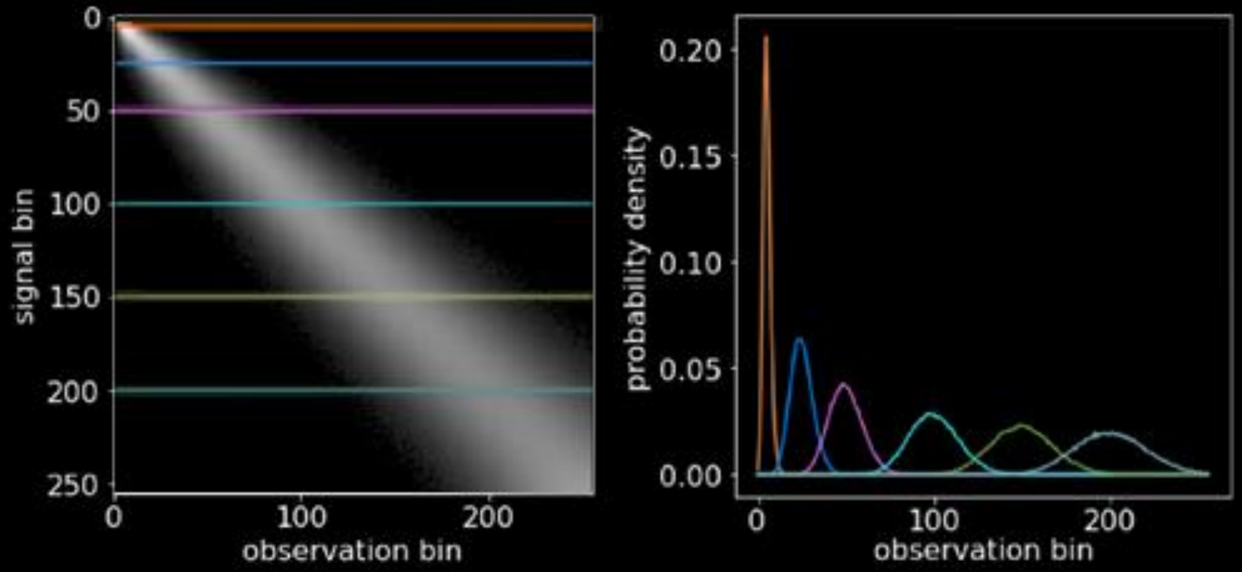
$$p(\boldsymbol{x}_i | s_i) = ?$$



- Poisson noise
- Readout noise
- Amplification
- Digitisation

Recording a Pixel Noise Model





- image static object approx. 100 times
- average result ➡ pseudo ground truth
- build 2D histogram
- build parametric model, fit Gaussians

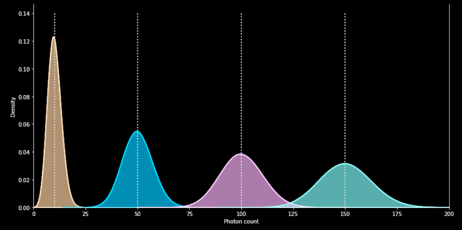
row s_i corresponds to $p(x_i|s_i)$

Noise model is …

- **a collection of distributions.**
- **property of the camera/detector.**
- **independent of sample.**

Image Noise Models

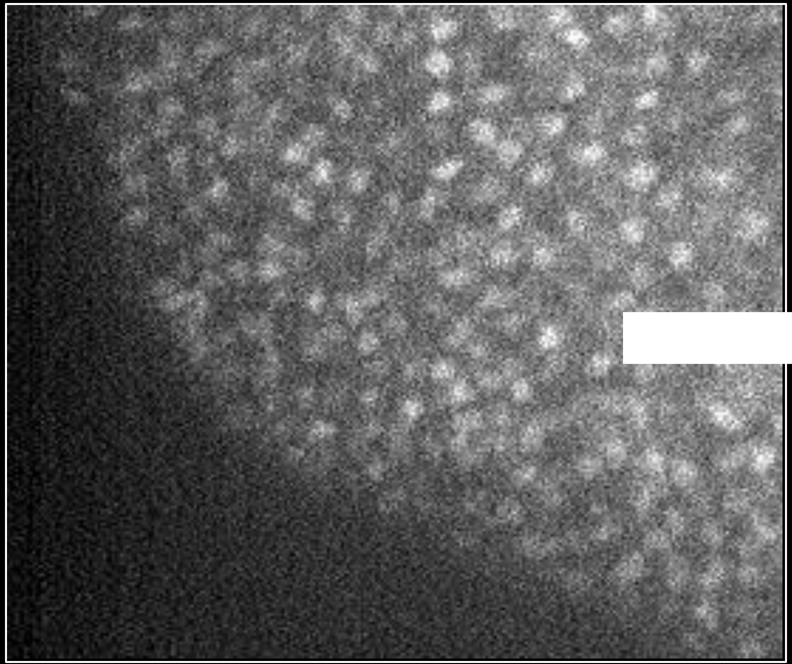
$$p(\boldsymbol{x}|\boldsymbol{s}) = \prod_{\boldsymbol{i}} p(x_i|s_i)$$

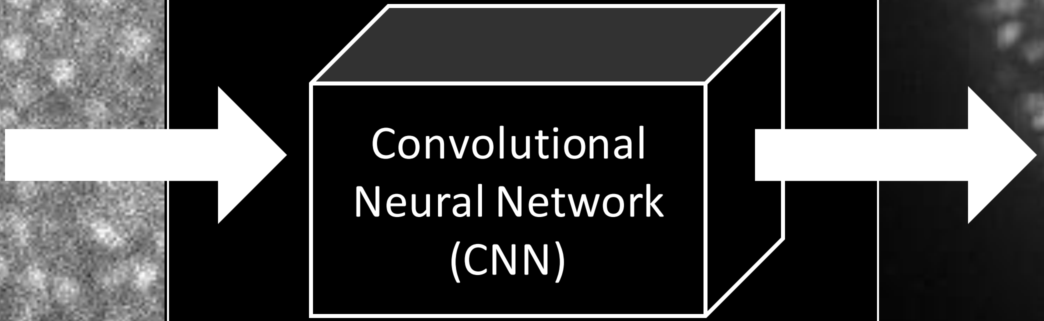


Traditional Supervised Training

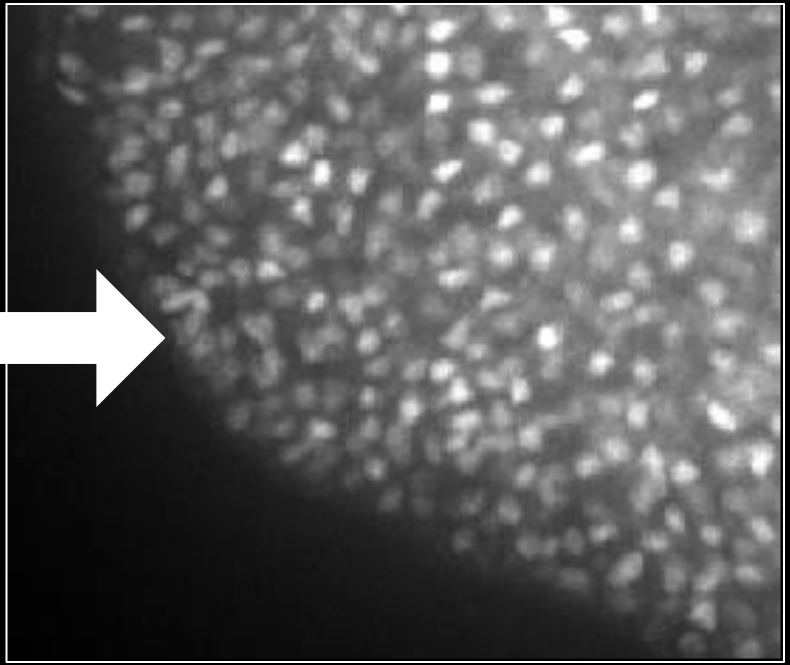
You need clean data.

Deep Learning for Denoising





- 100k – 1M parameters
- Have to be trained



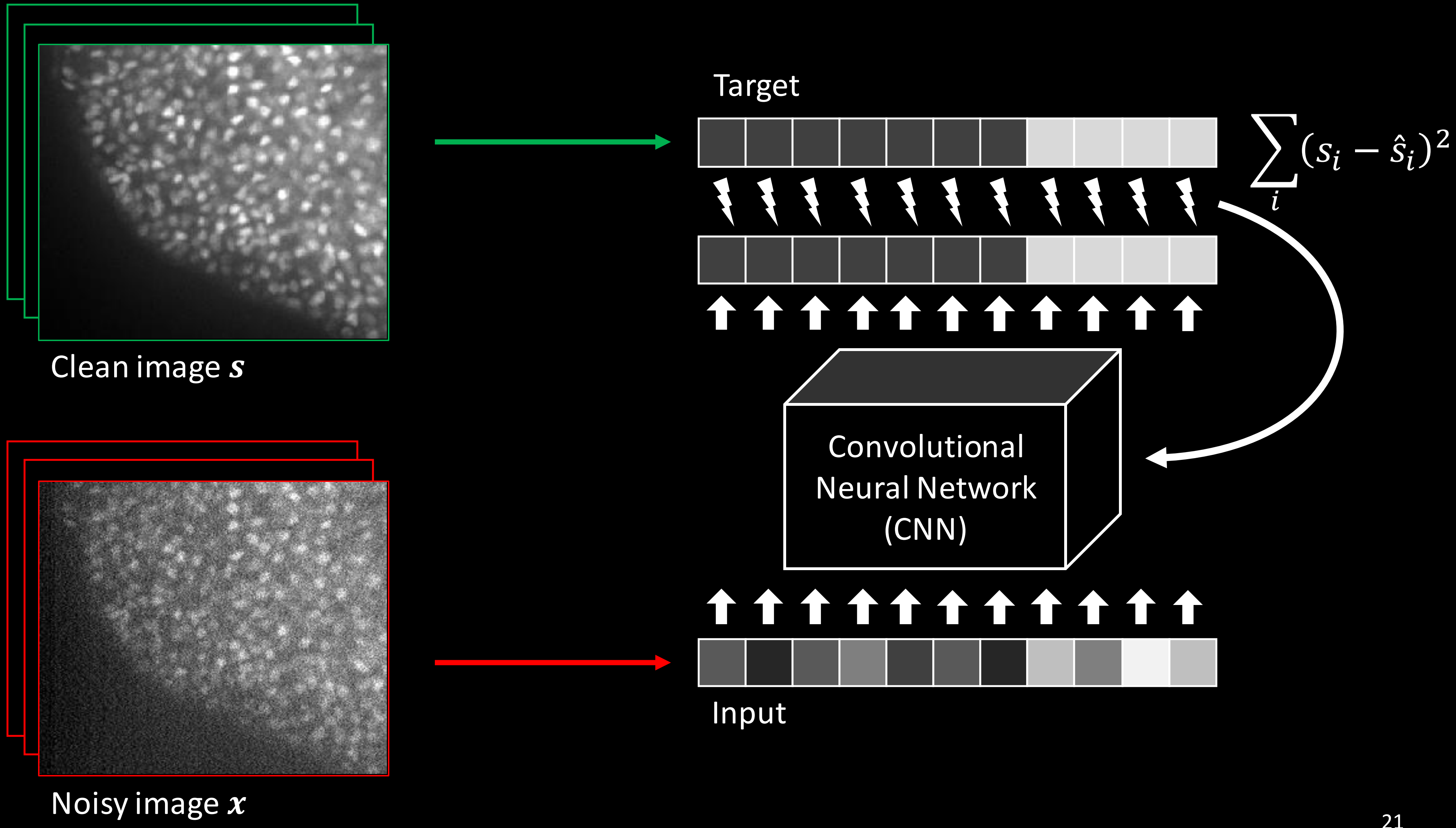
Low exposure:

- Low photo toxicity
- Low bleaching
- Noisy

High exposure:

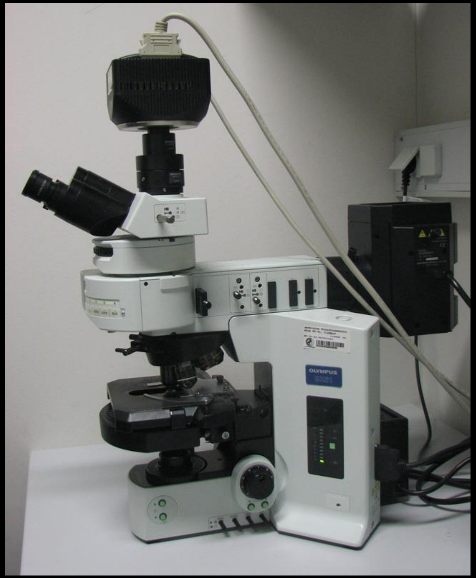
- Strong photo toxicity
- Strong bleaching
- Less noise

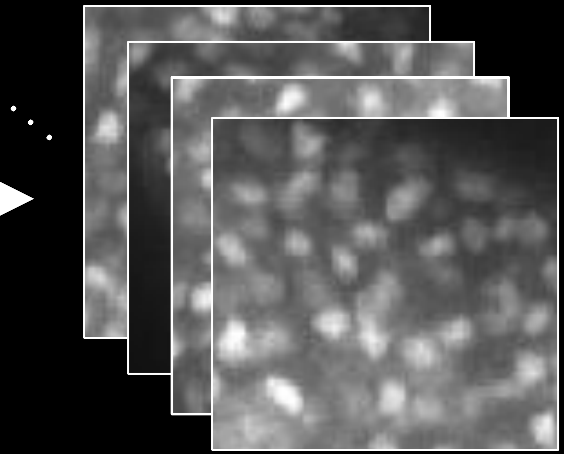
CARE – Traditional Supervised Training



What is Going On?

The Distribution of Clean Images





$p(\boldsymbol{s})$

The distr. of clean images

Signal

$\boldsymbol{s}$

Image Generation Model

$p(\boldsymbol{s})$

The distr. of clean images

$$p(\boldsymbol{x}|\boldsymbol{s}) = \prod_{\boldsymbol{i}} p(x_i|s_i)$$

The noise model

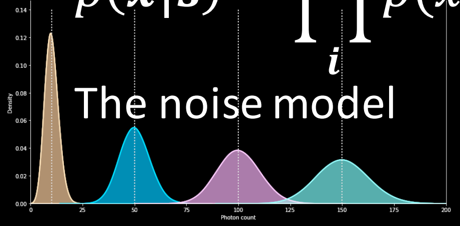


Image Generation Model

$p(\boldsymbol{s})$

The distr. of clean images

$$p(\boldsymbol{x}|\boldsymbol{s}) = \prod_{i} p(x_i|s_i)$$

The noise model

$$p(\boldsymbol{s}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s})}{\int p(\boldsymbol{x}|\boldsymbol{s}')p(\boldsymbol{s}')\, d\boldsymbol{s}'}$$

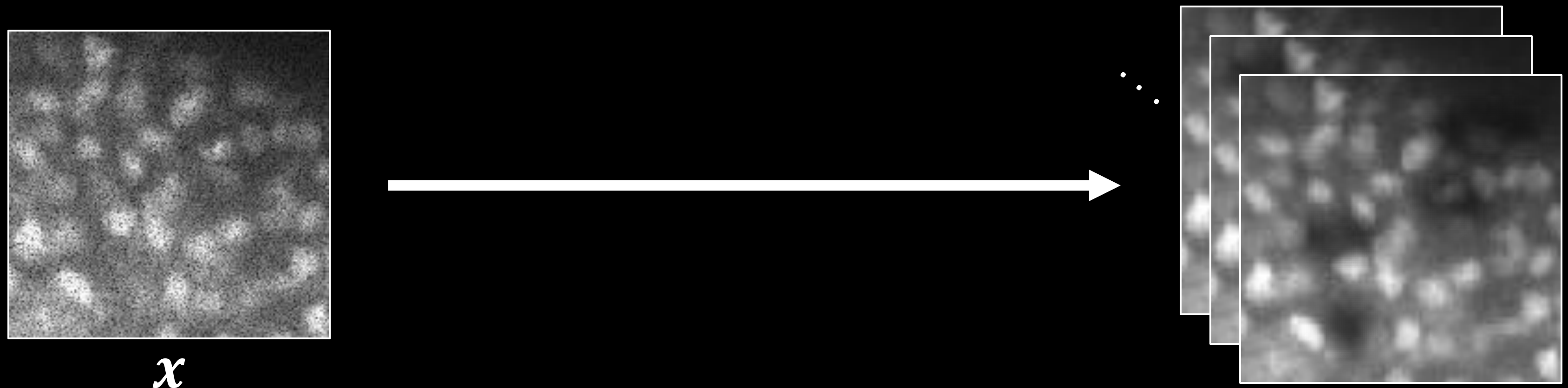
Bayes' Theorem

The Denoising Problem

Noisy Image | Signal

$p(\boldsymbol{x})$
The distr. of noisy images

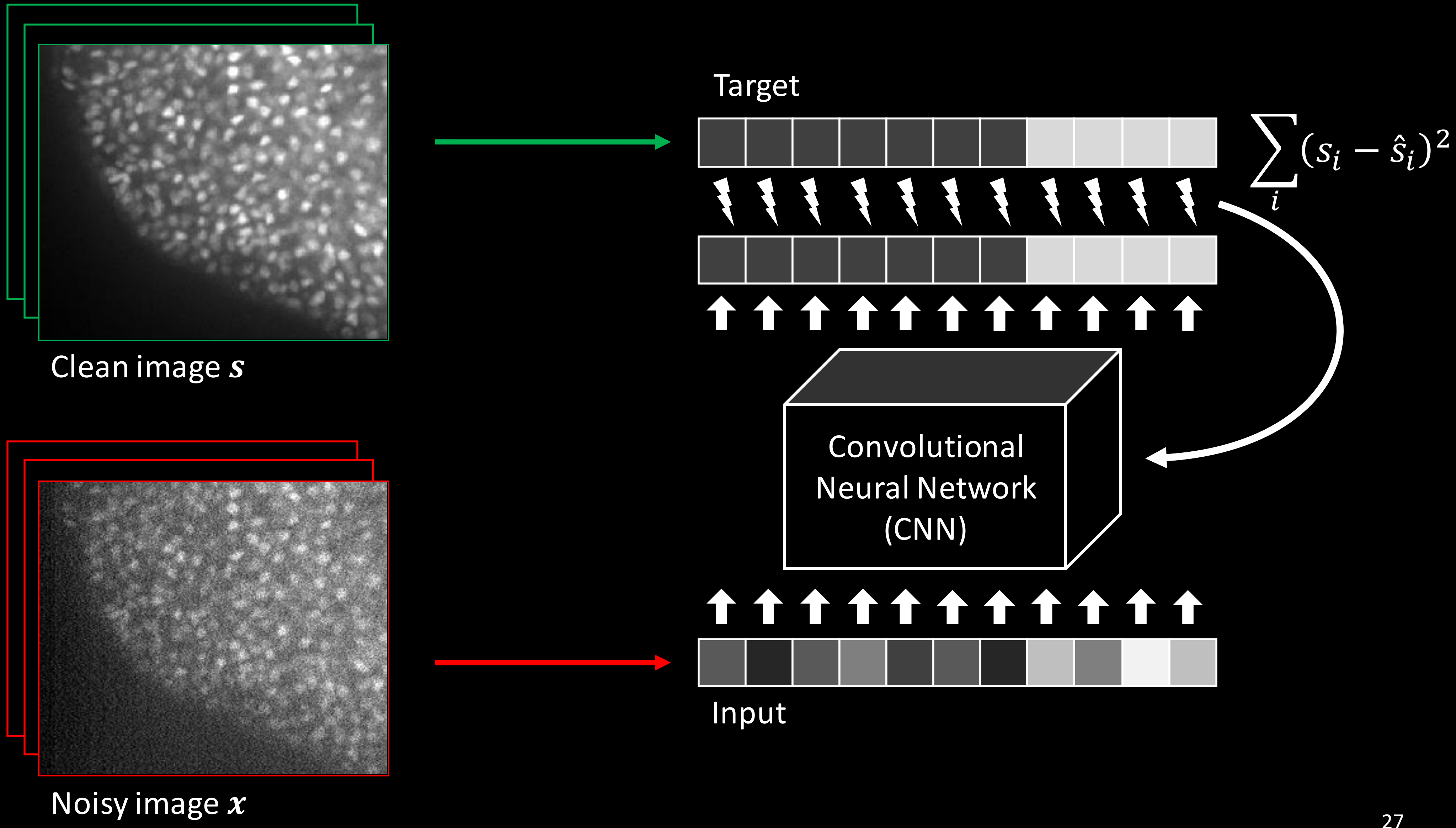
$p(\boldsymbol{s}|\boldsymbol{x})$
Posterior



$\boldsymbol{x}$

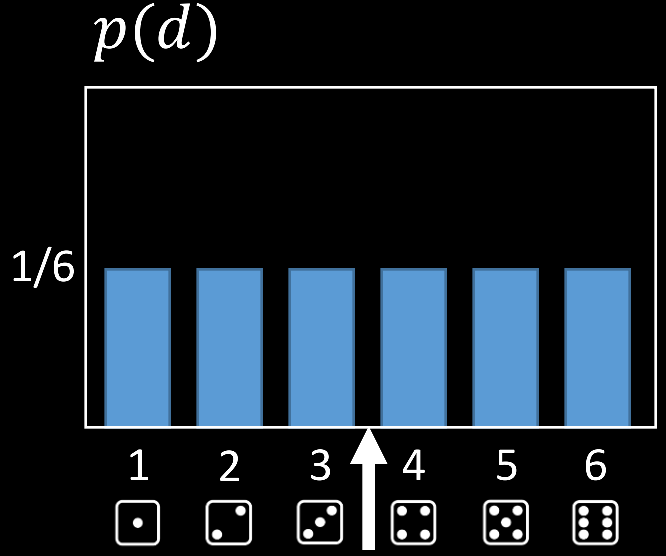
$p(\boldsymbol{s}|\boldsymbol{x})$

CARE – Traditional Supervised Training



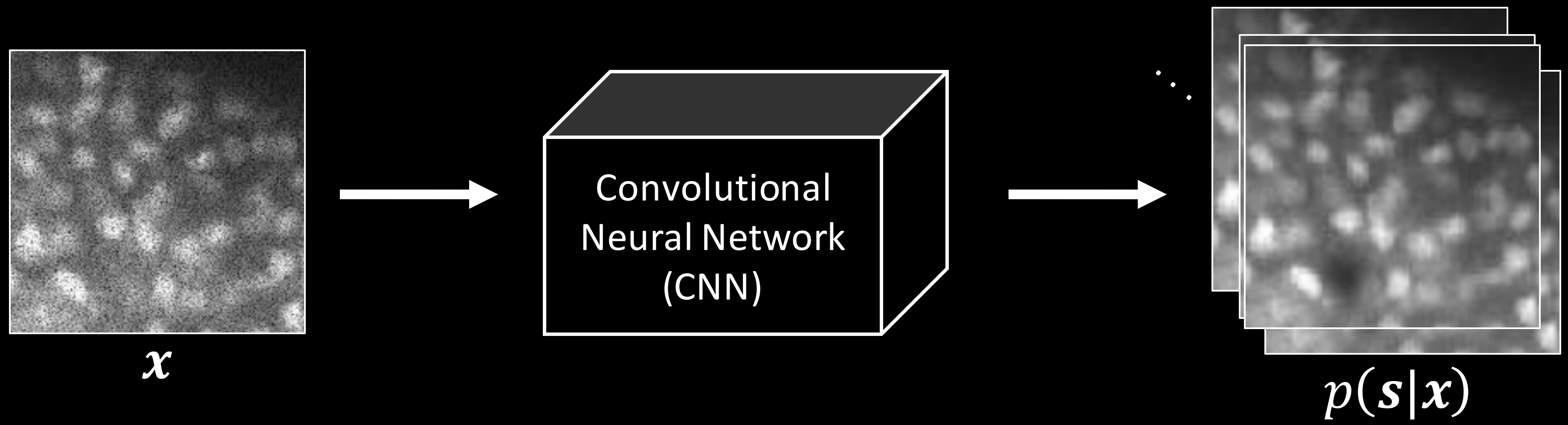
Minimising the Squared Error

Rolling a die:



Minimising $\left(d - \hat{d}\right)^2$ $\longrightarrow$ $\hat{d} \approx \mathbb{E}_{p(d)}[d]$

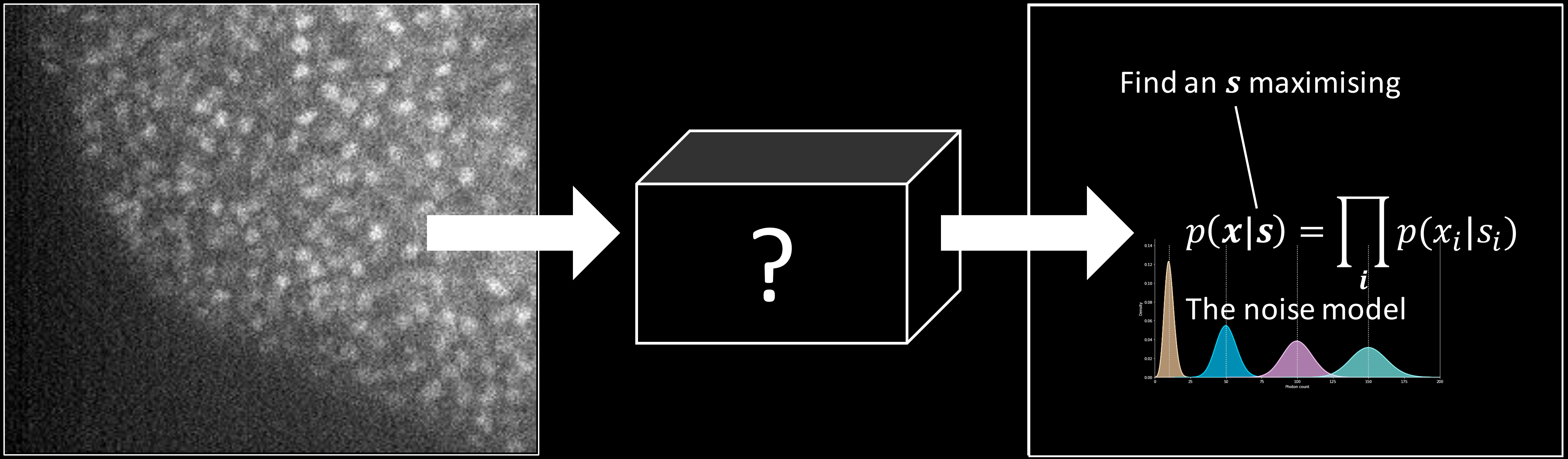
Minimising the Squared Error



Minimising $\sum_i (s_i - \hat{s}_i)^2$ → $\hat{\boldsymbol{s}} \approx \mathbb{E}_{p(\boldsymbol{s}|\boldsymbol{x})}[\boldsymbol{s}]$

$$p(\boldsymbol{s}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s})}{\int p(\boldsymbol{x}|\boldsymbol{s}')p(\boldsymbol{s}')\,d\boldsymbol{s}'}$$

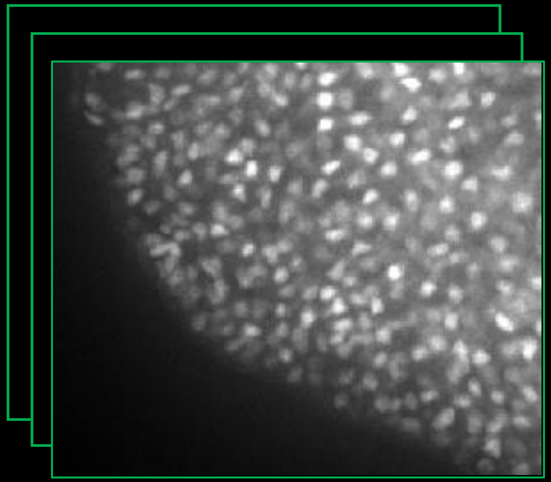
Directly Using a Noise Model for Denoising?

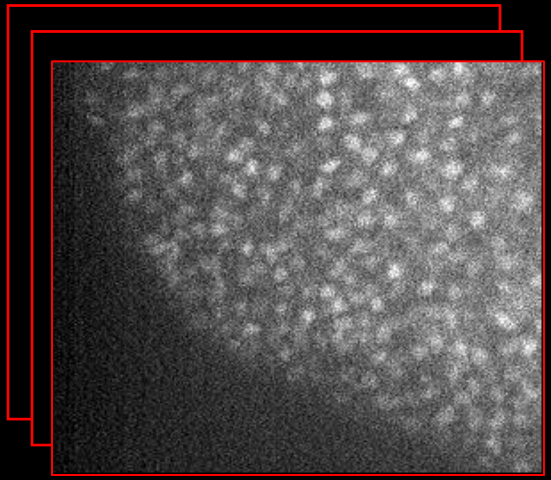


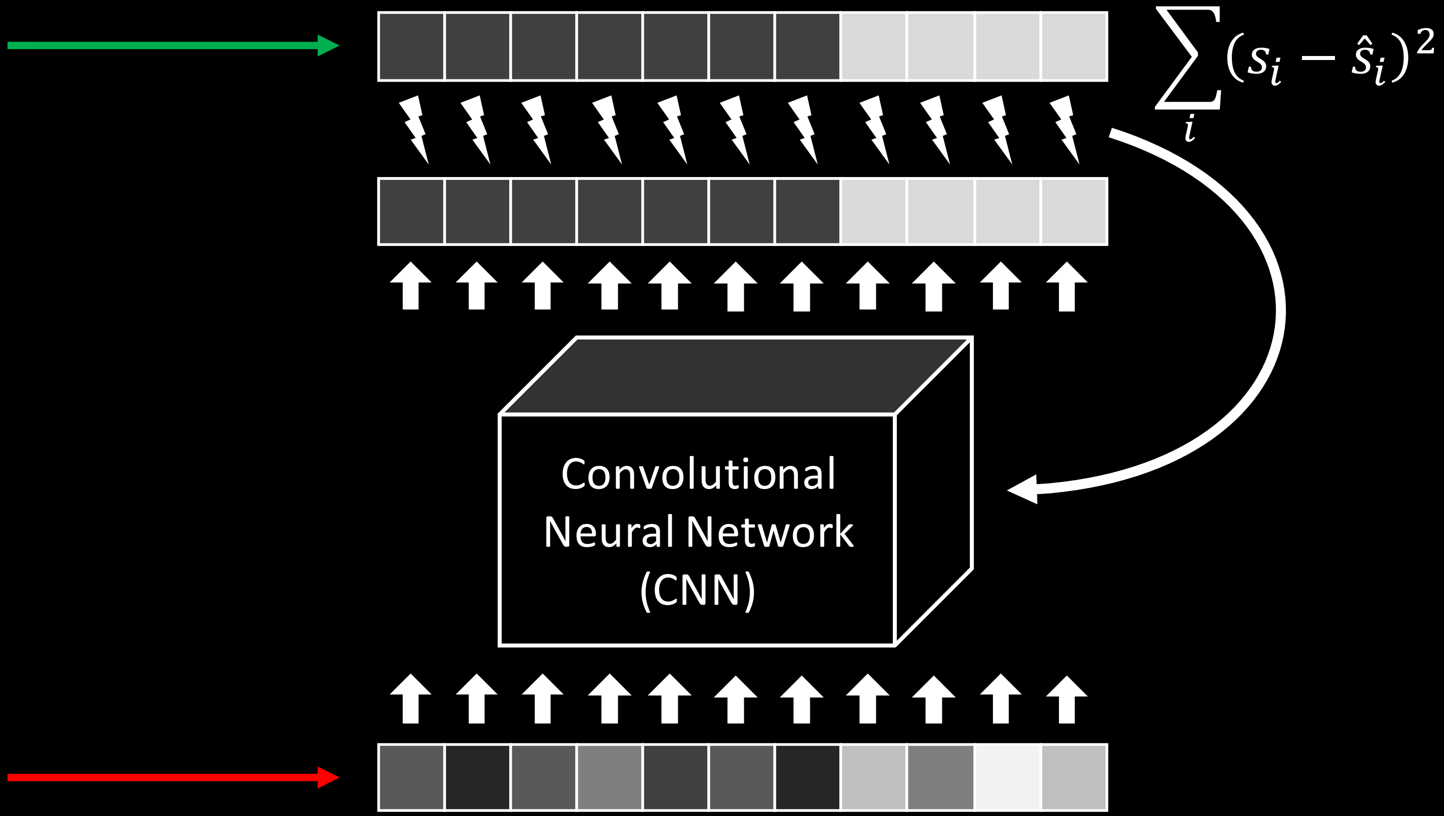
- How can we find $\boldsymbol{s}$?
- How will it compare to the supervised approach?
  - Advantages?
  - Disadvantes?

We need noise model **and** prior. $p(\boldsymbol{s}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s})}{\int p(\boldsymbol{x}|\boldsymbol{s}')p(\boldsymbol{s}')\,d\boldsymbol{s}'}$

CARE – Traditional Supervised Training


Clean image $\boldsymbol{s}$


Noisy image $\boldsymbol{x}$

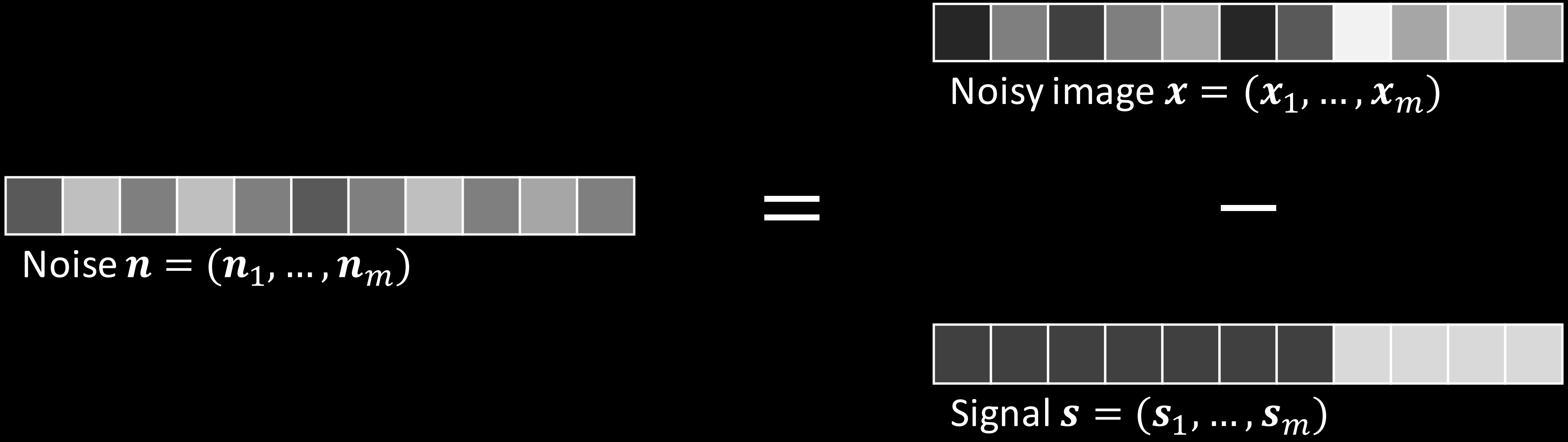


Noise2Noise

You only need noisy data!

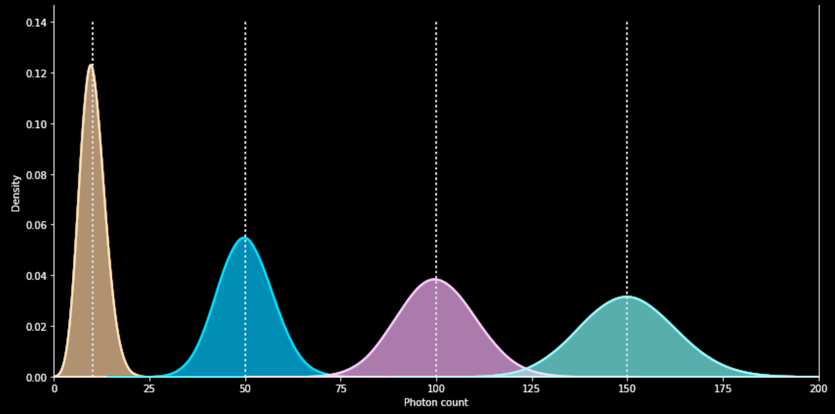
Lehtinen et al. 2018

Noise2Noise



Noise2Noise

Noisy image $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_m)$



=

Signal $\boldsymbol{s} = (\boldsymbol{s}_1, \dots, \boldsymbol{s}_m)$

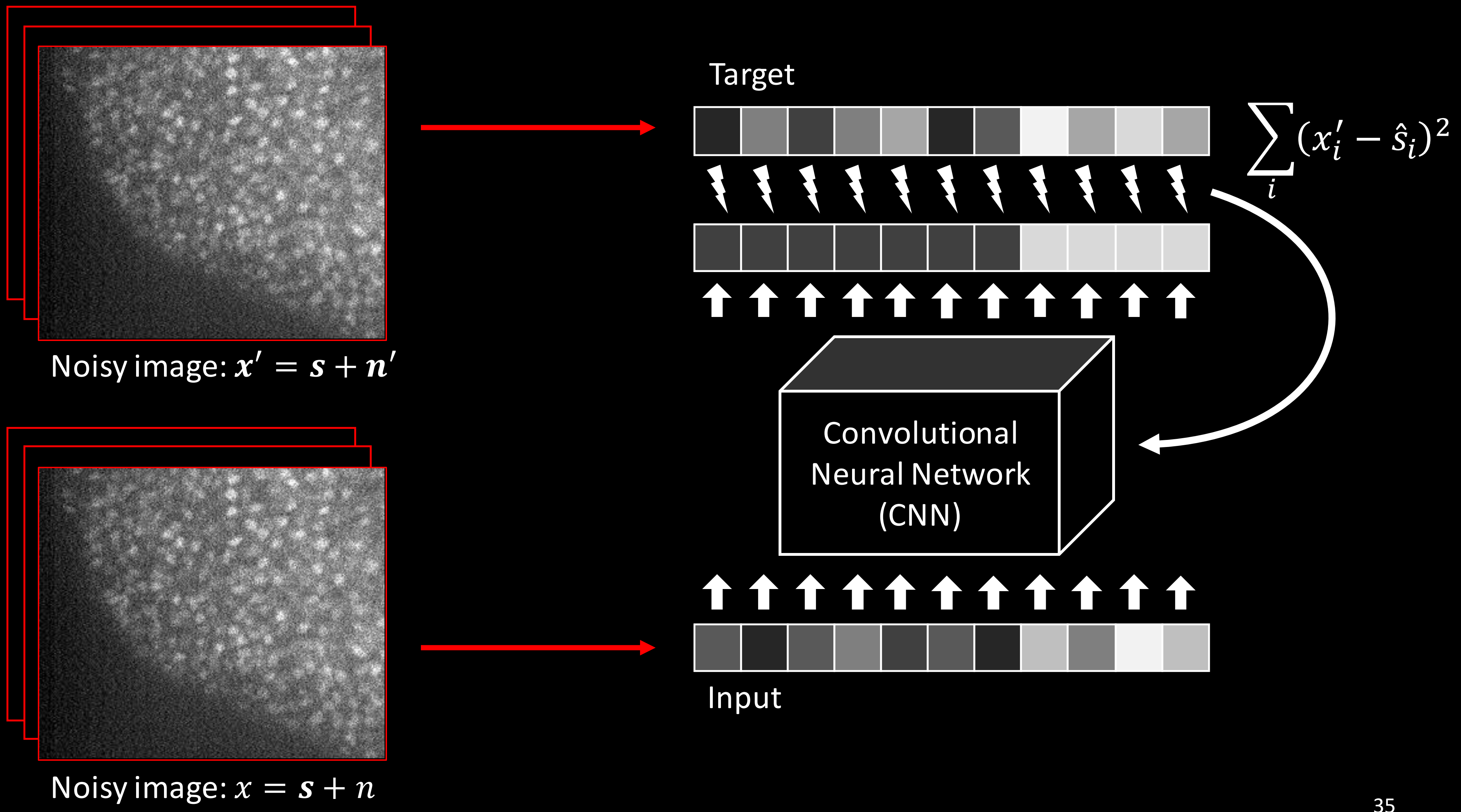
+

Noise $\boldsymbol{n} = (\boldsymbol{n}_1, \dots, \boldsymbol{n}_m)$

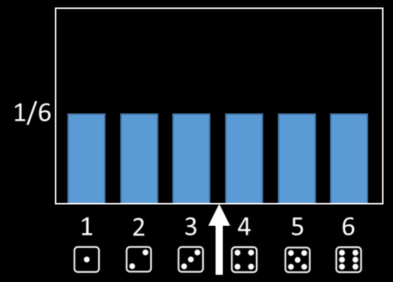
$$\mathbb{E}_{p(n_i|s_i)}[n_i] = 0$$

Zero centred noise

Noise2Noise training



Noise2Noise - Why does it work?



Supervised:

Minimising $\sum_i (s_i - \hat{s}_i)^2 \longrightarrow \hat{s}_i \approx \mathbb{E}_{p(s_i|\boldsymbol{x})}[s_i]$

N2N:

Minimising $\sum_i (x_i' - \hat{s}_i)^2 \longrightarrow \hat{s}_i \approx \mathbb{E}_{p(x_i'|\boldsymbol{x})}[x_i']$

$= \mathbb{E}_{p(s_i|\boldsymbol{x})}[s_i]$

Noise2Noise - Why does it work?

$$\mathbb{E}_{p(x_i' \mid \boldsymbol{x})}[x_i']$$

$$= \int p(x_i'|\boldsymbol{x}) x_i' \, dx_i'$$

Marginalisation

$$= \int x_i' \int p(x_i', s_i|\boldsymbol{x}) \; ds_i \, dx_i'$$

Product rule

$$= \int x_i' \int p(x_i'|s_i, \boldsymbol{x}) p(s_i|\boldsymbol{x}) \; ds_i \, dx_i'$$

$$= \int \int x_i' \, p(x_i'|s_i, \boldsymbol{x}) p(s_i|\boldsymbol{x}) \; ds_i \, dx_i'$$

$$= \int \int x_i' \, p(x_i'|s_i, \boldsymbol{x}) p(s_i|\boldsymbol{x}) \; dx_i' \, ds_i$$

$$= \int p(s_i|\boldsymbol{x}) \int x_i' \, p(x_i'|s_i, \boldsymbol{x}) \; dx_i' \, ds_i$$

$$= \int p(s_i|\boldsymbol{x}) \int x_i' \, p(x_i'|s_i, \boldsymbol{x}) \; dx_i' \, ds_i$$

Cond. independence

$$= \int p(s_i|\boldsymbol{x}) \int x_i' \, p(x_i'|s_i \,) \; dx_i' \, ds_i$$

Expected value

$$= \int p(s_i|\boldsymbol{x}) \, \mathbb{E}_{p(x_i'|s_i\,)}[x_i'] \; ds_i$$

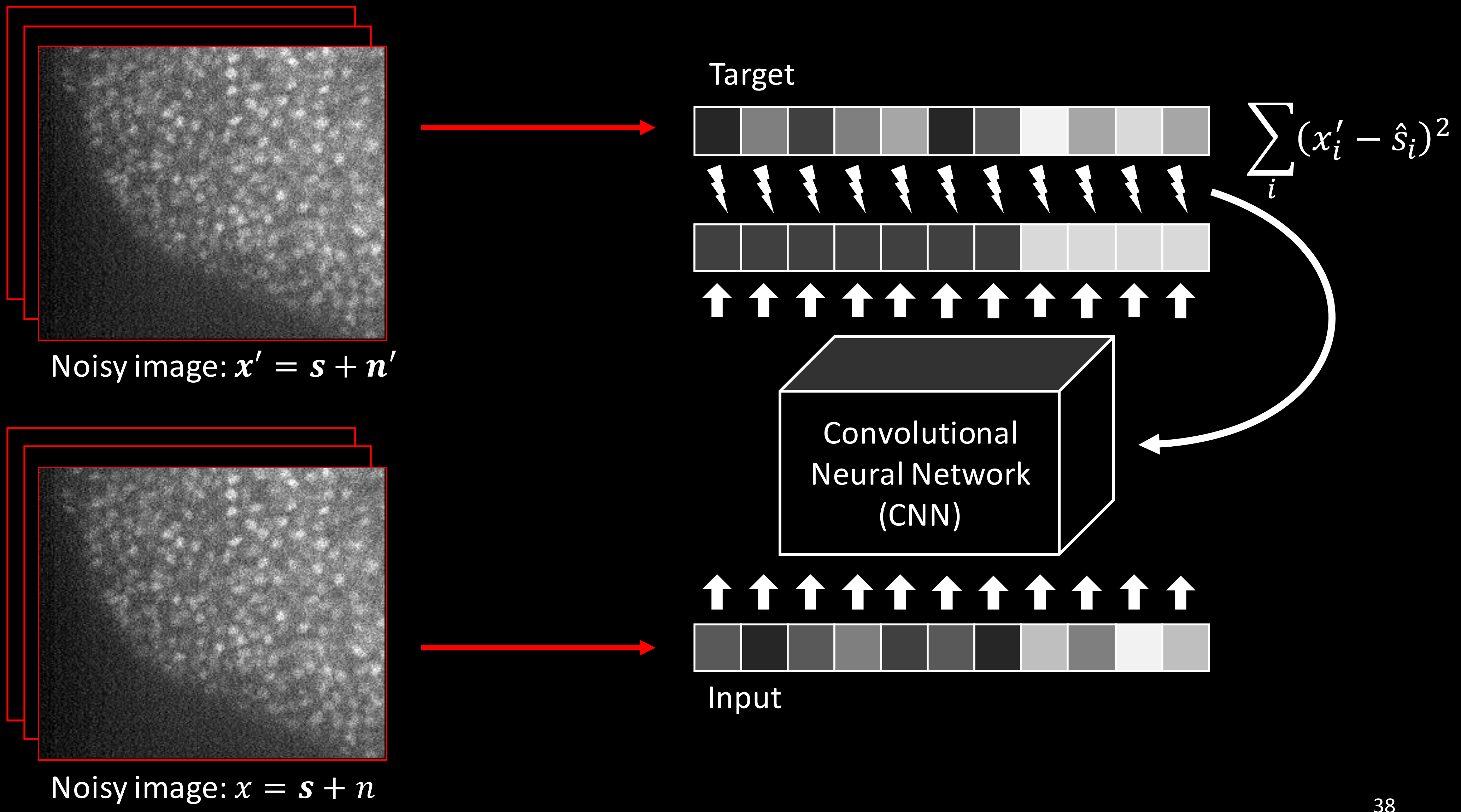
Zero centred noise

$$= \int p(s_i|\boldsymbol{x}) s_i \; ds_i$$

Expected value

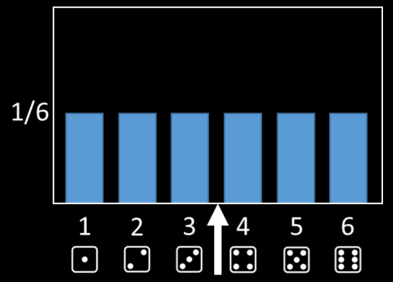
$$= \mathbb{E}_{p(s_i|\boldsymbol{x})}[s_i]$$

Noise2Noise training



Summary

- Traditional supervised training:
  - Tries to map noisy image to clean image.
  - Impossible: map to expected value of clean image.
  - Downside: requires clean images during training.



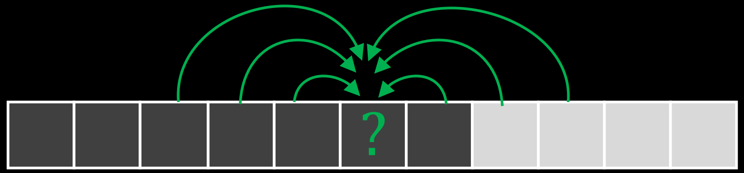
- Noise2Noise training:
  - Requires no clean data.
  - Tries to map noisy image to noisy image.
  - Impossible: also map to expected value of clean image.
  - Downside: Still requires image pairs.

Self-Supervised Denoising: Noise2Void

You only need individual noisy images!

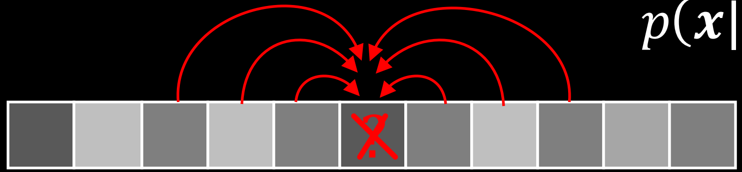
Krull, Buchholz, and Jug 2019

Noise2Void – Assumptions



Signal $\boldsymbol{s} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_m)$

$+$

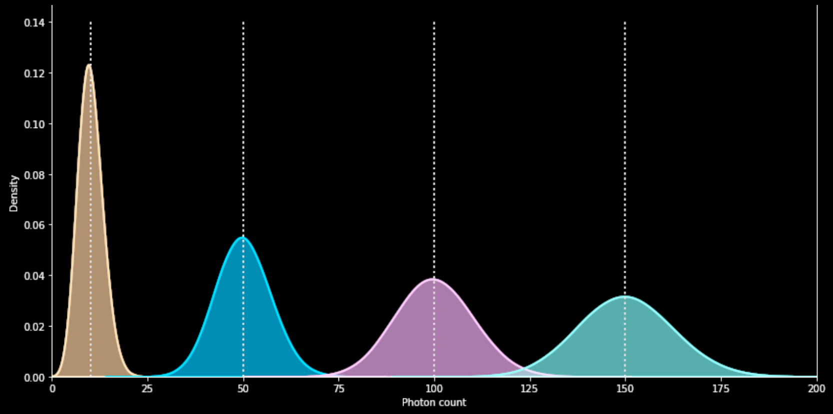


Noise $\boldsymbol{n} = (\boldsymbol{n}_1, \ldots, \boldsymbol{n}_m)$

$$p(\boldsymbol{x}|\boldsymbol{s}) = \prod_{\boldsymbol{i}} p(x_i|s_i)$$

$=$

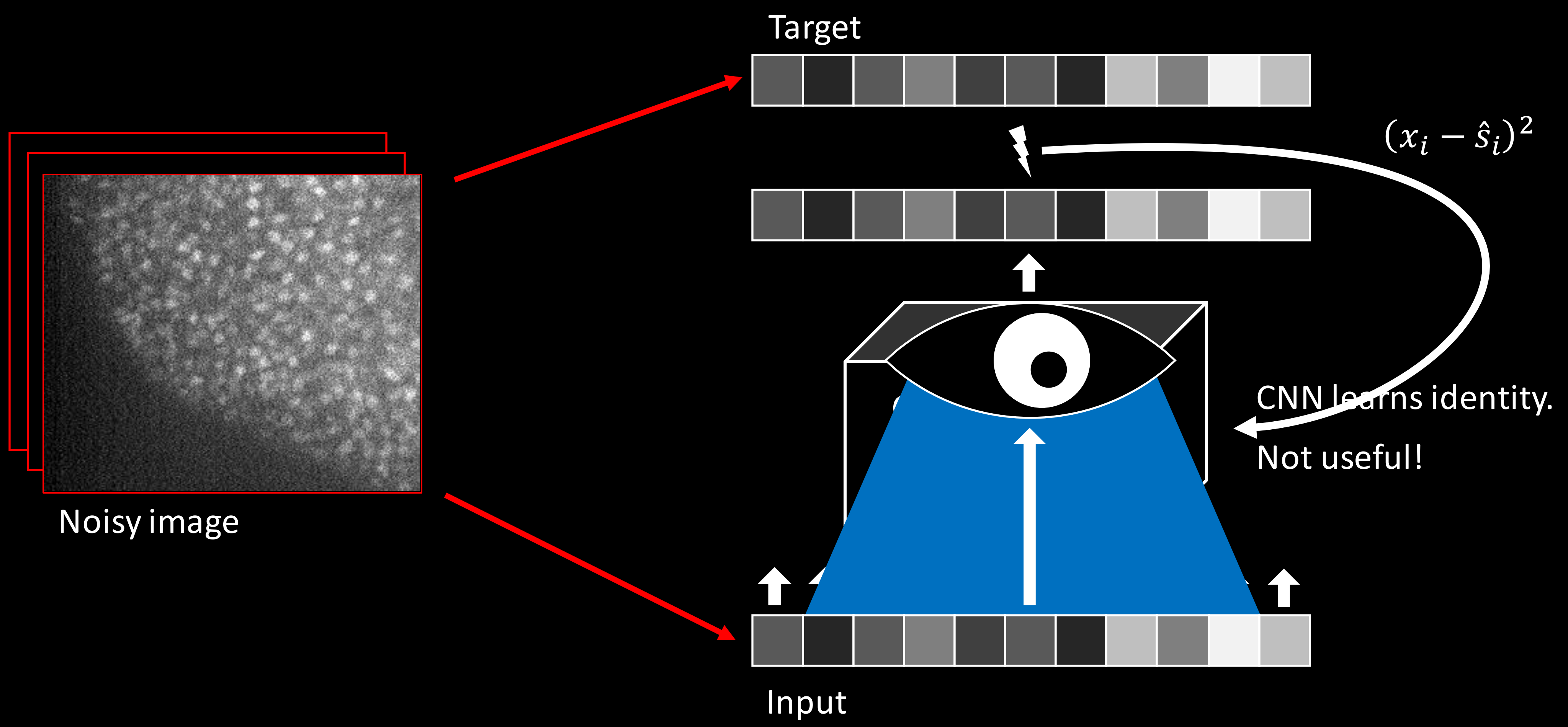
Noisy image ($\boldsymbol{x}$) aka *observation*



$$\mathbb{E}_{p(n_i|s_i)}[n_i] = 0$$

Zero centred noise

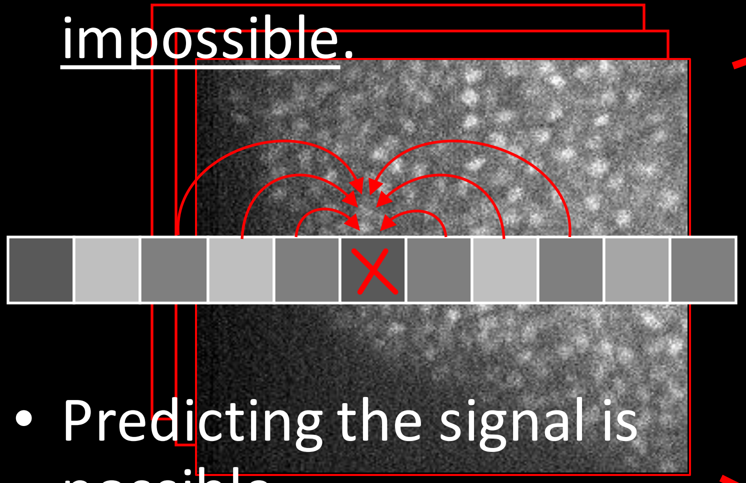
Noise2Void

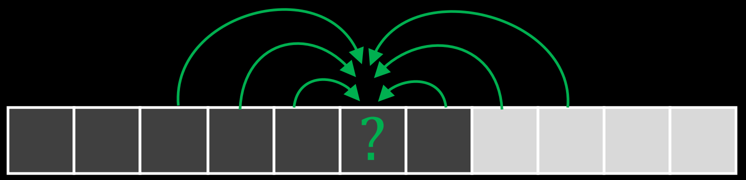

Target
$(x_i - \hat{s}_i)^2$
CNN learns identity.
Not useful!
Noisy image
Input

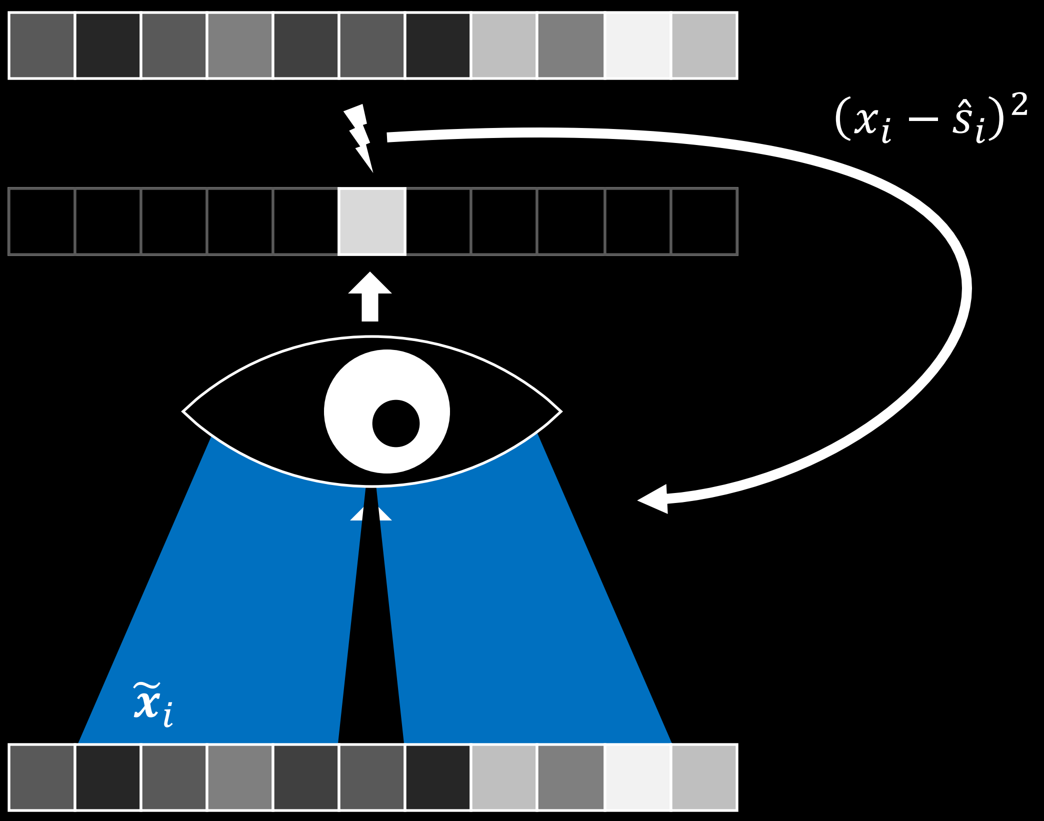
Noise2Void - Blind Spot Network

Why does it work?

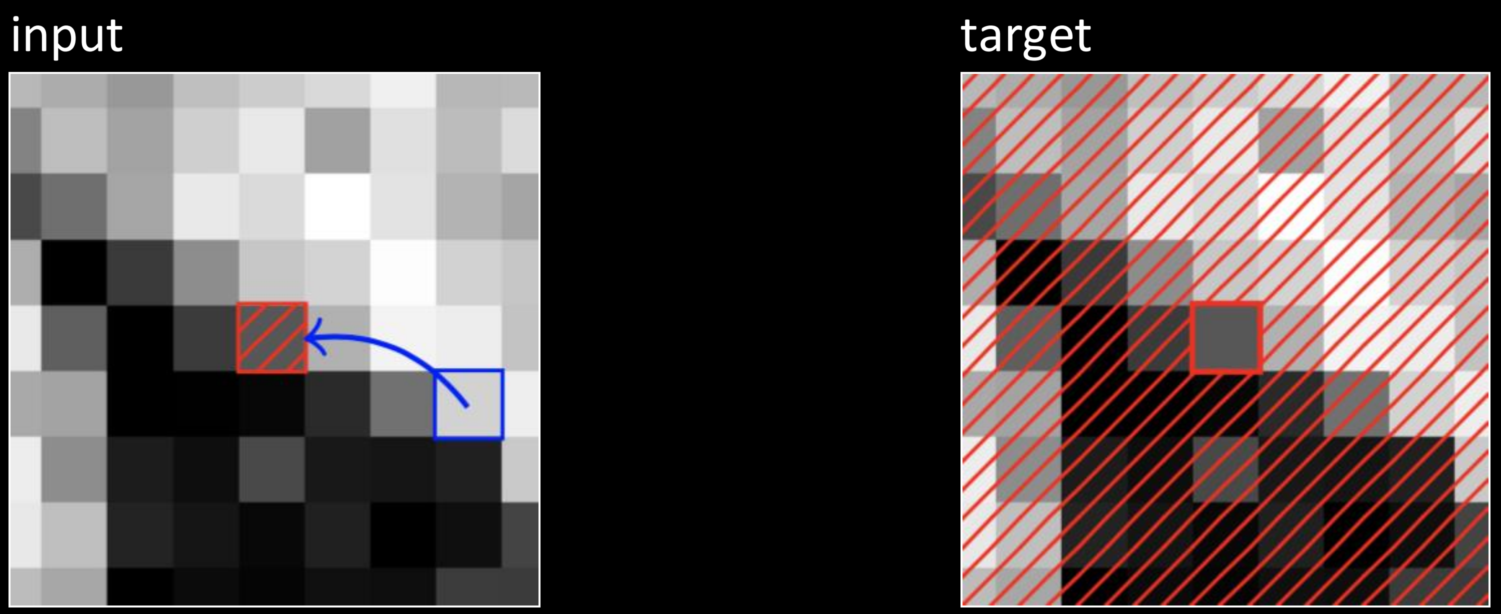
- Predicting the noise is impossible.
- Predicting the signal is possible.



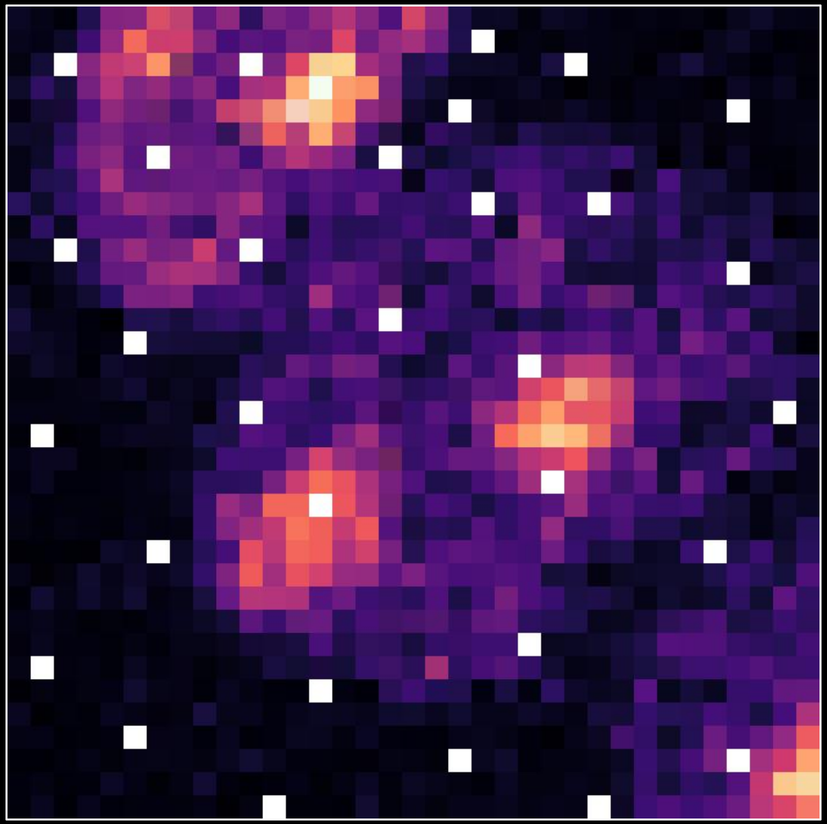




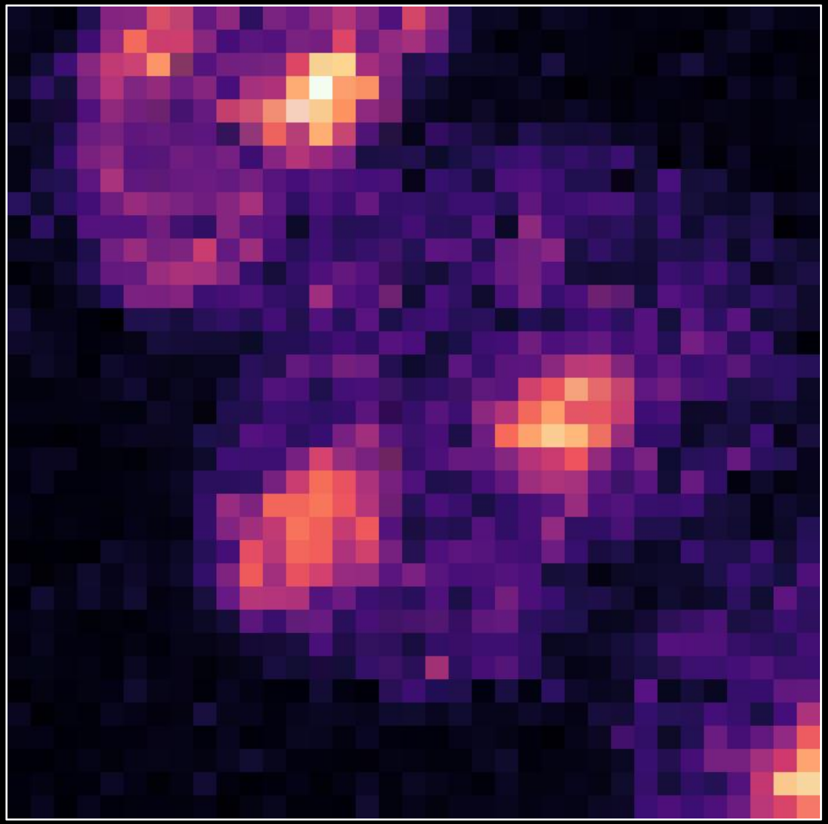
Noise2Void - Blind Spot Implementation



Noise2Void - Blind Spot Implementation

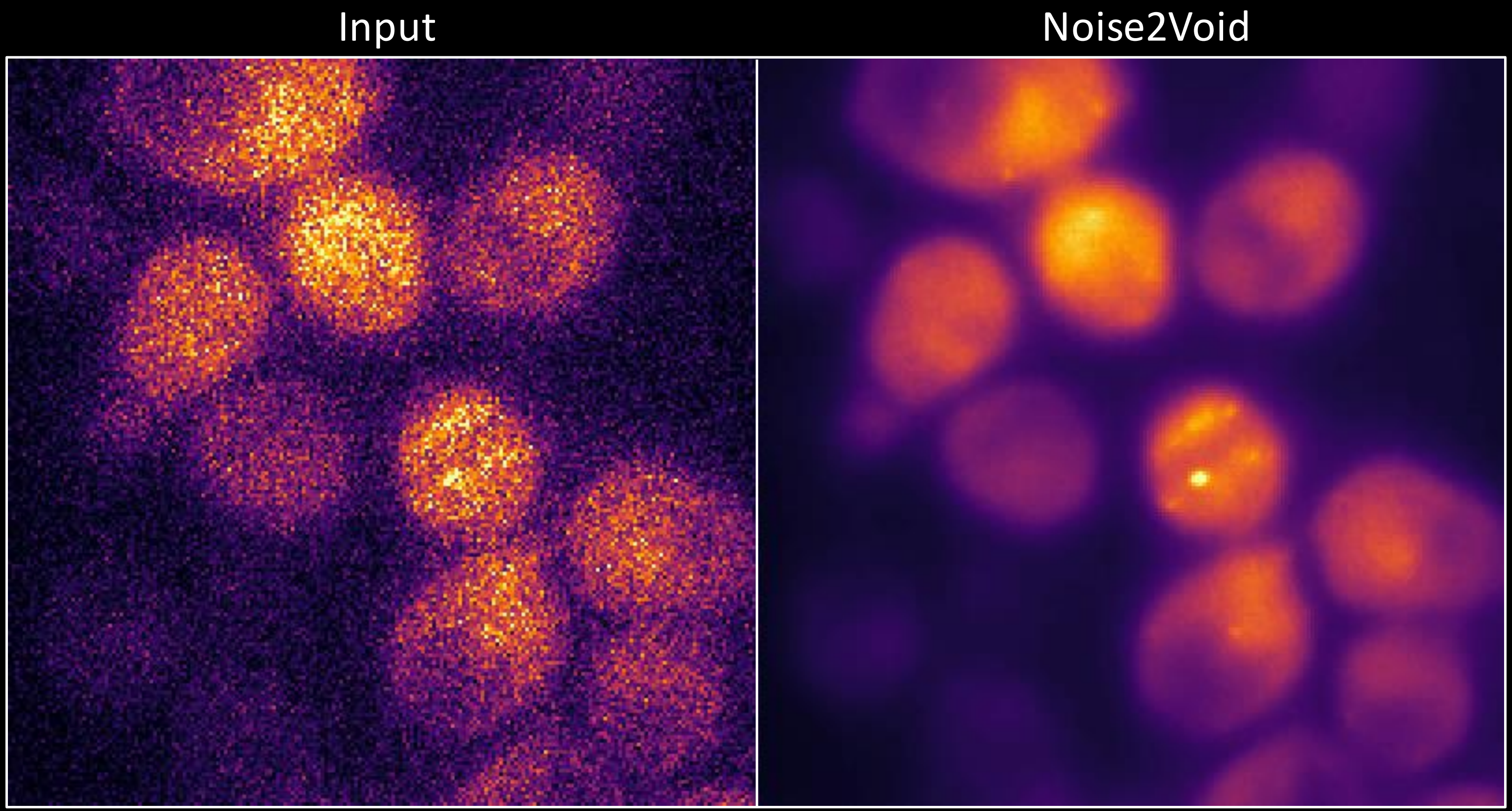


Input



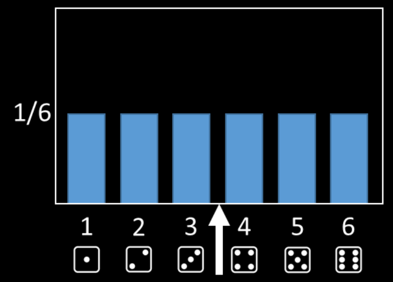
Target

Noise2Void - Results



Data by Stephanie Heinrich

Noise2Void - Why does it work?



Supervised:

Minimising $\sum_i (s_i - \hat{s}_i)^2 \longrightarrow \hat{s}_i \approx \mathbb{E}_{p(s_i|\boldsymbol{x})}[s_i]$

N2V:

Minimising $\sum_i (x_i - \hat{s}_i)^2 \longrightarrow \hat{s}_i \approx \mathbb{E}_{p(x_i|\tilde{\boldsymbol{x}}_i)}[x_i]$

$$= \mathbb{E}_{p(s_i|\tilde{\boldsymbol{x}}_i)}[s_i]$$

Why it works:

$$\mathbb{E}_{p(x_i|\tilde{\boldsymbol{x}}_i)}[x_i]$$

$$= \int x_i \, p(x_i|\tilde{\boldsymbol{x}}_i) \, dx_i$$

Marginalisation

$$= \int x_i \int p(x_i, s_i|\tilde{\boldsymbol{x}}_i) \; ds_i \, dx_i$$

Product rule

$$= \int x_i \int p(x_i|s_i, \tilde{\boldsymbol{x}}_i) p(s_i|\tilde{\boldsymbol{x}}_i) \; ds_i \, dx_i$$

$$= \int \int x_i p(x_i|s_i, \tilde{\boldsymbol{x}}_i) p(s_i|\tilde{\boldsymbol{x}}_i) \; ds_i \, dx_i$$

$$= \int \int x_i p(x_i|s_i, \tilde{\boldsymbol{x}}_i) p(s_i|\tilde{\boldsymbol{x}}_i) \; dx_i \, ds_i$$

$$= \int p(s_i|\tilde{\boldsymbol{x}}_i) \int x_i \, p(x_i|s_i, \tilde{\boldsymbol{x}}_i) \; dx_i \, ds_i$$

$$= \int p(s_i|\tilde{\boldsymbol{x}}_i) \int x_i \, p(x_i|s_i, \tilde{\boldsymbol{x}}_i) \; dx_i \, ds_i$$

Cond. independence

$$= \int p(s_i|\tilde{\boldsymbol{x}}_i) \int x_i \, p(x_i|s_i) \; dx_i \, ds_i$$

Expected value

$$= \int p(s_i|\tilde{\boldsymbol{x}}_i) \, \mathbb{E}_{p(x_i|\boldsymbol{s}_i)}[x_i] \; ds_i$$

Zero centred noise

$$= \int p(s_i|\tilde{\boldsymbol{x}}_i) s_i \; ds_i$$

Expected value

$$= \mathbb{E}_{p(s_i|\tilde{\boldsymbol{x}}_i)}[s_i]$$

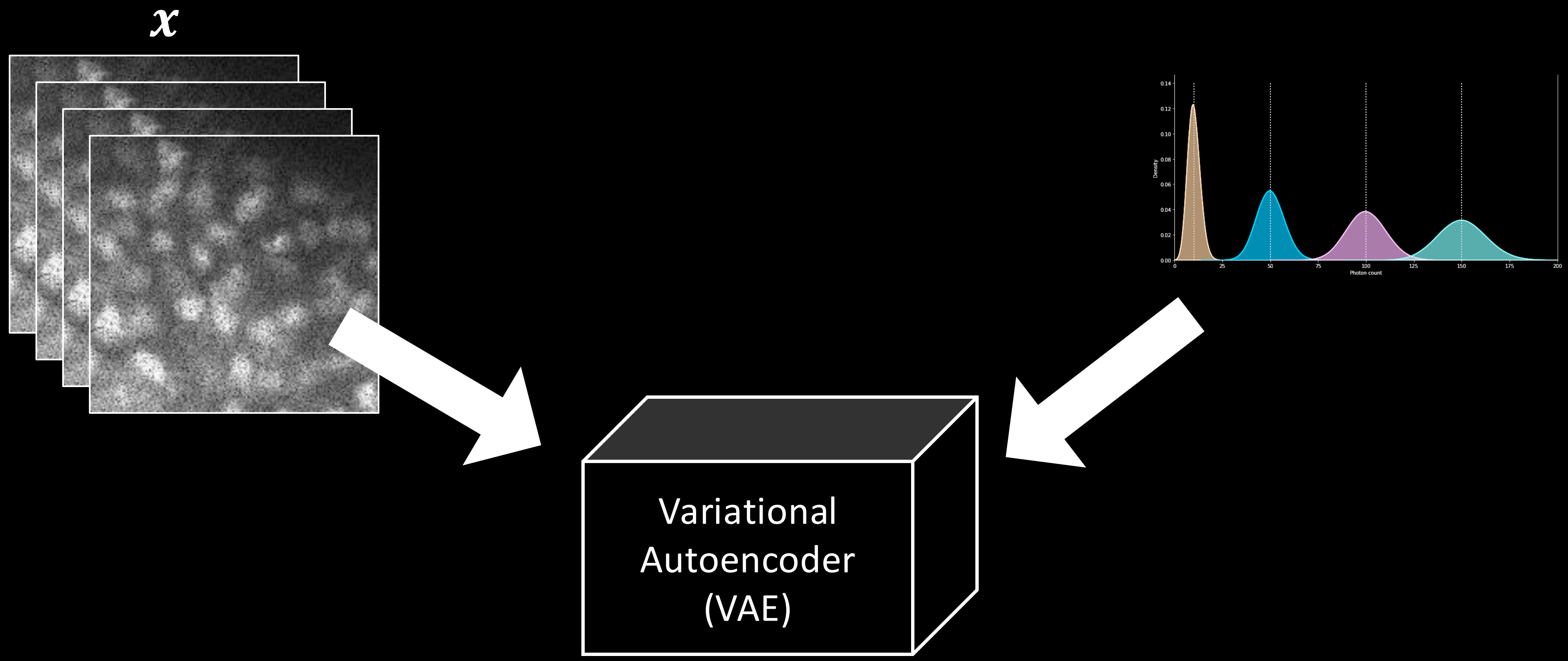
DivNoising and HDN

Diverse Solutions - Accounting for uncertainty.

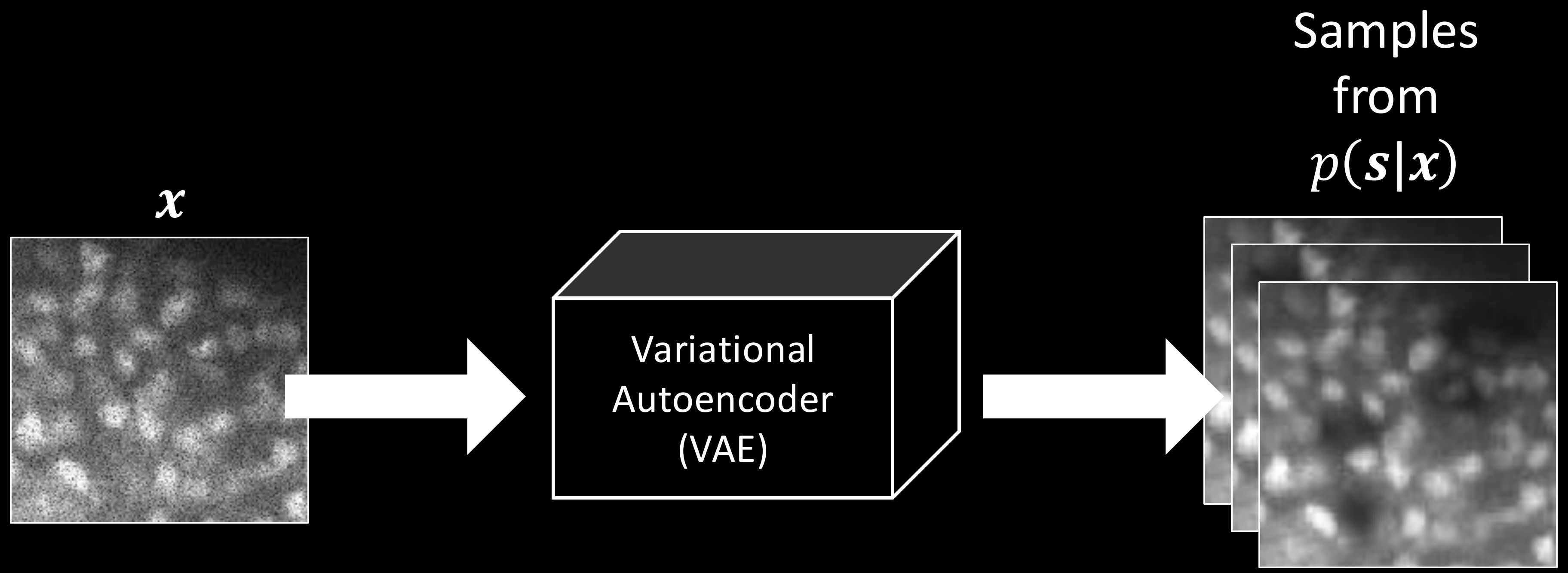
Prakash, Krull, Jug 2020

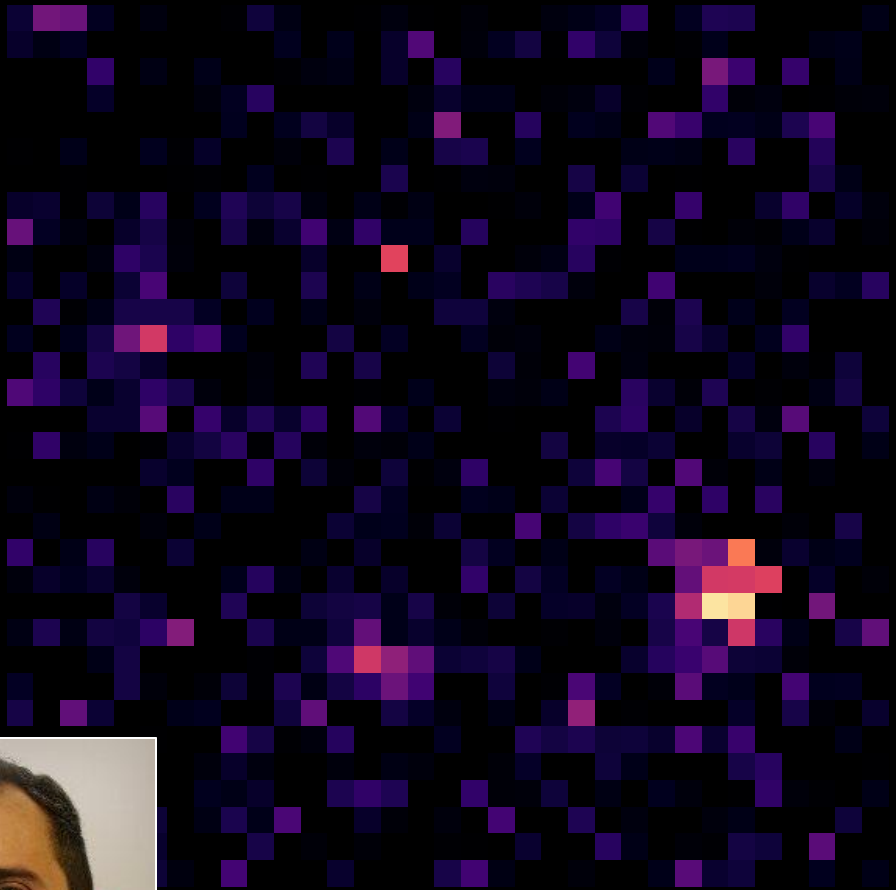
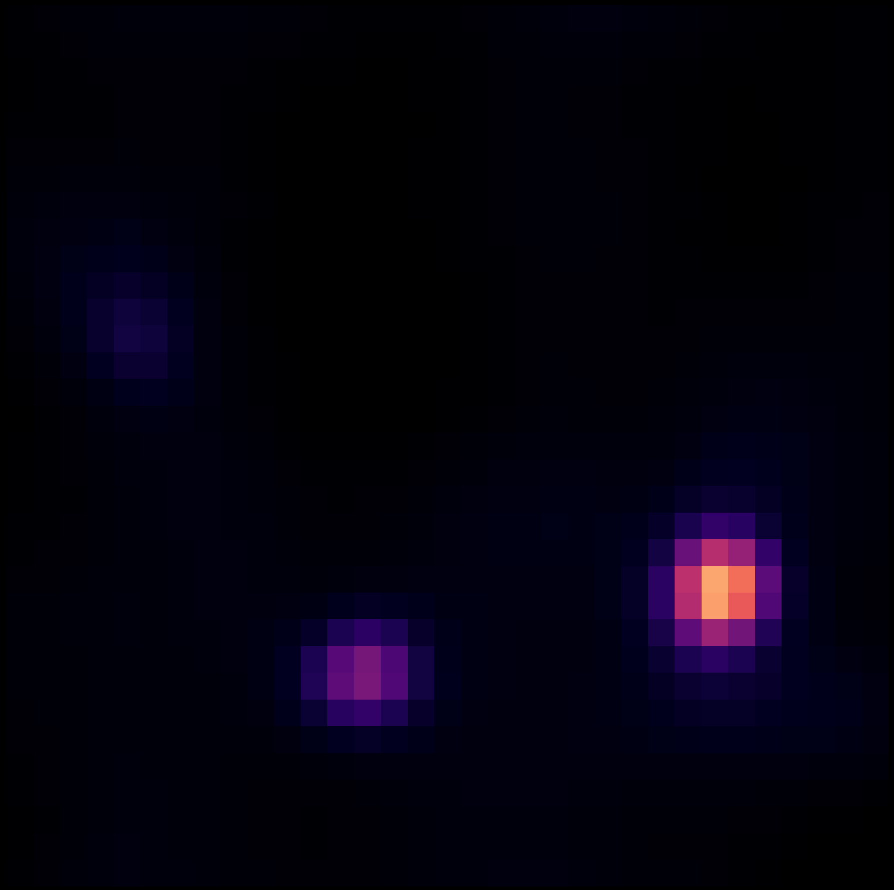
Prakash, Delbracio, Milanfar, Jug 2022

DivNoising Training



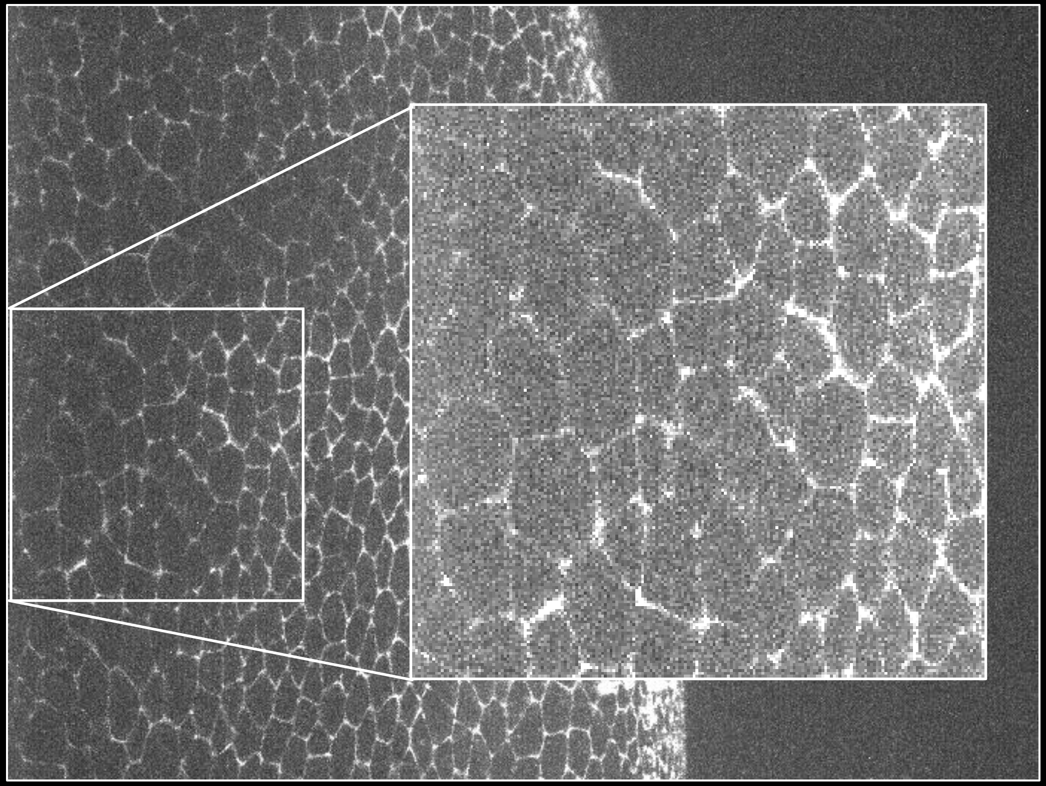
DivNoising Testing



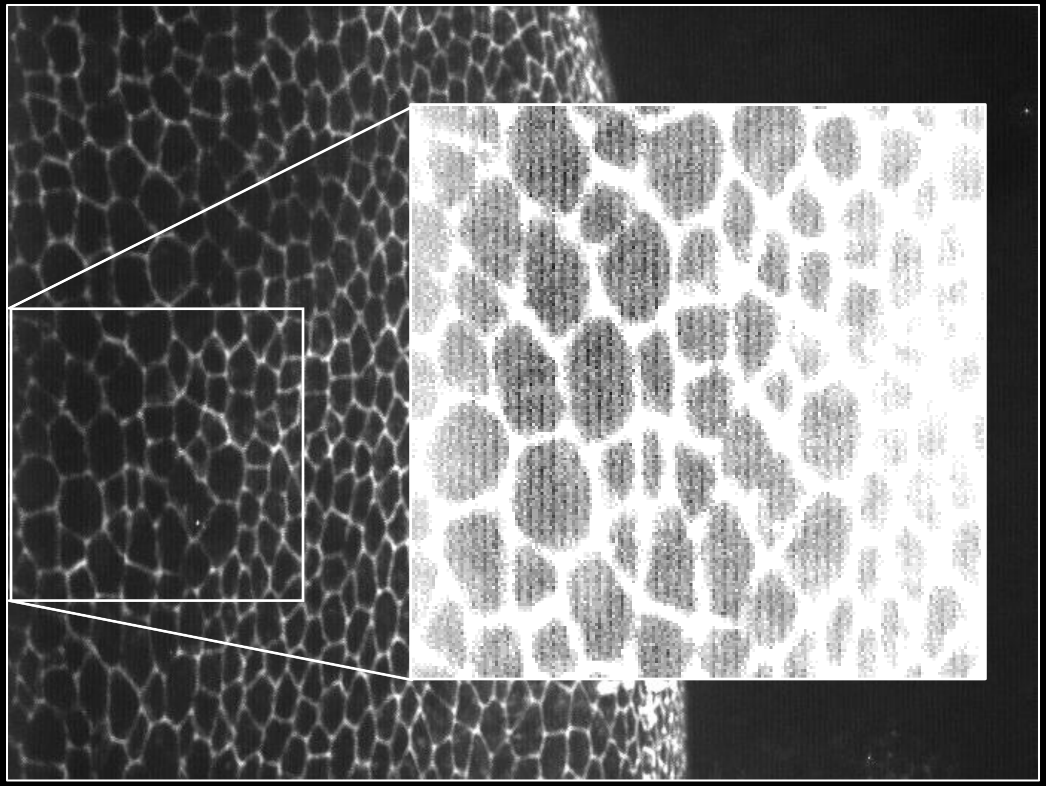



Data by Davide Calebiro

Noise2Void - results



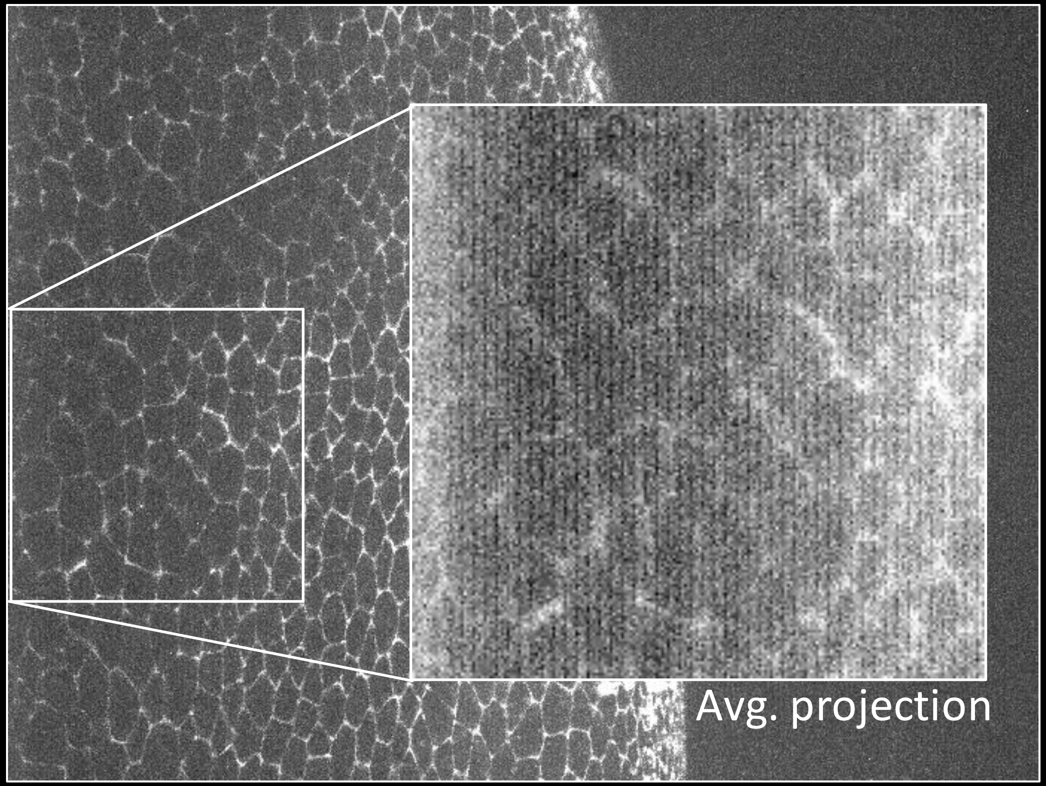
Input (max projection)



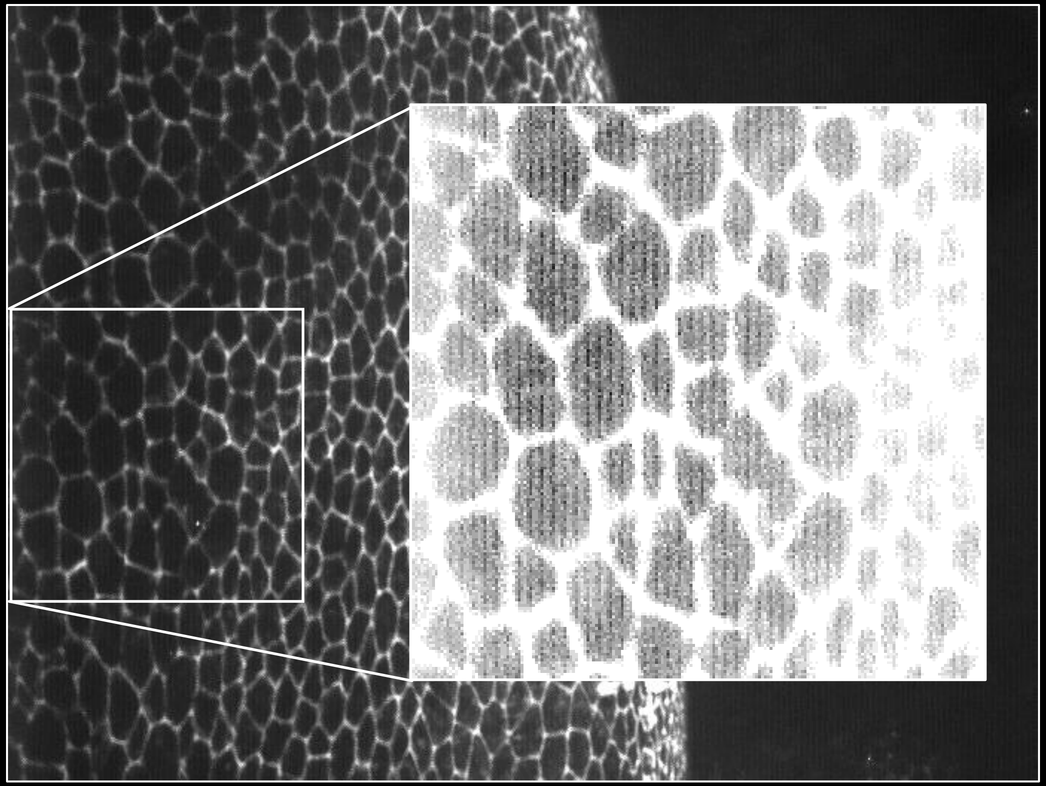
Noise2Void (max projection)

Data by Romina Piscitel, Eaton lab at MPI-CBG

Noise2Void - limitations

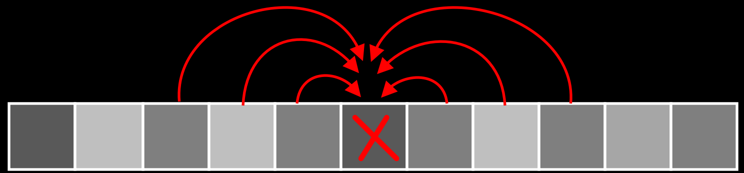


Input (max projection)

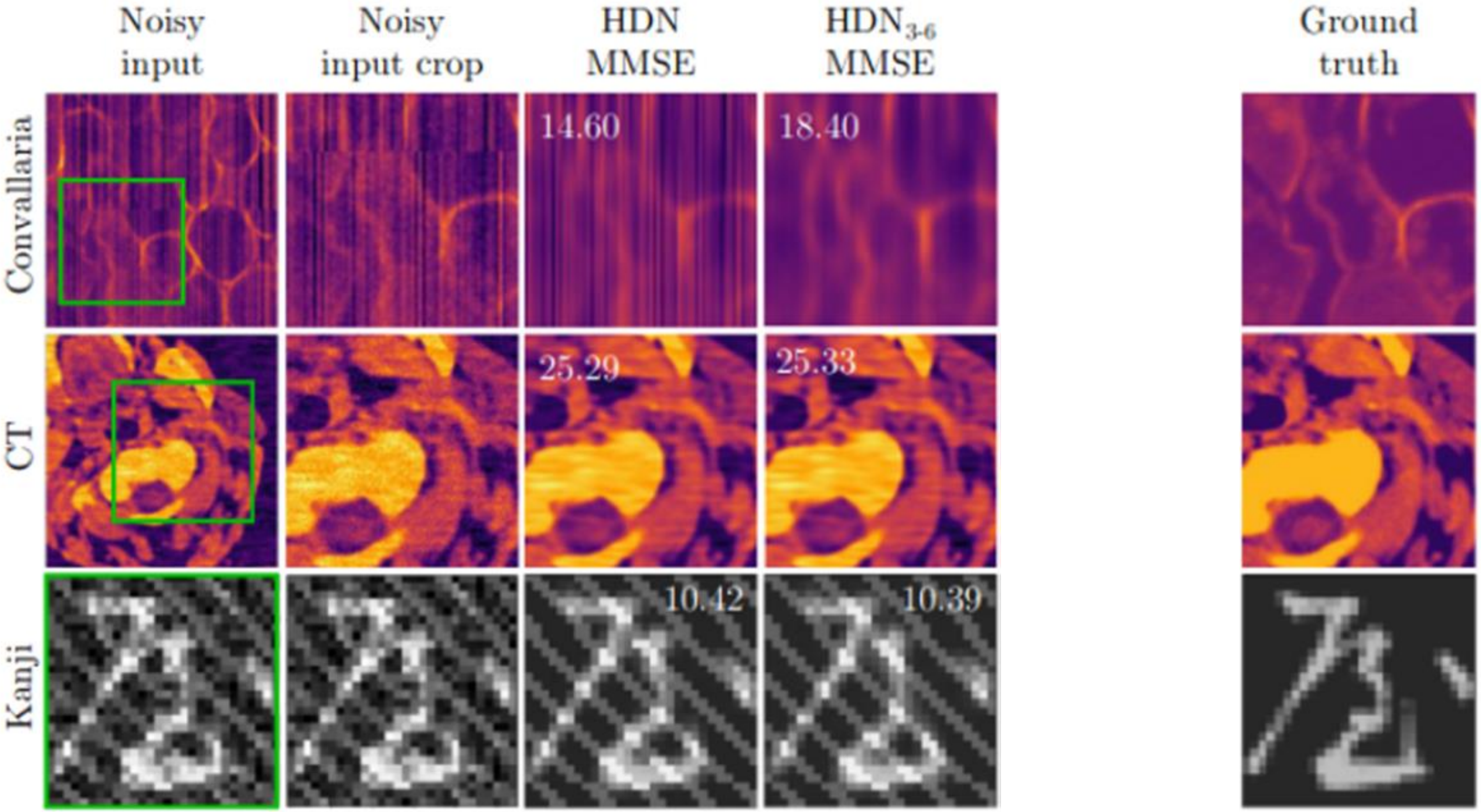


Noise2Void (max projection)

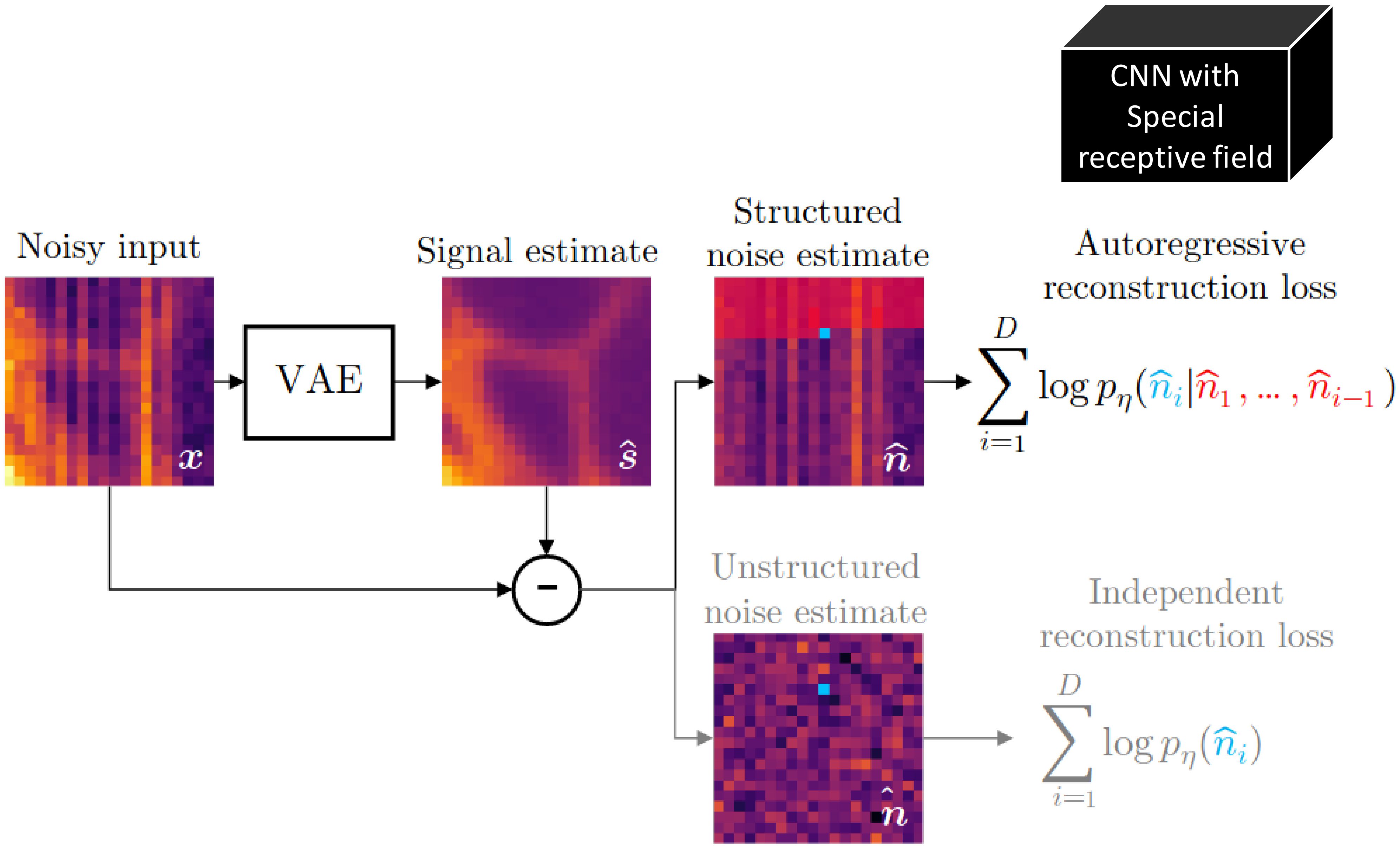
Data by Romina Piscitel, Eaton lab at MPI-CBG



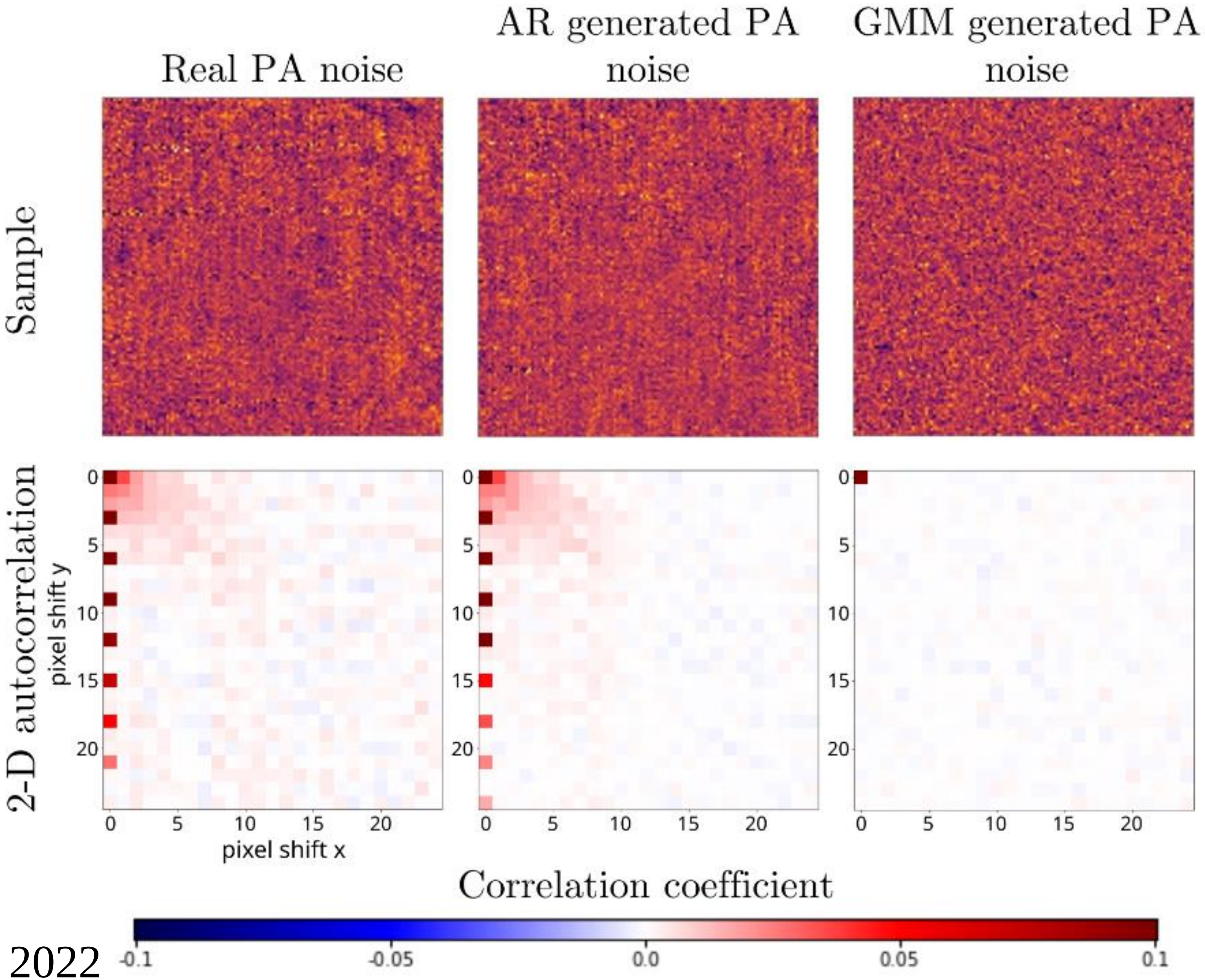
What about structured Noise? (Thanks to Ben Salmon)



Salmon and Krull 2022



Salmon and Krull 2022



Salmon and Krull 2022

Thank you!