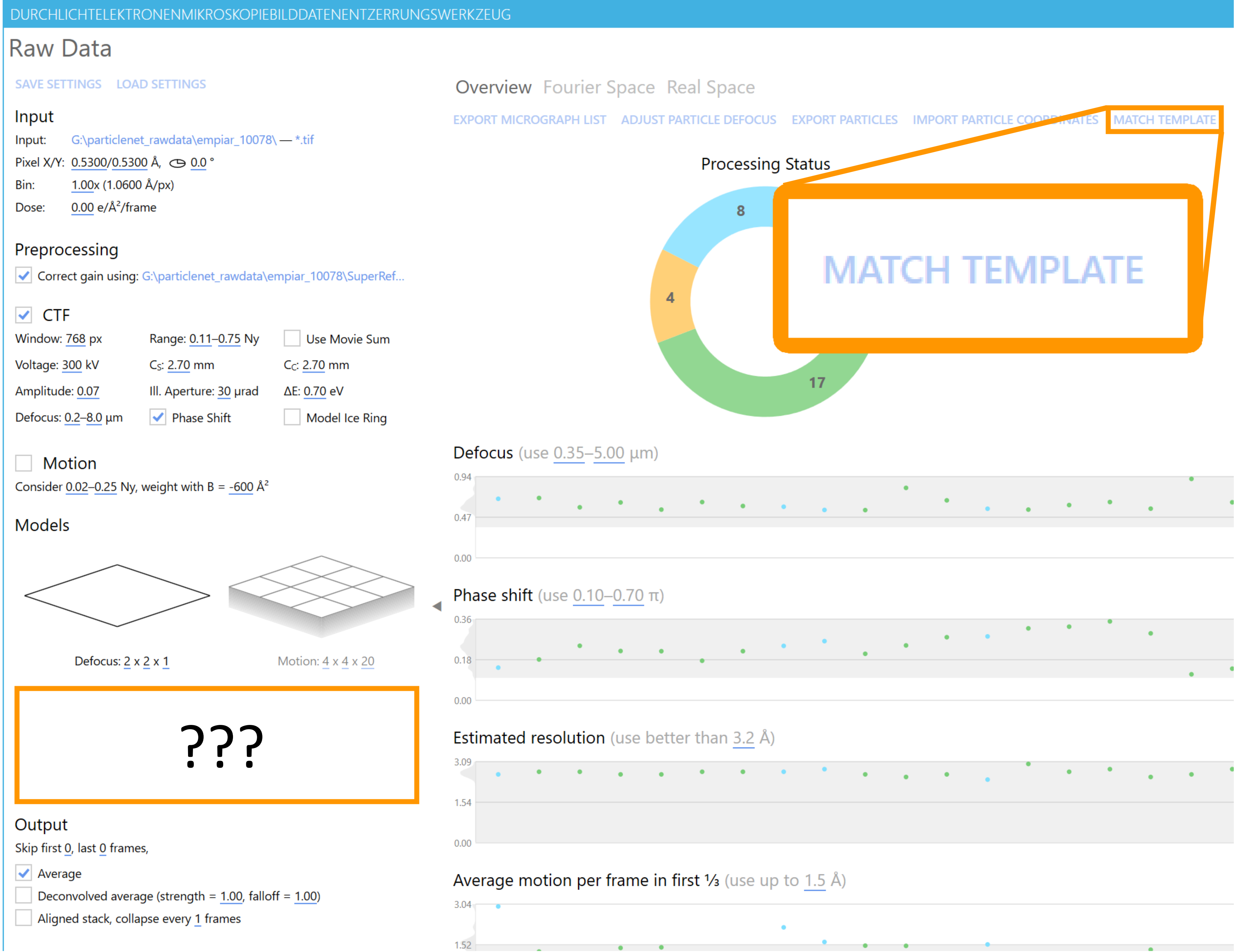
Getting started with Differentiable Programming in cryo–EM

$$\frac{dy}{dx}$$



DURCHLICHTELEKTRONENMIKROSKOPIEBILDDATENENTZERRUNGSWERKZEUG
Raw Data
SAVE SETTINGS
LOAD SETTINGS
Input
Input: G:\particlenet_rawdata\empiar_10078\ — *.tif
Pixel X/Y: 0.5300/0.5300 Å, 0.0 °
Bin: 1.00x (1.0600 Å/px)
Dose: 0.00 e/Å²/frame
Preprocessing
Correct gain using: G:\particlenet_rawdata\empiar_10078\SuperRef...
CTF
Window: 768 px
Range: 0.11–0.75 Ny
Use Movie Sum
Voltage: 300 kV
Cs: 2.70 mm
Cc: 2.70 mm
Amplitude: 0.07
Ill. Aperture: 30 µrad
ΔE: 0.70 eV
Defocus: 0.2–8.0 µm
Phase Shift
Model Ice Ring
Motion
Consider 0.02–0.25 Ny, weight with B = -600 Å²
Models
Defocus: 2 x 2 x 1
Motion: 4 x 4 x 20
???
Output
Skip first 0, last 0 frames,
Average
Deconvolved average (strength = 1.00, falloff = 1.00)
Aligned stack, collapse every 1 frames
Overview
Fourier Space
Real Space
EXPORT MICROGRAPH LIST
ADJUST PARTICLE DEFOCUS
EXPORT PARTICLES
IMPORT PARTICLE COORDINATES
MATCH TEMPLATE
Processing Status
8
4
17
MATCH TEMPLATE
Defocus (use 0.35–5.00 µm)
0.94
0.47
0.00
Phase shift (use 0.10–0.70 π)
0.36
0.18
0.00
Estimated resolution (use better than 3.2 Å)
3.09
1.54
0.00
Average motion per frame in first ⅓ (use up to 1.5 Å)
3.04
1.52

Oli B. Clarke
@OliBClarke

Following

Beautiful structure... but what poor sod had the task of picking 270k particles manually!?? 😱 rdcu.be/KDbX

Bui lab
@builab

Replying to @OliBClarke

@Yoshi__Ichikawa lol u can do it

5:45 PM - 7 Apr 2018

Alyazan Albarghash @Aly_Albarghash · Apr 5

Replying to @OliBClarke @kshbeckham

Oh dear ... Oh dear
not a very unusual task though :$

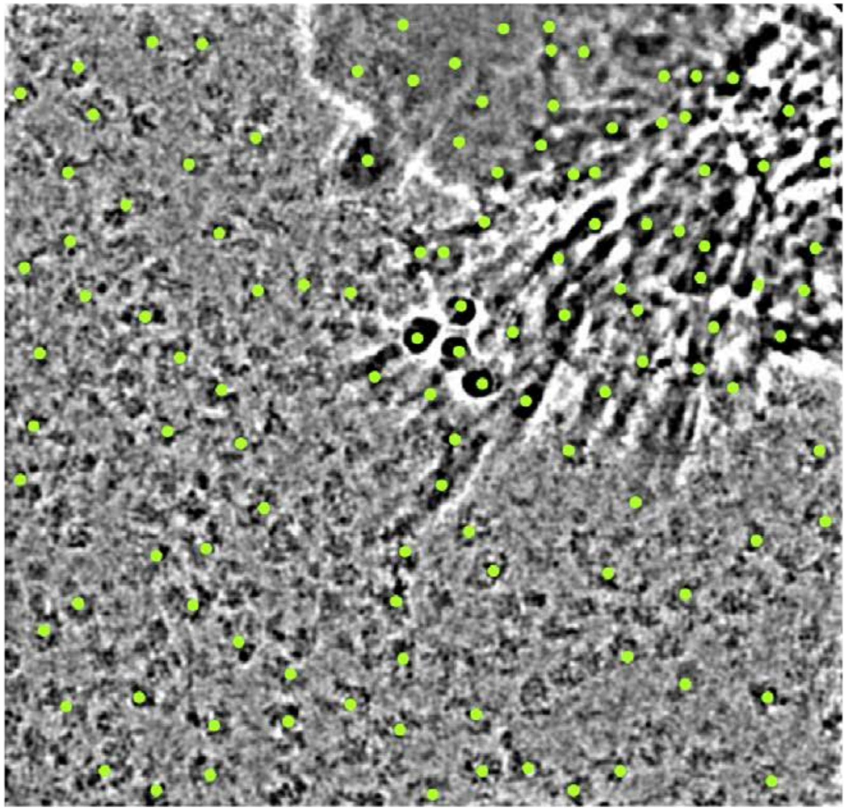
1

Joshua Lobo @boreas_cryo · Apr 5

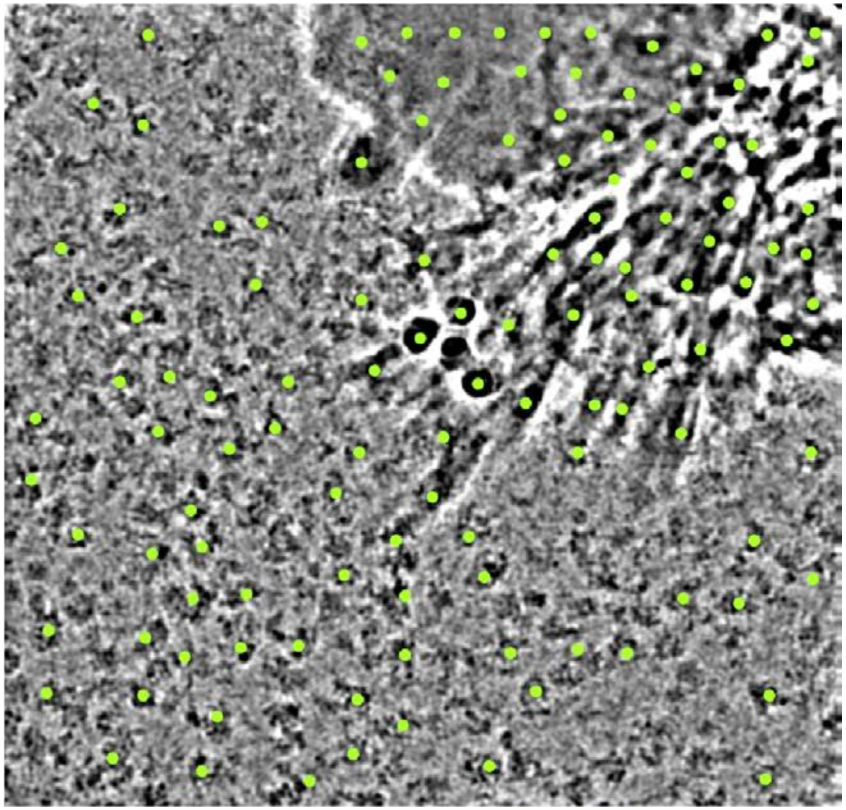
Really ? :O Haven't crossed 9k yet

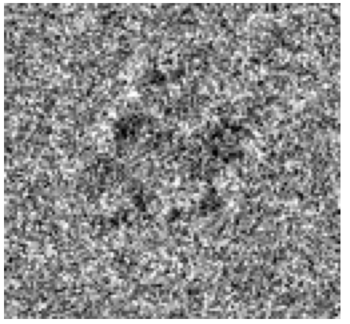
1

RELION, Gaussian blob



RELION, 2D class templates





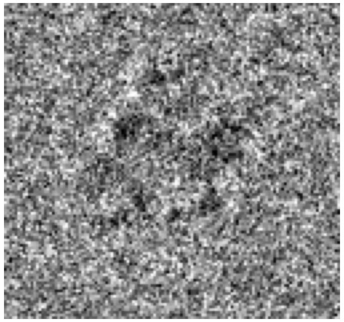
- Noisy
- Irregularly shaped
- Binary decision

Who would win against a machine?

ImageNet1000

- 1M+ natural images
- 1000 classes
- 5 % human error rate

This 1 particle boi



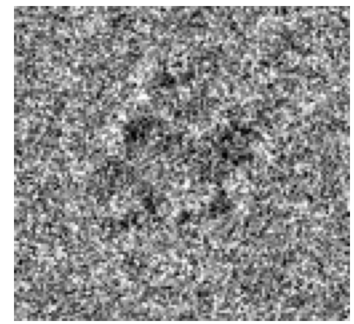
- Noisy
- Irregularly shaped
- Binary decision

Who would win against a machine?

Driving a car

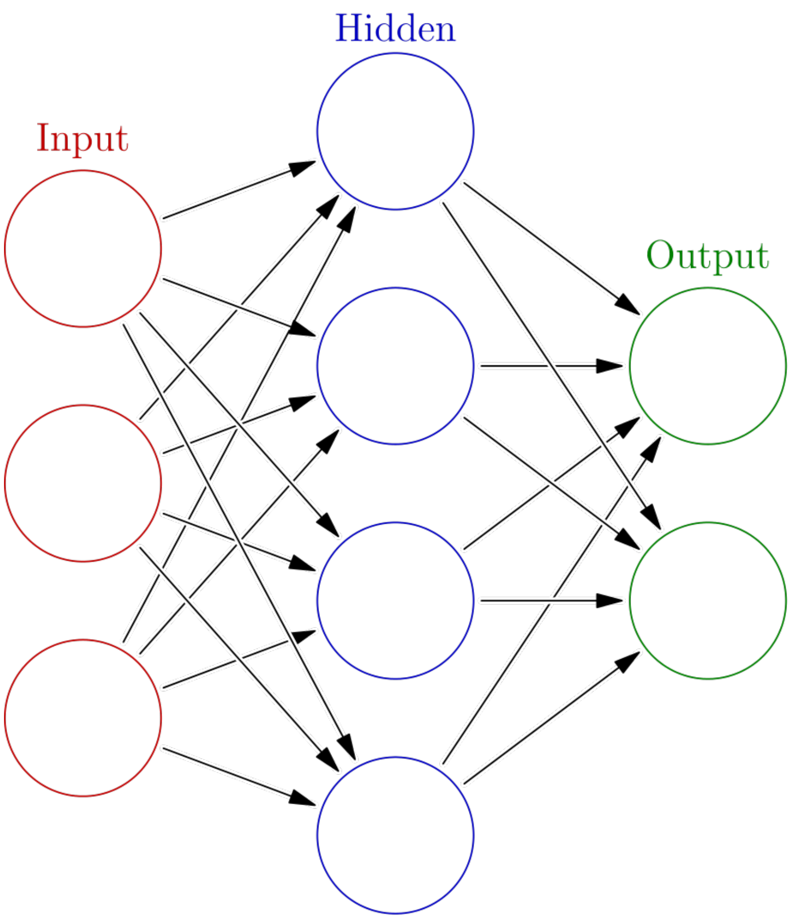
- &^%#@!

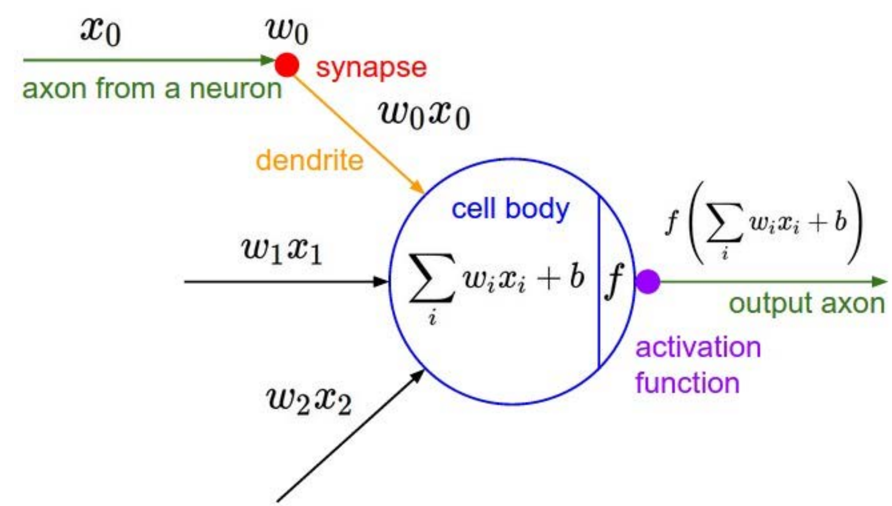
This 1 particle boi

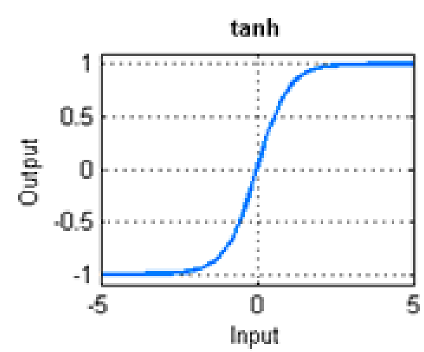


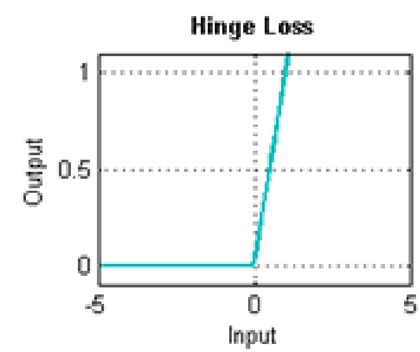
- Noisy
- Irregularly shaped
- Binary decision

A graph of differentiable functions

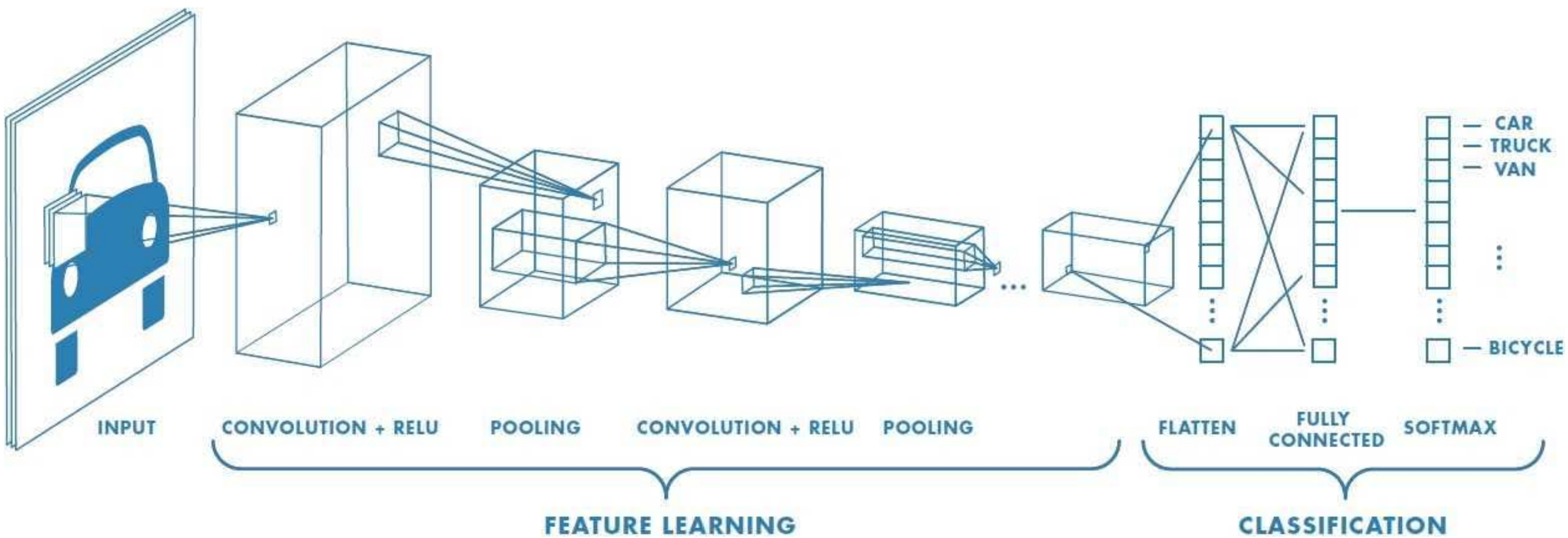




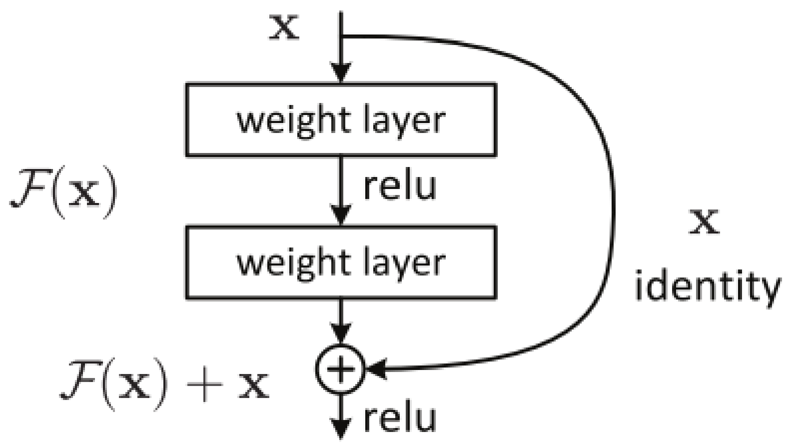


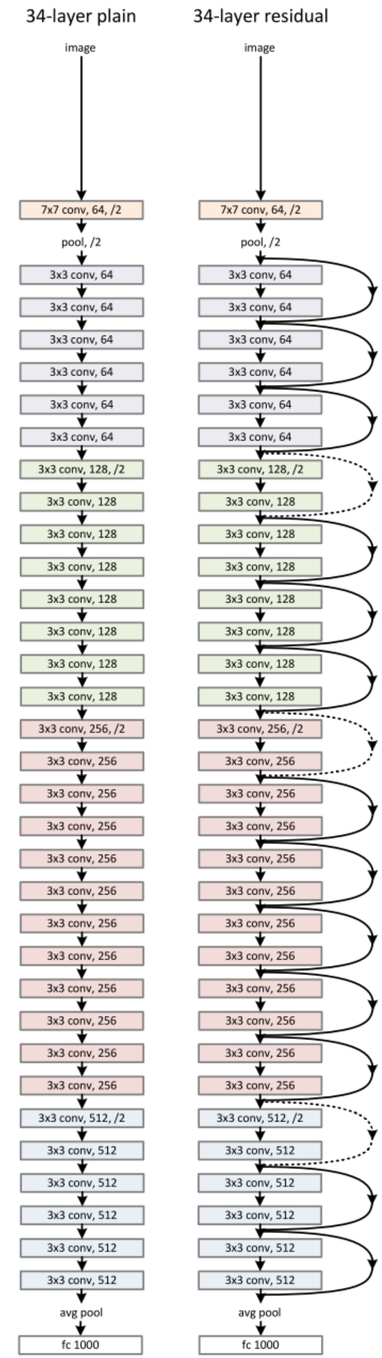


How does a ConvNet work?



Residual networks

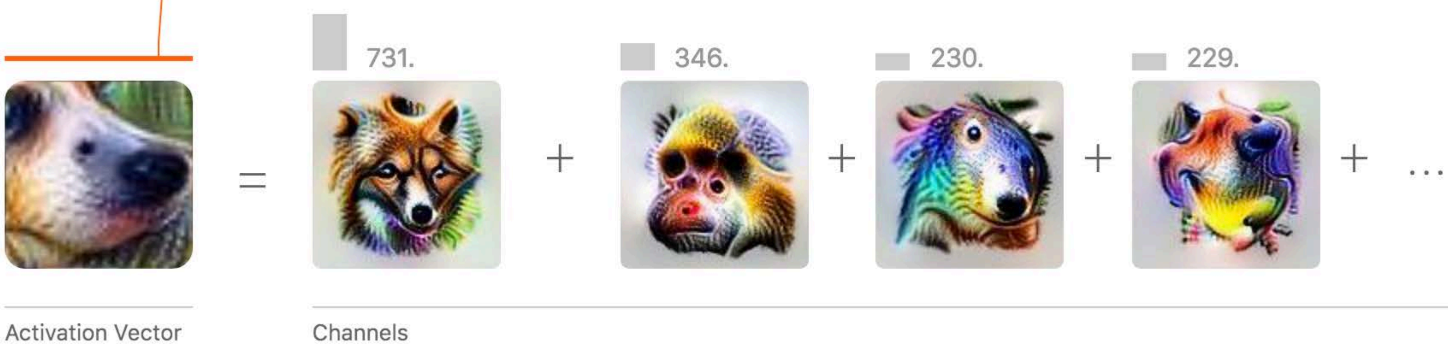




Not a black box

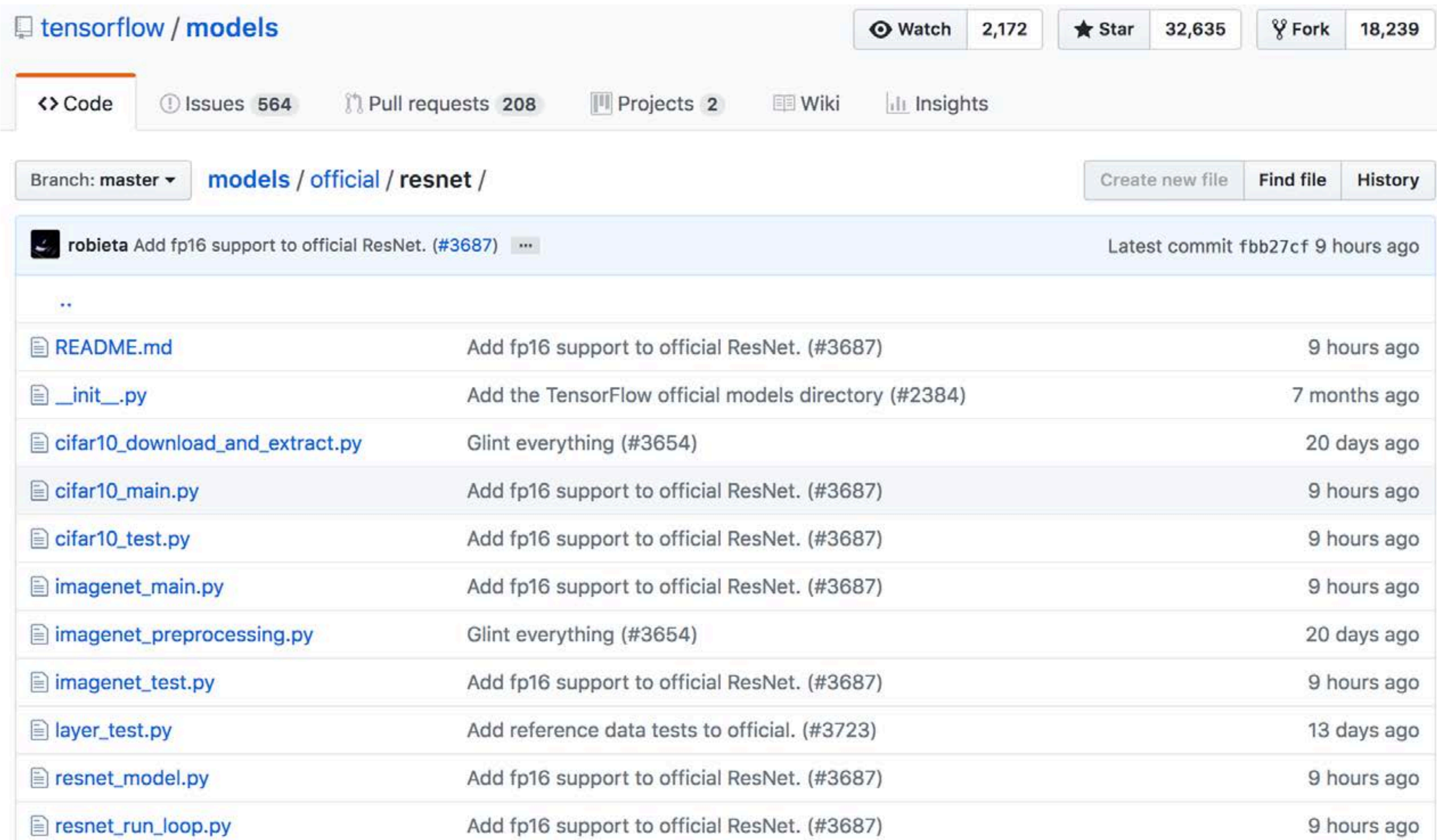
What Does the Network See?

Semantic dictionaries give us a fine-grained look at an activation: what does each single neuron detect? Building off this representation, we can also consider an activation vector as a whole. Instead of visualizing individual neurons, we can instead visualize the *combination* of neurons that fire at a given spatial location. (Concretely, we optimize the image to maximize the dot product of its activations with the original activation vector.)



https://distill.pub/2018/building-blocks/

To start: Don‘t start from scratch!



Code, part 1

```python
def boxnet_resnet_v2_generator(resnet_size, num_classes, data_format=None):
  num_blocks = (resnet_size - 2) // 6

  def model(inputs, is_training):

    inputs = conv2d_fixed_padding(
        inputs=inputs, filters=128, kernel_size=5, strides=3,
        data_format=data_format)
    inputs = tf.identity(inputs, 'initial_conv')

    inputs = block_layer(
        inputs=inputs, filters=16, block_fn=building_block, blocks=num_blocks,
        strides=1, is_training=is_training, name='block_layer1',
        data_format=data_format)
    inputs = block_layer(
        inputs=inputs, filters=32, block_fn=building_block, blocks=num_blocks,
        strides=2, is_training=is_training, name='block_layer2',
        data_format=data_format)
    inputs = block_layer(
        inputs=inputs, filters=64, block_fn=building_block, blocks=num_blocks,
        strides=2, is_training=is_training, name='block_layer3',
        data_format=data_format)

    inputs = batch_norm_relu(inputs, is_training, data_format)
    inputs = tf.layers.average_pooling2d(
        inputs=inputs, pool_size=8, strides=1, padding='VALID',
        data_format=data_format)
    inputs = tf.identity(inputs, 'final_avg_pool')
    inputs = tf.reshape(inputs, [-1, 64])
    inputs = tf.layers.dense(inputs=inputs, units=num_classes)
    inputs = tf.identity(inputs, 'final_dense')
    return inputs

  return model
```

```python
def batch_norm_relu(inputs, is_training, data_format):
  inputs = tf.layers.batch_normalization(
      inputs=inputs, axis=1 if data_format == 'channels_first' else 3,
      momentum=_BATCH_NORM_DECAY, epsilon=_BATCH_NORM_EPSILON, center=True,
      scale=True, training=is_training, fused=True)
  inputs = tf.nn.relu(inputs)
  return inputs


def fixed_padding(inputs, kernel_size, data_format):
  pad_total = kernel_size - 1
  pad_beg = pad_total // 2
  pad_end = pad_total - pad_beg

  padded_inputs = tf.pad(inputs, [[0, 0], [pad_beg, pad_end],
                                  [pad_beg, pad_end], [0, 0]])
  return padded_inputs


def conv2d_fixed_padding(inputs, filters, kernel_size, strides, data_format):
  if strides > 1:
    inputs = fixed_padding(inputs, kernel_size, data_format)

  return tf.layers.conv2d(
      inputs=inputs, filters=filters, kernel_size=kernel_size, strides=strides,
      padding=('SAME' if strides == 1 else 'VALID'), use_bias=False,
      kernel_initializer=tf.variance_scaling_initializer(),
      data_format=data_format)


def building_block(inputs, filters, is_training, projection_shortcut, strides,
                   data_format):
  shortcut = inputs
  inputs = batch_norm_relu(inputs, is_training, data_format)

  if projection_shortcut is not None:
    shortcut = projection_shortcut(inputs)

  inputs = conv2d_fixed_padding(
      inputs=inputs, filters=filters, kernel_size=3, strides=strides,
      data_format=data_format)

  inputs = batch_norm_relu(inputs, is_training, data_format)
  inputs = conv2d_fixed_padding(
      inputs=inputs, filters=filters, kernel_size=3, strides=1,
      data_format=data_format)

  return inputs + shortcut
```

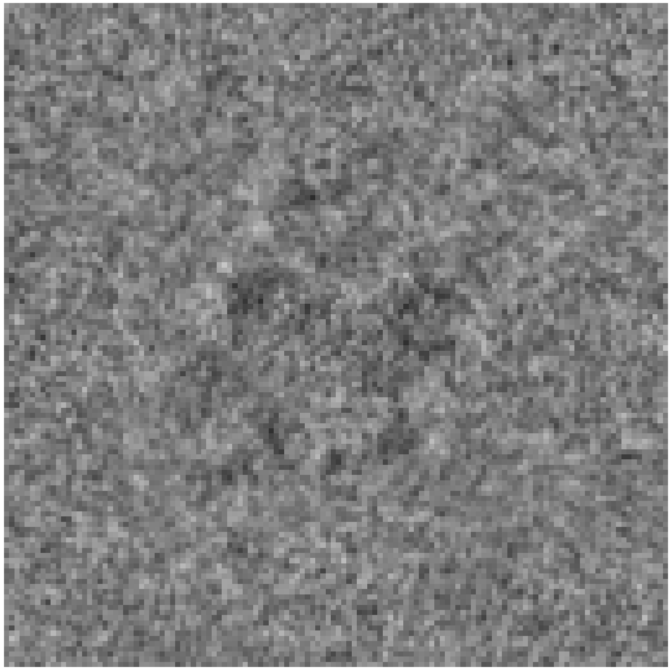
Code, part 2

45 particle species, 1500–3000 particles each:

- **23 simulated from PDBs with InSilicoTEM**
- **22 from EMPIAR & in-house**

Augmentation

- Rotate
- Shear
- Noise



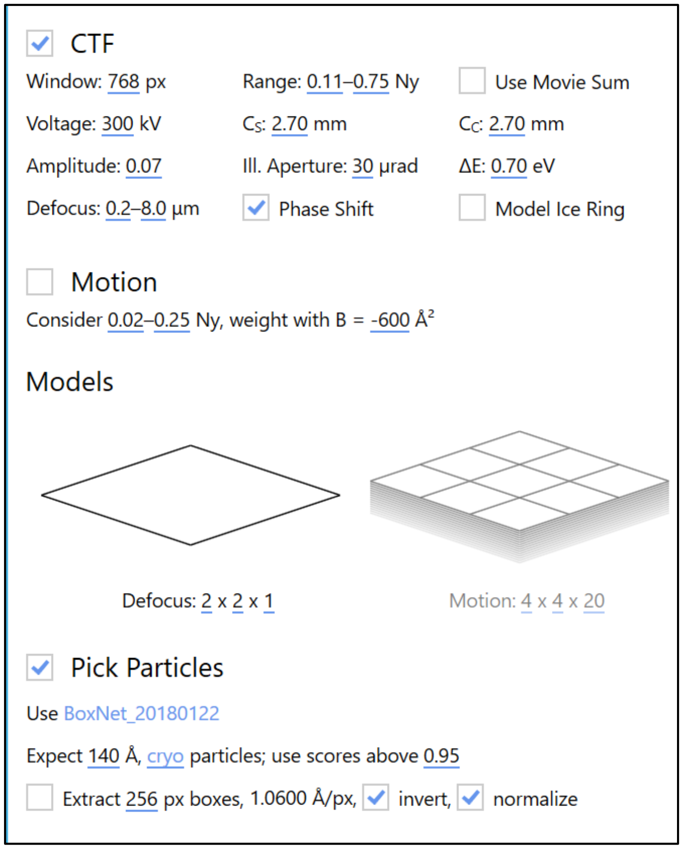
Training

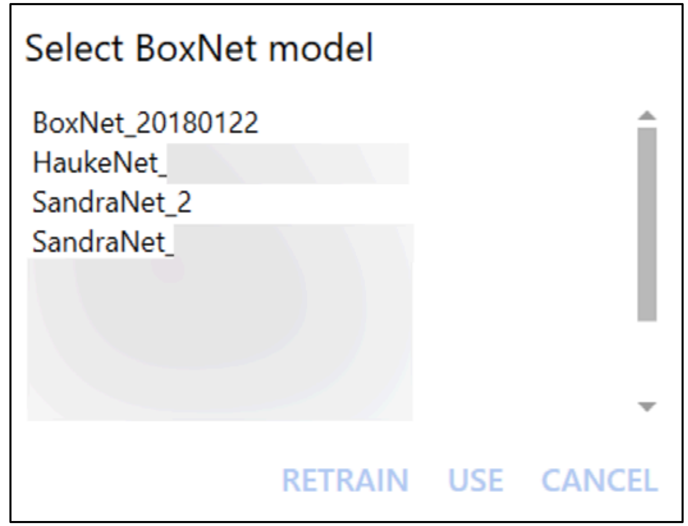
- Rescale all images to 8 A/px
- Extract positive and negative examples
- Augment examples
- Look at each example 100 times in random order
- Gradually decrease learning rate from 10^{-3} to 10^{-5}

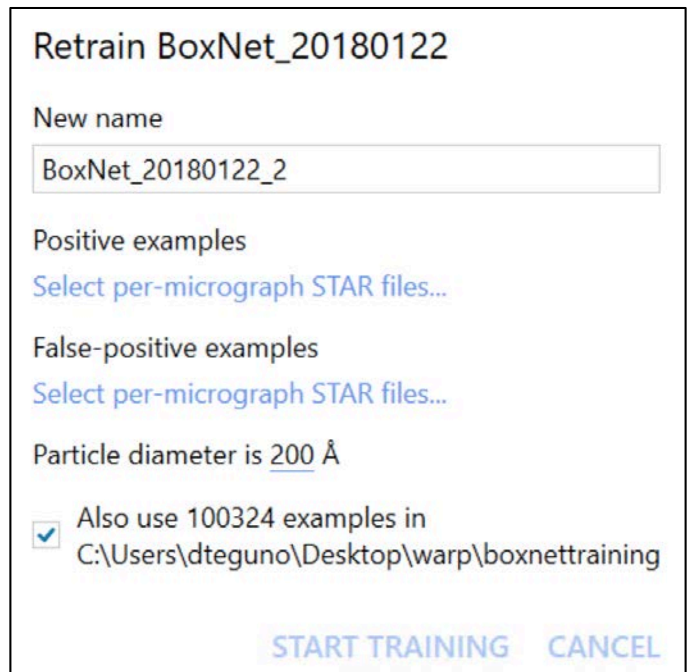
Inference

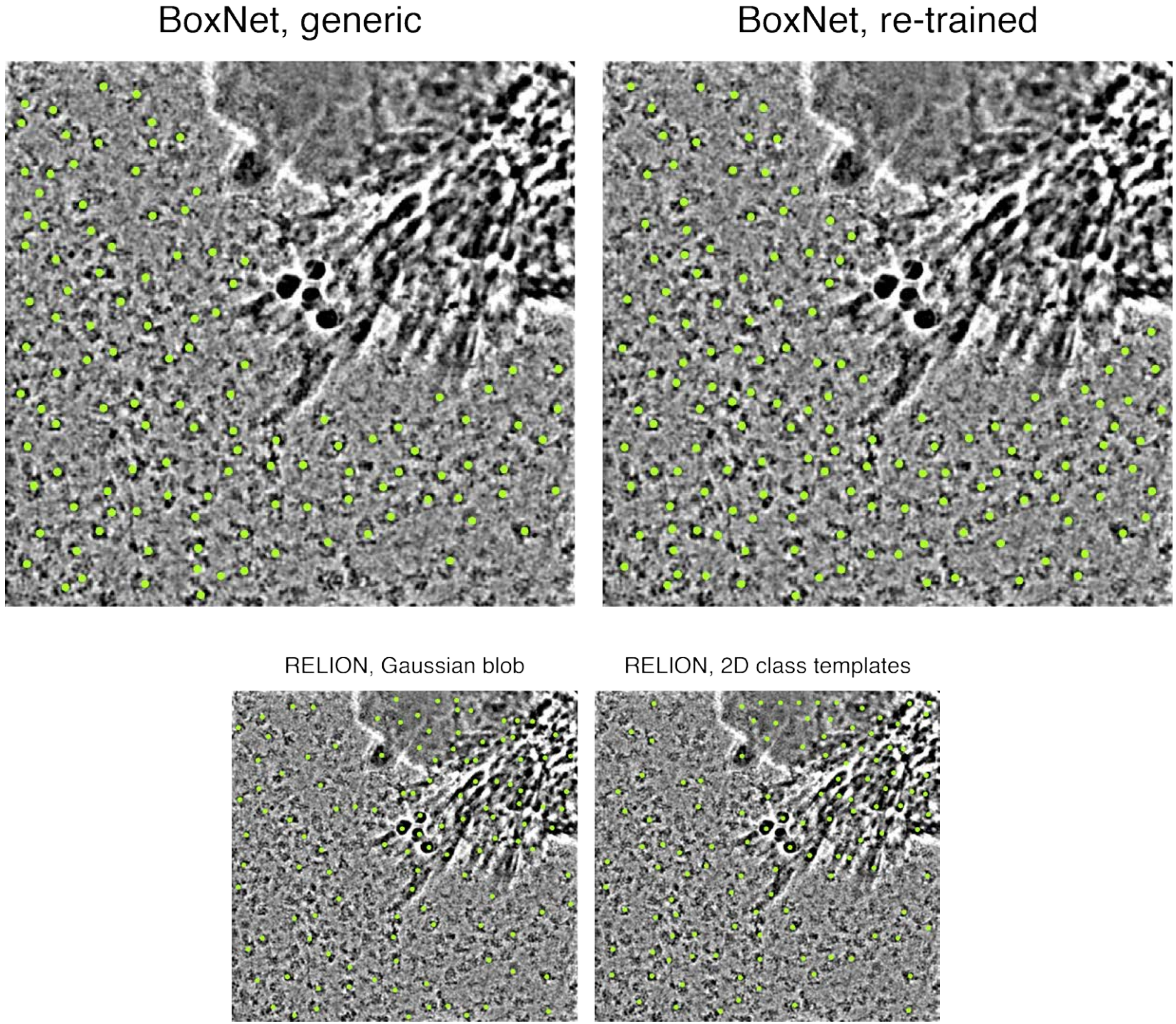
- Rescale micrograph to 8 A/px
- Extract and normalize a running window of 96^2 px
- Send it through BoxNet
- Store SoftMax results for all positions
- Binarize (typically at > 0.9)
- Find connected components
- Centroids = particle positions
- Optionally, enforce minimum distance

Integration

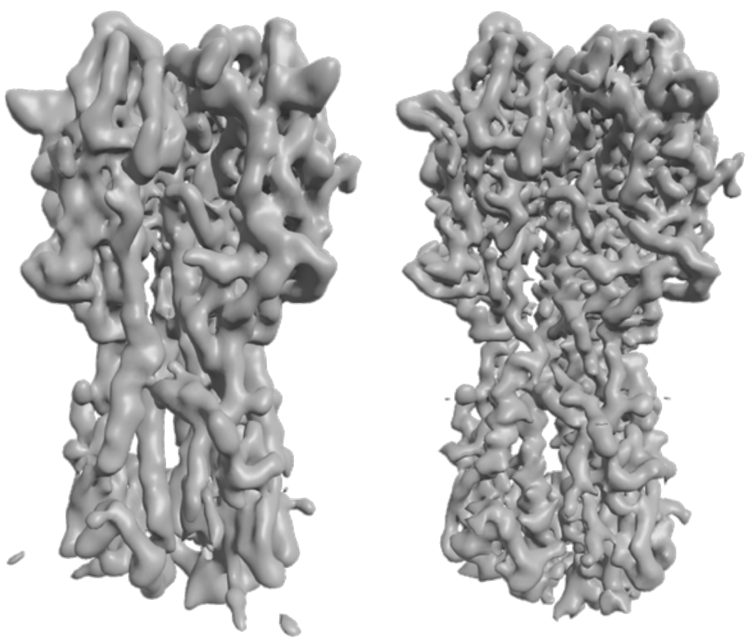


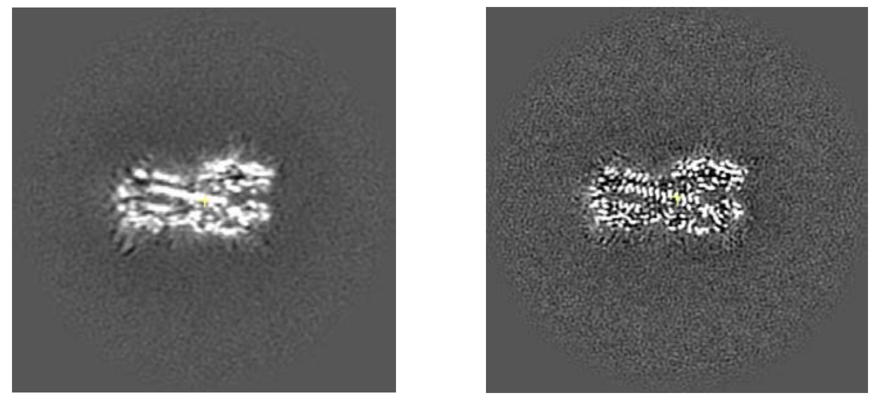


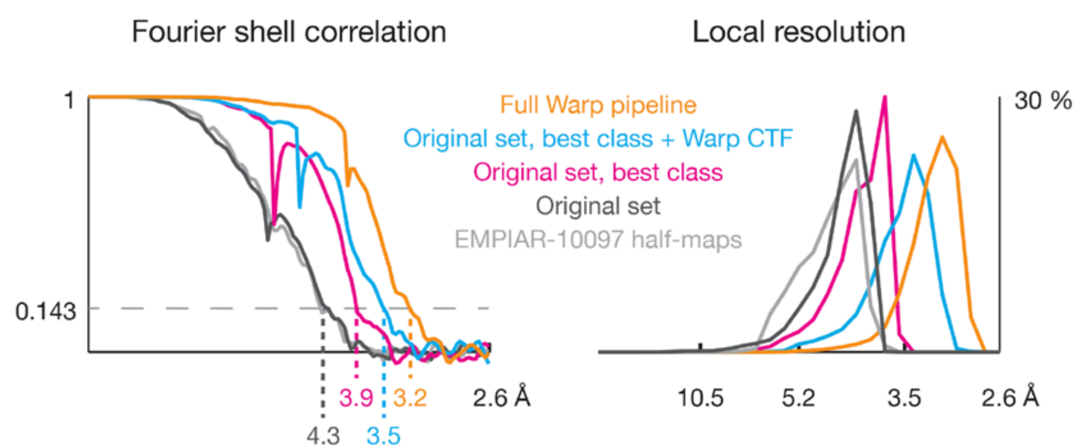


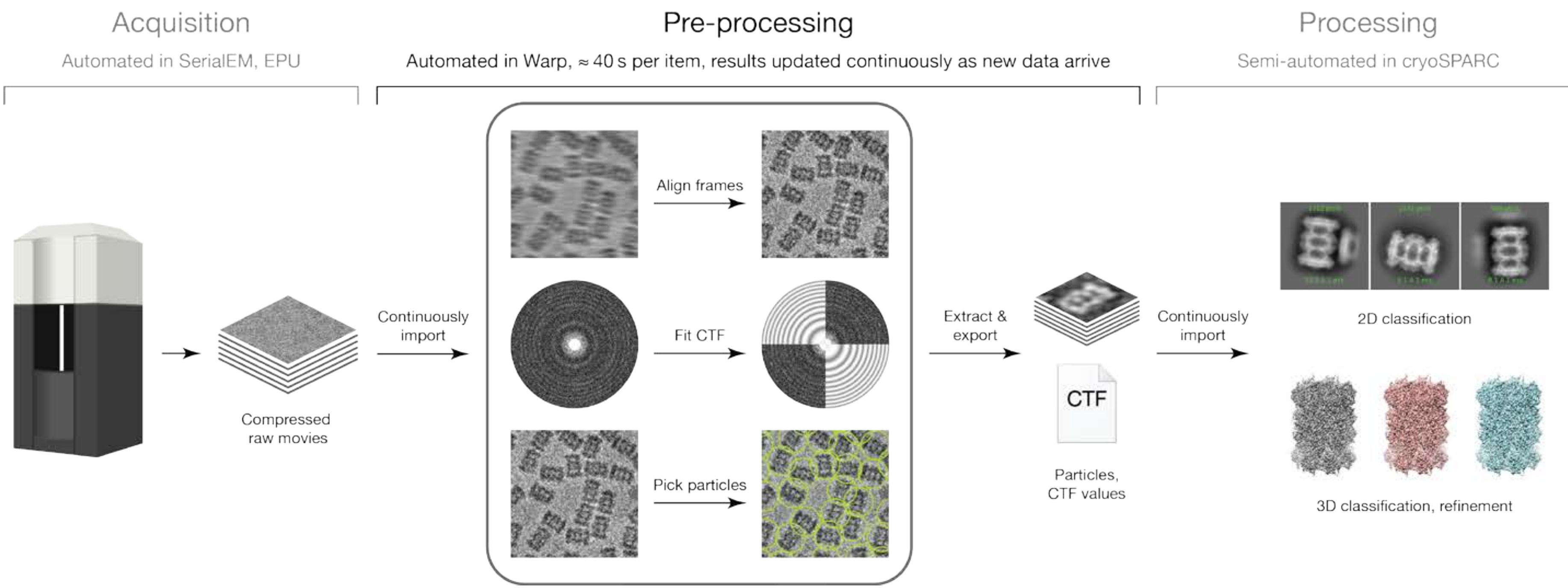

BoxNet, generic
BoxNet, re-trained
RELION, Gaussian blob
RELION, 2D class templates

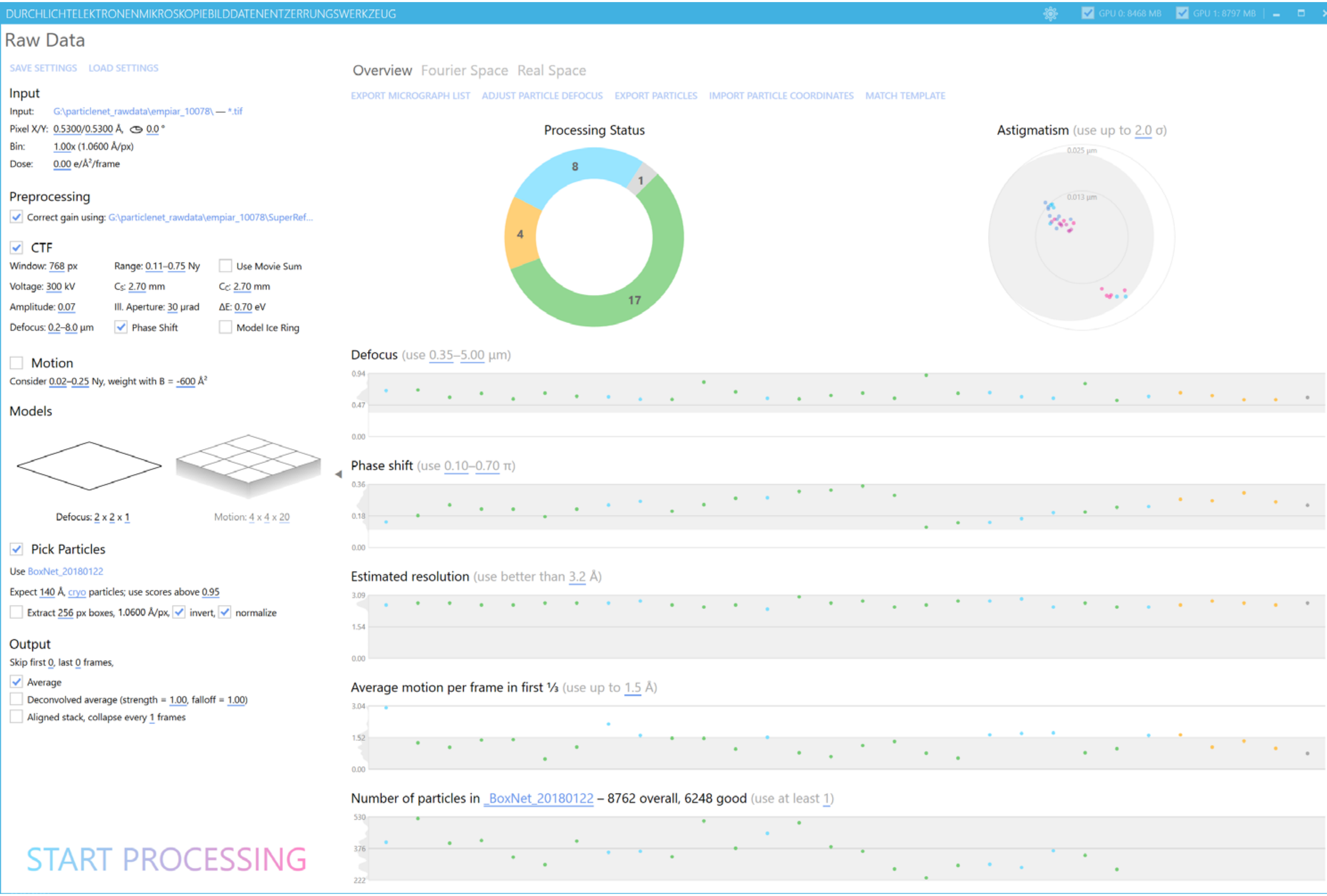
With EMPIAR-10097

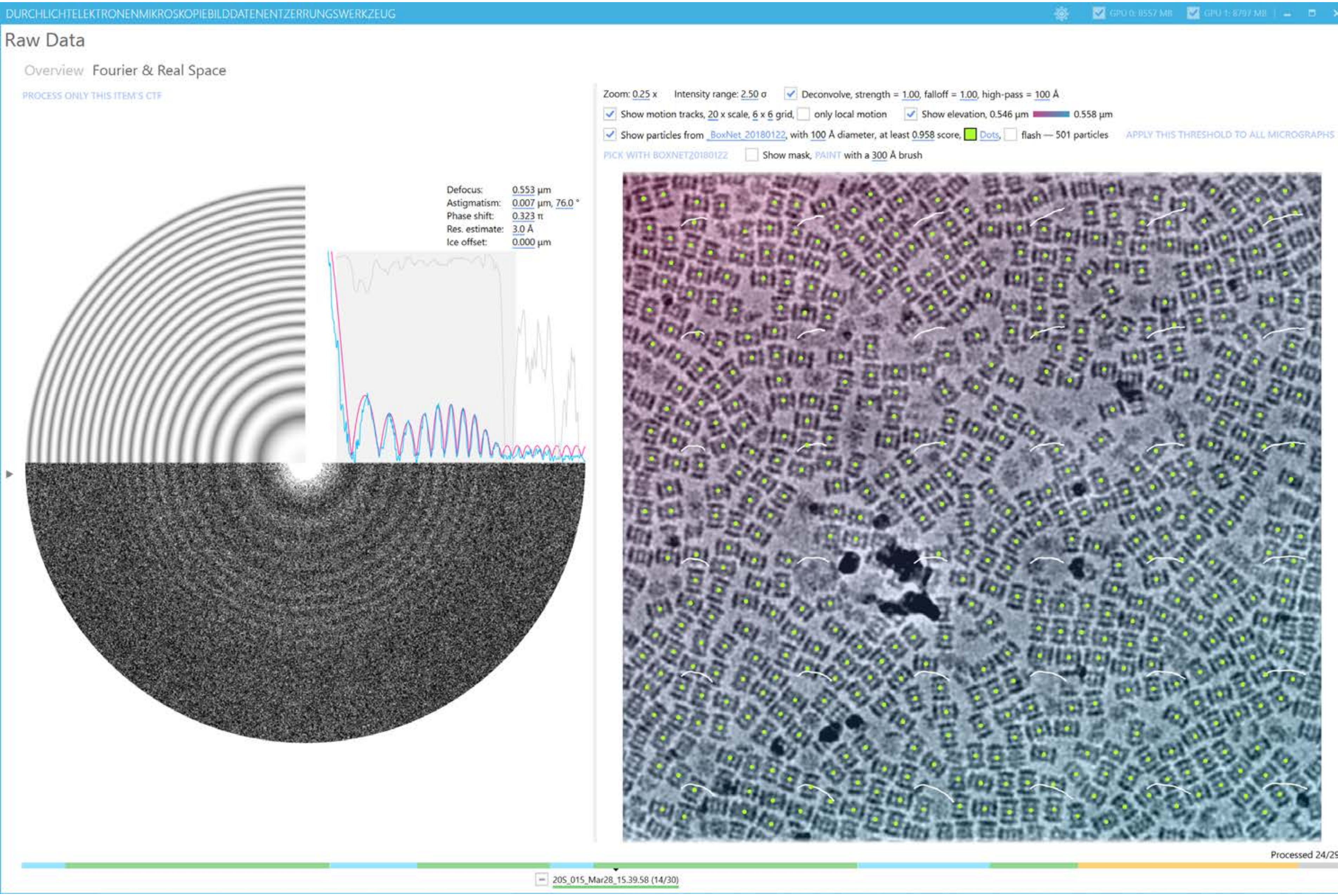







Acquisition
Automated in SerialEM, EPU
Pre-processing
Automated in Warp, ≈ 40 s per item, results updated continuously as new data arrive
Processing
Semi-automated in cryoSPARC
Compressed raw movies
Continuously import
Align frames
Fit CTF
Pick particles
Extract & export
CTF
Particles, CTF values
Continuously import
2D classification
3D classification, refinement


DURCHLICHTELEKTRONENMIKROSKOPIEBILDDATENENTZERRUNGSWERKZEUG
GPU 0: 8468 MB
GPU 1: 8797 MB
Raw Data
SAVE SETTINGS
LOAD SETTINGS
Input
Input: G:\particlenet_rawdata\empiar_10078\ — *.tif
Pixel X/Y: 0.5300/0.5300 Å, 0.0 °
Bin: 1.00x (1.0600 Å/px)
Dose: 0.00 e/Å²/frame
Preprocessing
Correct gain using: G:\particlenet_rawdata\empiar_10078\SuperRef...
CTF
Window: 768 px
Range: 0.11–0.75 Ny
Use Movie Sum
Voltage: 300 kV
Cs: 2.70 mm
Cc: 2.70 mm
Amplitude: 0.07
Ill. Aperture: 30 µrad
ΔE: 0.70 eV
Defocus: 0.2–8.0 µm
Phase Shift
Model Ice Ring
Motion
Consider 0.02–0.25 Ny, weight with B = -600 Å²
Models
Defocus: 2 x 2 x 1
Motion: 4 x 4 x 20
Pick Particles
Use BoxNet_20180122
Expect 140 Å, cryo particles; use scores above 0.95
Extract 256 px boxes, 1.0600 Å/px, invert, normalize
Output
Skip first 0, last 0 frames,
Average
Deconvolved average (strength = 1.00, falloff = 1.00)
Aligned stack, collapse every 1 frames
START PROCESSING
Overview
Fourier Space
Real Space
EXPORT MICROGRAPH LIST
ADJUST PARTICLE DEFOCUS
EXPORT PARTICLES
IMPORT PARTICLE COORDINATES
MATCH TEMPLATE
Processing Status
8
1
4
17
Astigmatism (use up to 2.0 σ)
0.025 µm
0.013 µm
Defocus (use 0.35–5.00 µm)
0.94
0.47
0.00
Phase shift (use 0.10–0.70 π)
0.36
0.18
0.00
Estimated resolution (use better than 3.2 Å)
3.09
1.54
0.00
Average motion per frame in first ⅓ (use up to 1.5 Å)
3.04
1.52
0.00
Number of particles in BoxNet_20180122 – 8762 overall, 6248 good (use at least 1)
530
376
222


DURCHLICHTELEKTRONENMIKROSKOPIEBILDDATENENTZERRUNGSWERKZEUG
Raw Data
Overview
Fourier & Real Space
PROCESS ONLY THIS ITEM'S CTF
Defocus: 0.553 µm
Astigmatism: 0.007 µm, 76.0 °
Phase shift: 0.323 π
Res. estimate: 3.0 Å
Ice offset: 0.000 µm
Zoom: 0.25 x
Intensity range: 2.50 σ
Deconvolve, strength = 1.00, falloff = 1.00, high-pass = 100 Å
Show motion tracks, 20 x scale, 6 x 6 grid,
only local motion
Show elevation, 0.546 µm
0.558 µm
Show particles from BoxNet 20180122, with 100 Å diameter, at least 0.958 score,
Dots,
flash — 501 particles
APPLY THIS THRESHOLD TO ALL MICROGRAPHS
PICK WITH BOXNET20180122
Show mask, PAINT with a 300 Å brush
Processed 24/29.
20S_015_Mar28_15.39.58 (14/30)

Things to try

- 3D map denoising
- GANs for realistic data simulation
- Autoencoders in 2D, 3D to deal with flexibility
- Refinement with better scoring metric
- Reconstruction

General challenges

- No training data for most problems
- Memory consumption in 3D
- Very little research applicable to cryo-EM