CEITEC

Central European Institute of Technology
BRNO | CZECH REPUBLIC

NIH
National Institute on Aging

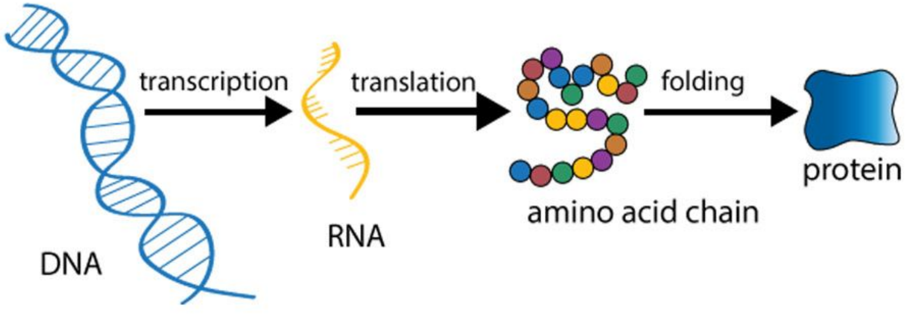
UNIVERSITAS MASARYKIANA BRUNENSIS

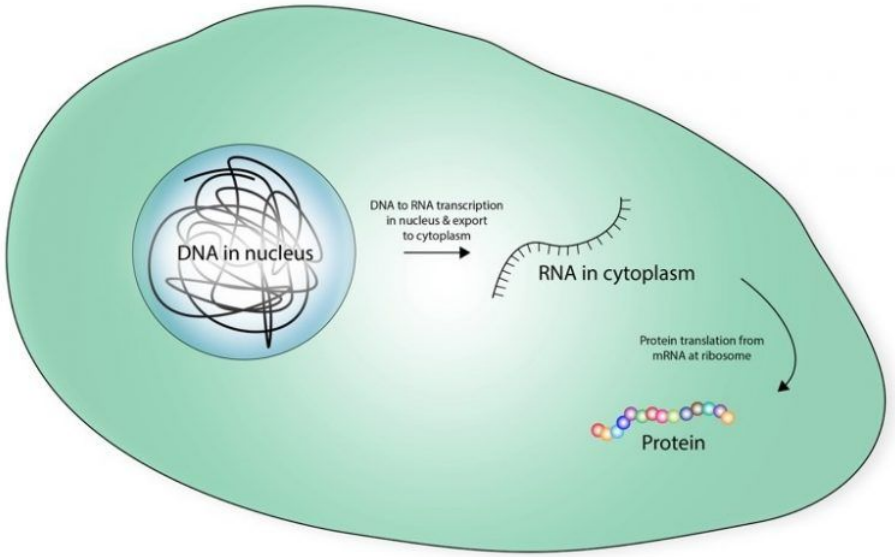
Vlastimil Martinek

Detecting RNA modification from nanopore signal

+ exercises

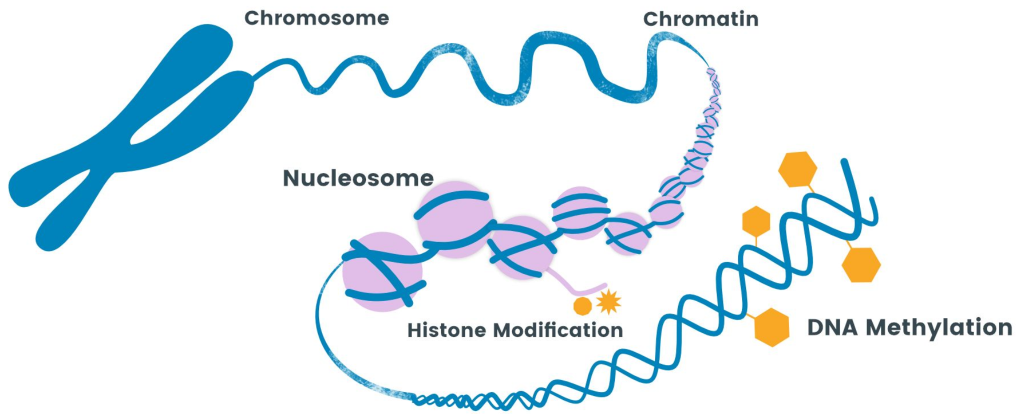
Central dogma of molecular biology - refresh



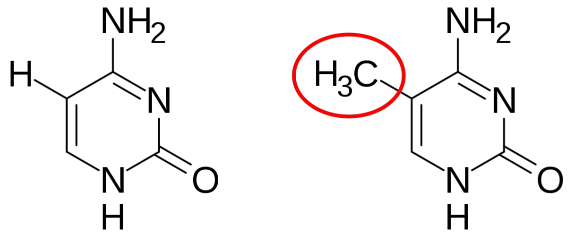

DNA in nucleus
DNA to RNA transcription in nucleus & export to cytoplasm
RNA in cytoplasm
Protein translation from mRNA at ribosome
Protein

Modifications

- Alter exposed DNA/RNA
- Change gene expression
- Can be a biological target
  - Immune system
  - Repair
- Multiple types of mods



ACCGTC <=> AC'C'GTC'

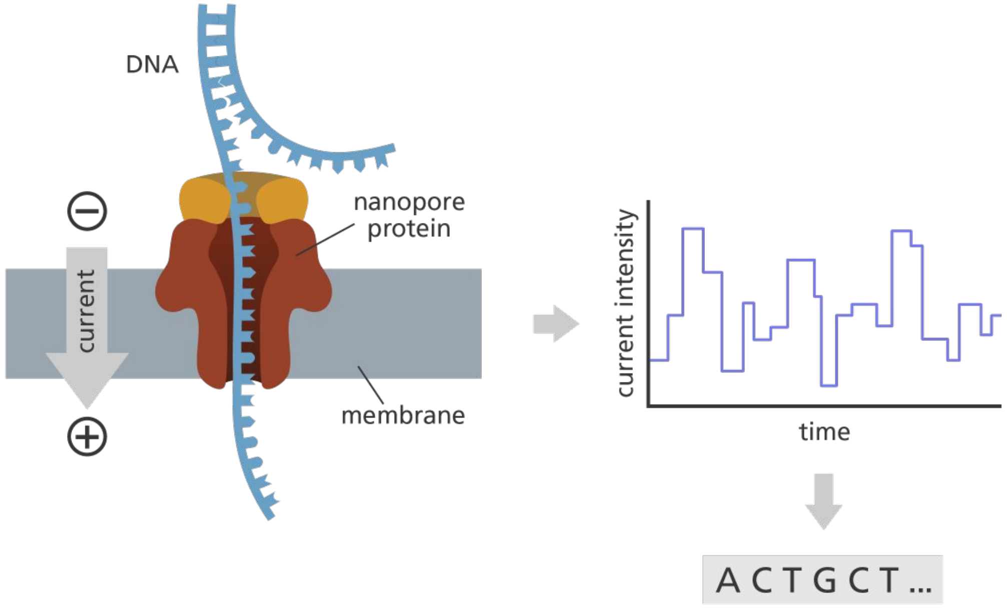


cytosine

methylated cytosine

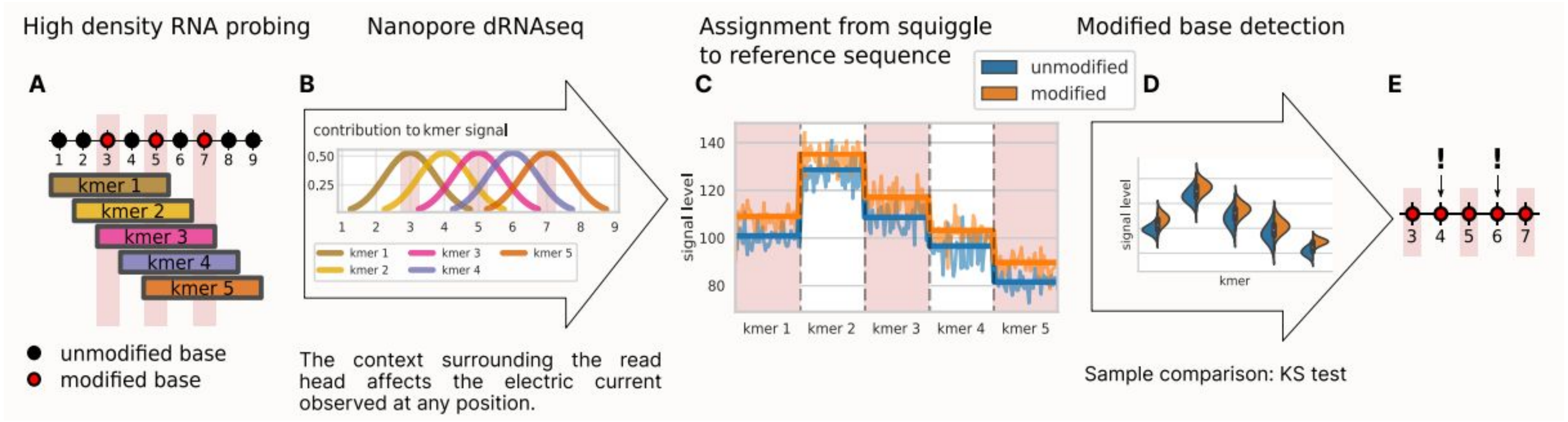
ONT (Oxford nanopore technologies)

- Sequencing to attain digital data
- Basecalling



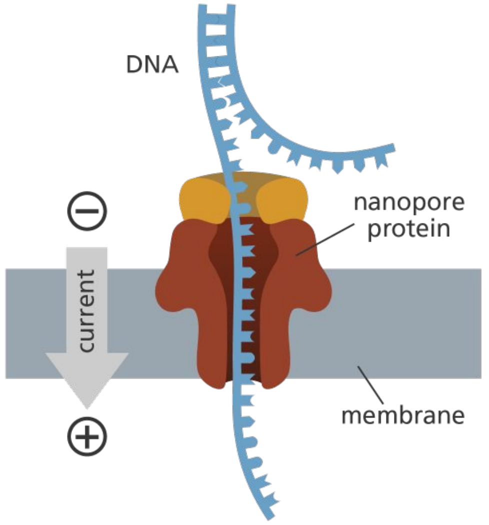
Existing methods

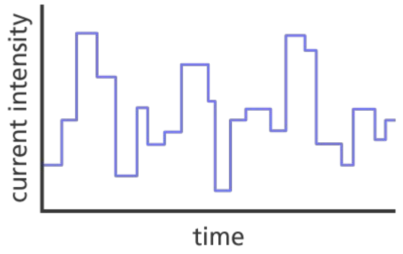
- Signal has mods information
- Need for a control sequence
- Dependent on basecaller



New approach

Detect modifications without reference





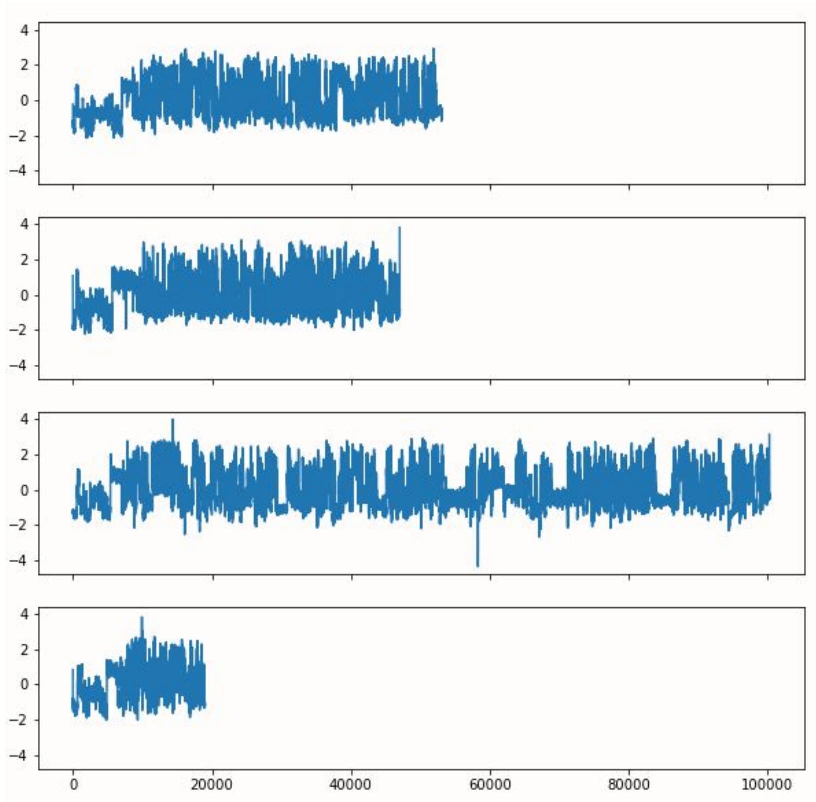
A C/T G C/T ...

OR

Modified / Not modified

Data

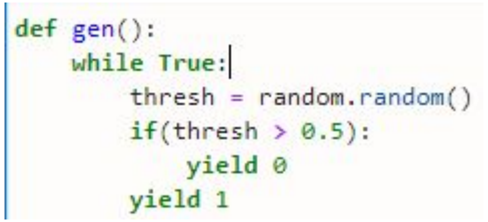
- 1D signal per sequence
- Variable length
- Two labels - modified/non-modified
- 1M positives, 2M negatives
- Preprocessing?



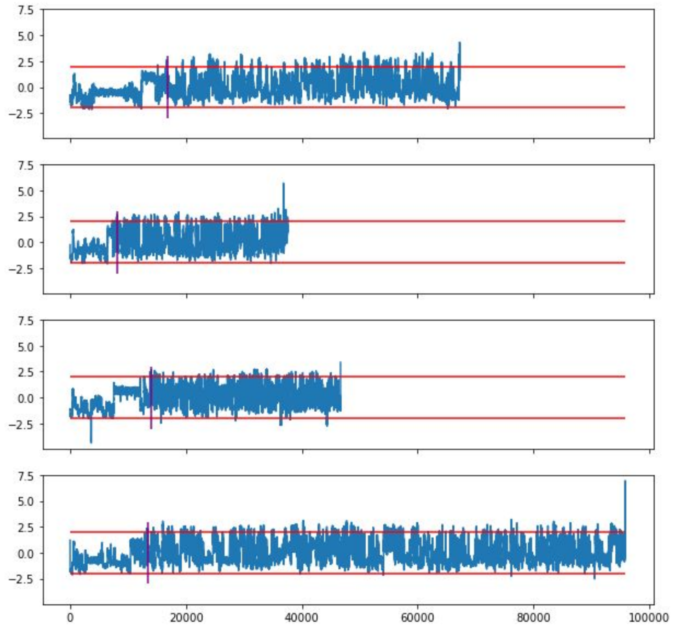
Data exploration and preparation

Preprocessing

- Cut primers
- Window
- Balanced sampling
- Standardized read-wise
- …?



```
def gen():
    while True:
        thresh = random.random()
        if(thresh > 0.5):
            yield 0
        yield 1
```



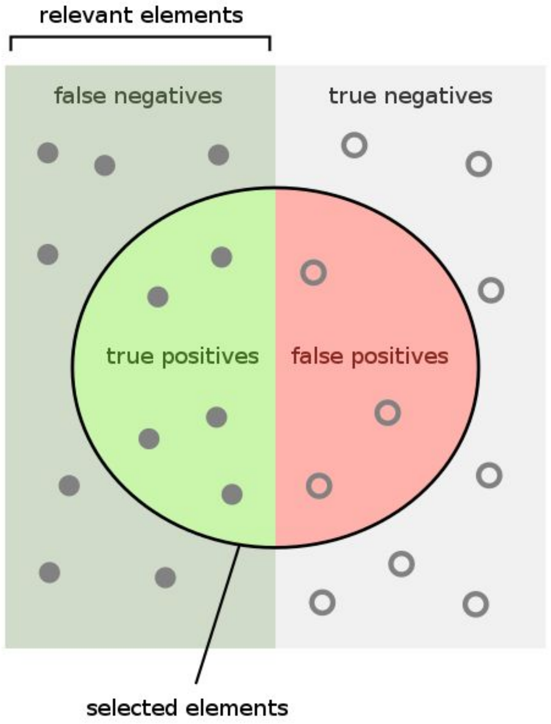
Noisy data labels

High false-positive rate (50-70%)

All negatives are correct

Affects metrics - perfect model 65-75%

How to tackle?



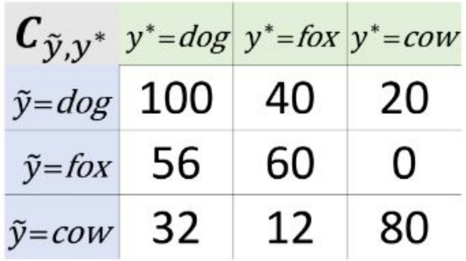
Label cleaning

Confident learning (google, mit, cleanlab)

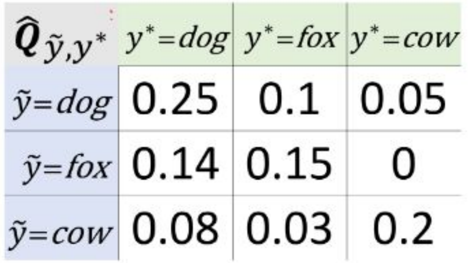
Input: Labels + predicted probabilities

Predicts label mistakes on validation data

(kfold needed to clean the dataset)



$C_{\tilde{y},y^*}$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	100	40	20
$\tilde{y}=fox$	56	60	0
$\tilde{y}=cow$	32	12	80



$\hat{Q}_{\tilde{y},y^*}$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	0.25	0.1	0.05
$\tilde{y}=fox$	0.14	0.15	0
$\tilde{y}=cow$	0.08	0.03	0.2

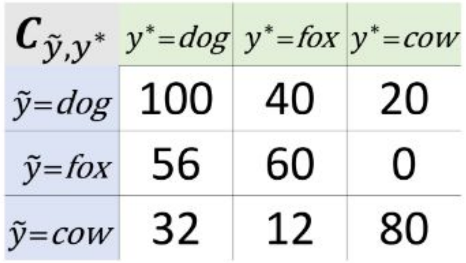
y~ current labels
y* true labels

From the matrix on the right in the figure, to estimate label issues:

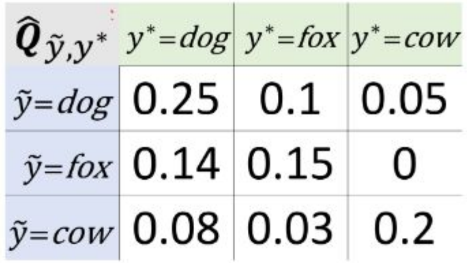
1. Multiply the joint distribution matrix by the number of examples. Let’s assume 100 examples in our dataset. So, by the figure above (*Q* matrix on the right), there are 10 images labeled *dog* that are actually images of *foxes*.
2. Mark the 10 images labeled dog with *largest* probability of belonging to class *fox* as label issues.
3. Repeat for all non-diagonal entries in the matrix

Confident learning

“The central idea is that when the predicted probability of an example is greater than a per-class-threshold, we *confidently count* that example as actually belonging to that threshold’s class. The thresholds for each class are the average predicted probability of examples in that class.”



$\boldsymbol{C}_{\tilde{y},y^*}$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	100	40	20
$\tilde{y}=fox$	56	60	0
$\tilde{y}=cow$	32	12	80

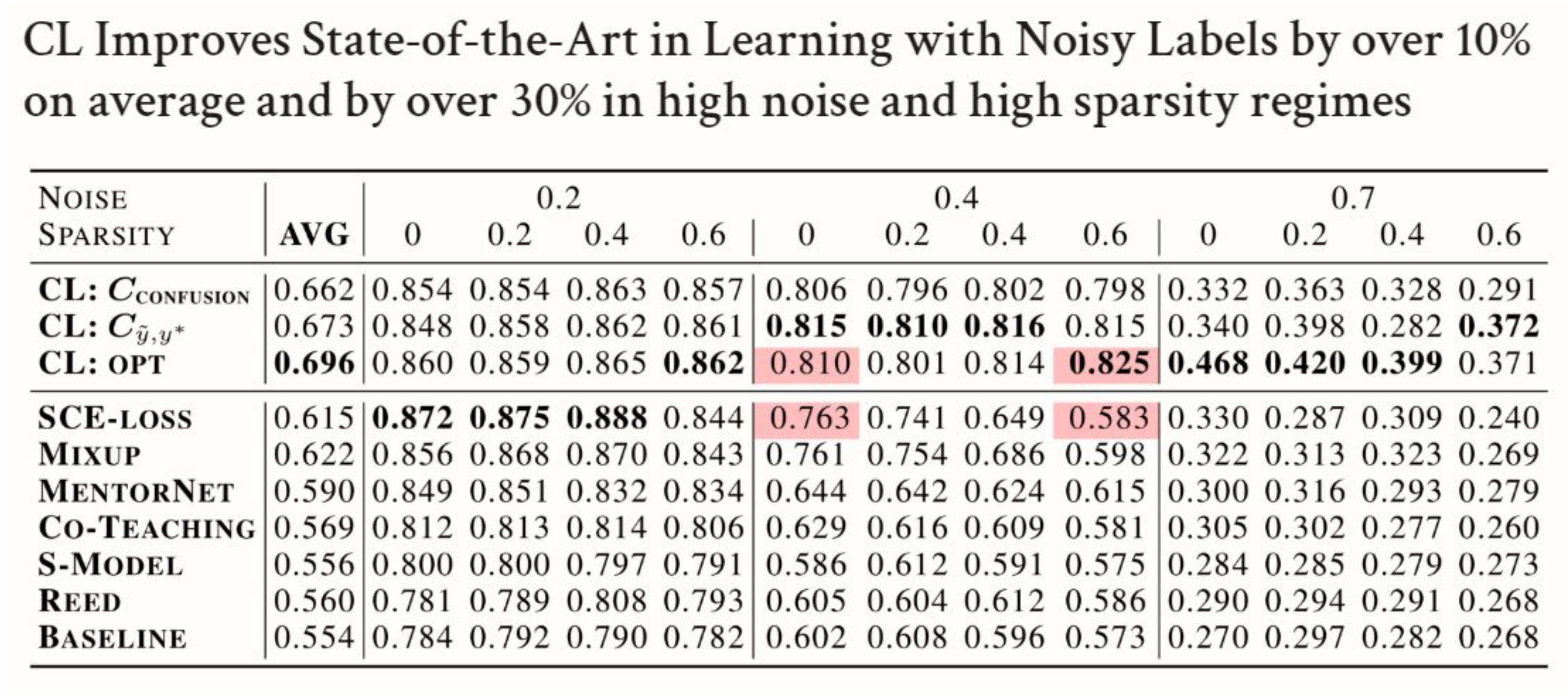


$\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$	$y^*=dog$	$y^*=fox$	$y^*=cow$
$\tilde{y}=dog$	0.25	0.1	0.05
$\tilde{y}=fox$	0.14	0.15	0
$\tilde{y}=cow$	0.08	0.03	0.2

y~ actual labels
y* true labels

Preliminary results = it works (positive labels marked)

Confident learning (CIFAR-10 acc)

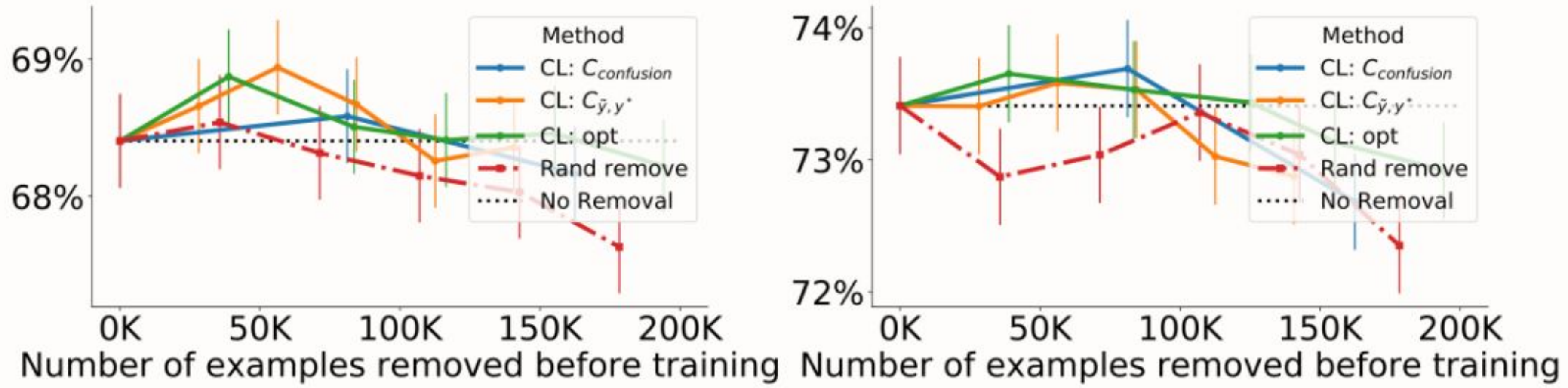


CL Improves State-of-the-Art in Learning with Noisy Labels by over 10% on average and by over 30% in high noise and high sparsity regimes

NOISE		0.2				0.4				0.7			
SPARSITY	**AVG**	0	0.2	0.4	0.6	0	0.2	0.4	0.6	0	0.2	0.4	0.6
CL: $C_{\text{CONFUSION}}$	0.662	0.854	0.854	0.863	0.857	0.806	0.796	0.802	0.798	0.332	0.363	0.328	0.291
CL: $C_{\tilde{y},y^*}$	0.673	0.848	0.858	0.862	0.861	**0.815**	**0.810**	**0.816**	0.815	0.340	0.398	0.282	**0.372**
CL: OPT	**0.696**	0.860	0.859	0.865	**0.862**	0.810	0.801	0.814	**0.825**	**0.468**	**0.420**	**0.399**	0.371
SCE-LOSS	0.615	**0.872**	**0.875**	**0.888**	0.844	0.763	0.741	0.649	0.583	0.330	0.287	0.309	0.240
MIXUP	0.622	0.856	0.868	0.870	0.843	0.761	0.754	0.686	0.598	0.322	0.313	0.323	0.269
MENTORNET	0.590	0.849	0.851	0.832	0.834	0.644	0.642	0.624	0.615	0.300	0.316	0.293	0.279
CO-TEACHING	0.569	0.812	0.813	0.814	0.806	0.629	0.616	0.609	0.581	0.305	0.302	0.277	0.260
S-MODEL	0.556	0.800	0.800	0.797	0.791	0.586	0.612	0.591	0.575	0.284	0.285	0.279	0.273
REED	0.560	0.781	0.789	0.808	0.793	0.605	0.604	0.612	0.586	0.290	0.294	0.291	0.268
BASELINE	0.554	0.784	0.792	0.790	0.782	0.602	0.608	0.596	0.573	0.270	0.297	0.282	0.268

Confident learning

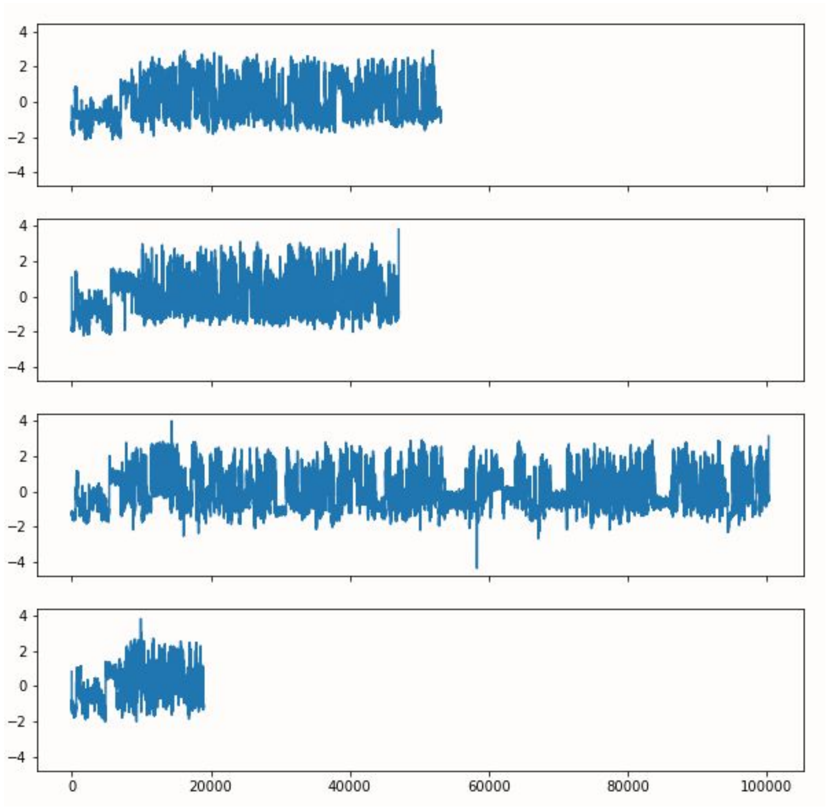
Training on ImageNet cleaned with CL Improves ResNet Test Accuracy



(a) ResNet18 Validation Accuracy

(b) ResNet50 Validation Accuracy

Model brainstorming

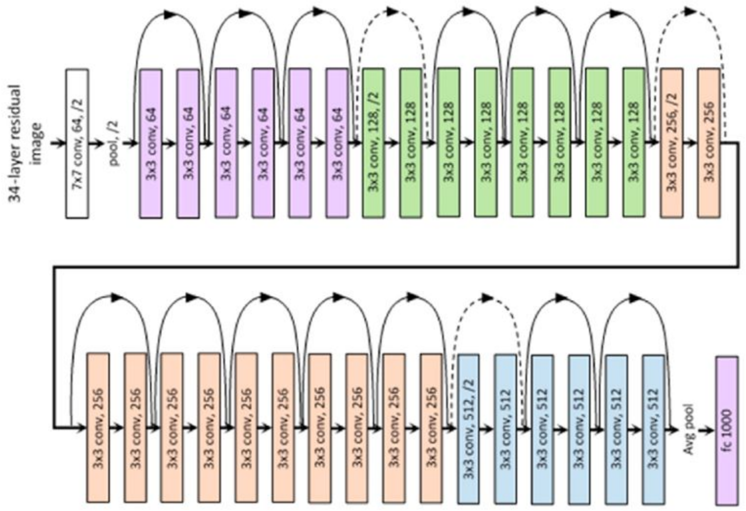


Initial architecture

1D resnet

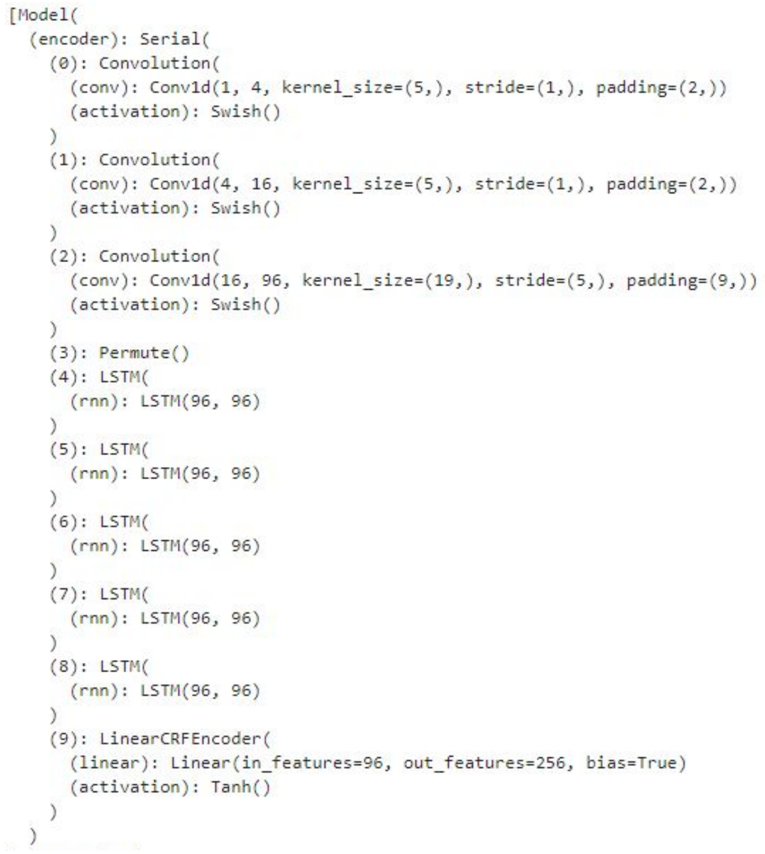
~60% accuracy

(a)



(b)

Transfer learning with basecallers (DNA)



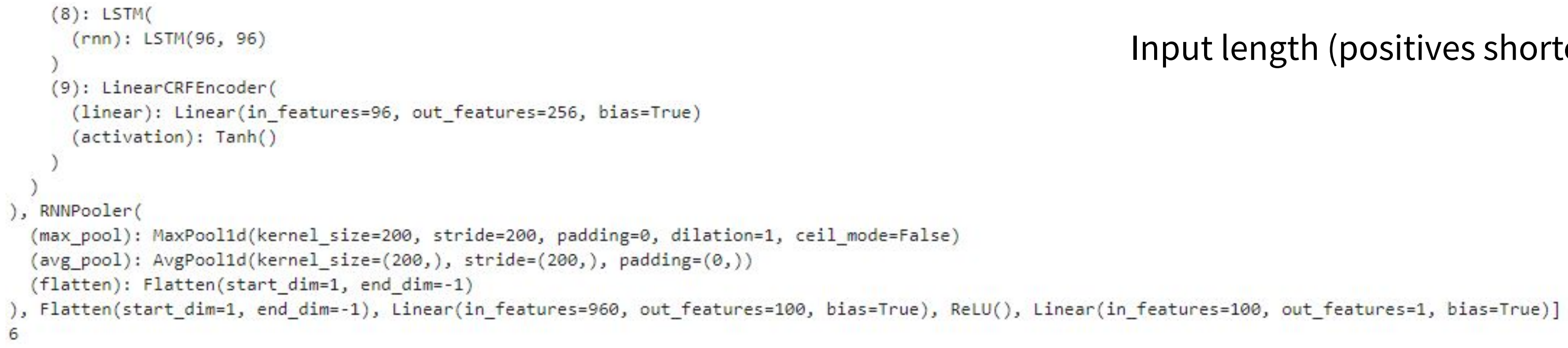
```
[Model(
  (encoder): Serial(
    (0): Convolution(
      (conv): Conv1d(1, 4, kernel_size=(5,), stride=(1,), padding=(2,))
      (activation): Swish()
    )
    (1): Convolution(
      (conv): Conv1d(4, 16, kernel_size=(5,), stride=(1,), padding=(2,))
      (activation): Swish()
    )
    (2): Convolution(
      (conv): Conv1d(16, 96, kernel_size=(19,), stride=(5,), padding=(9,))
      (activation): Swish()
    )
    (3): Permute()
    (4): LSTM(
      (rnn): LSTM(96, 96)
    )
    (5): LSTM(
      (rnn): LSTM(96, 96)
    )
    (6): LSTM(
      (rnn): LSTM(96, 96)
    )
    (7): LSTM(
      (rnn): LSTM(96, 96)
    )
    (8): LSTM(
      (rnn): LSTM(96, 96)
    )
    (9): LinearCRFEncoder(
      (linear): Linear(in_features=96, out_features=256, bias=True)
      (activation): Tanh()
    )
  )
```

Custom head
Training

Transfer learning with basecallers (DNA)

LR differences between modules
LR warmup
Accuracy ~72%

Input length (positives shorter)



```
    (8): LSTM(
      (rnn): LSTM(96, 96)
    )
    (9): LinearCRFEncoder(
      (linear): Linear(in_features=96, out_features=256, bias=True)
      (activation): Tanh()
    )
  )
), RNNPooler(
  (max_pool): MaxPool1d(kernel_size=200, stride=200, padding=0, dilation=1, ceil_mode=False)
  (avg_pool): AvgPool1d(kernel_size=(200,), stride=(200,), padding=(0,))
  (flatten): Flatten(start_dim=1, end_dim=-1)
), Flatten(start_dim=1, end_dim=-1), Linear(in_features=960, out_features=100, bias=True), ReLU(), Linear(in_features=100, out_features=1, bias=True)]
6
```

Future

Transformer + limited attention

RNA basecaller

Interpretability + basecalling info

Learning with cleaned labels

Speedup parallelization

Experiment differences

Use statistical modification methods

Base-wise labeling

Sources

https://inno-forum.org/single-cell-rna-sequencing-technique-come-age/

https://www.labclinics.com/2018/11/08/role-dna-methylation-disease/?lang=en

https://www.yourgenome.org/facts/what-is-oxford-nanopore-technology-ont-sequencing/