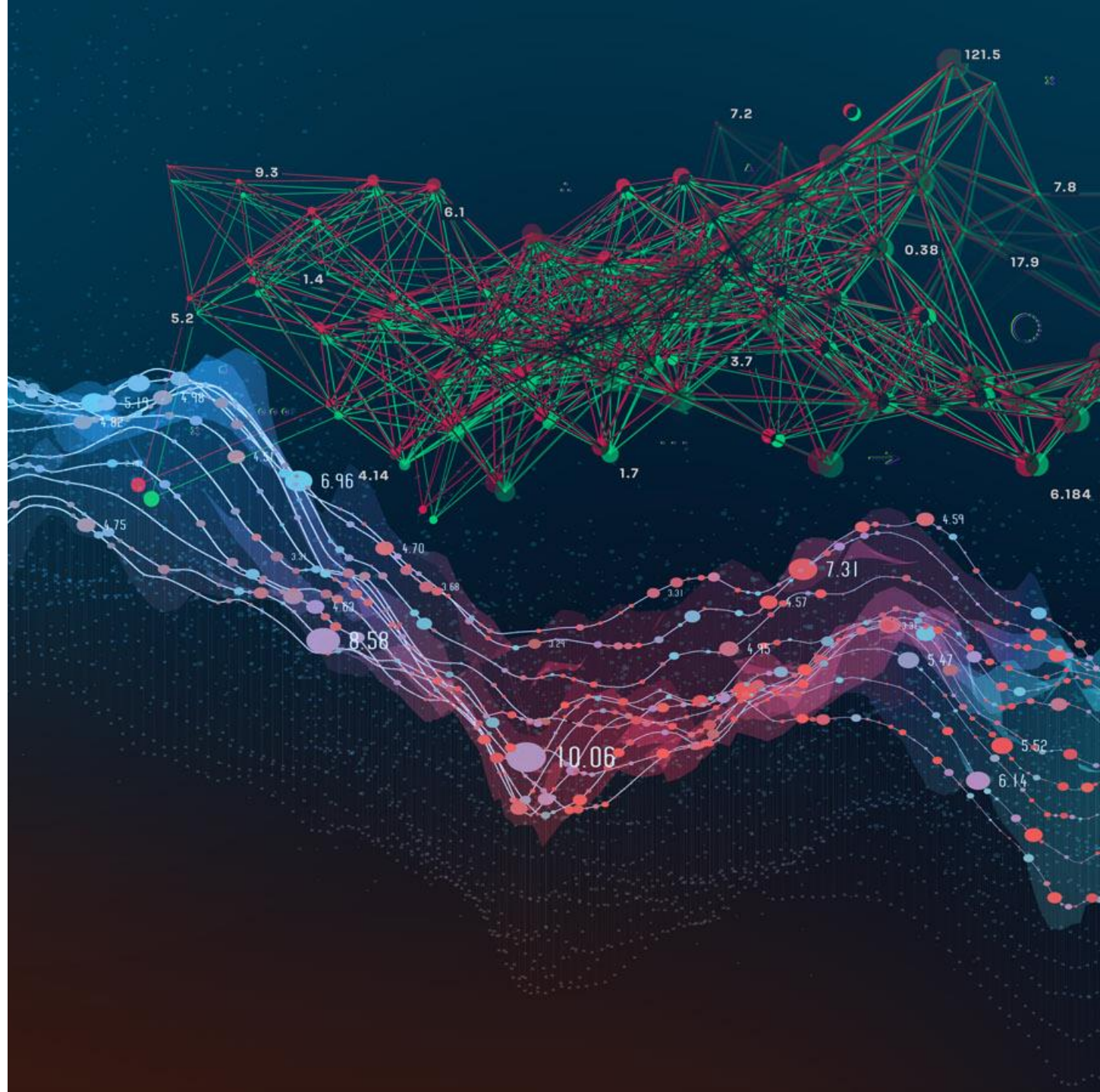


# Day 3 – A Variety of Few- shot Modalities



# Day 1 and 2 reminders

- Day 1
  - Intro to few-shot learning
  - Applying few-shot to new data
- Day 2
  - Training few-shot models
  - Tips and tricks
  - Few-shot model types
    - Metric-based
    - Meta-learners
    - Conditioned models

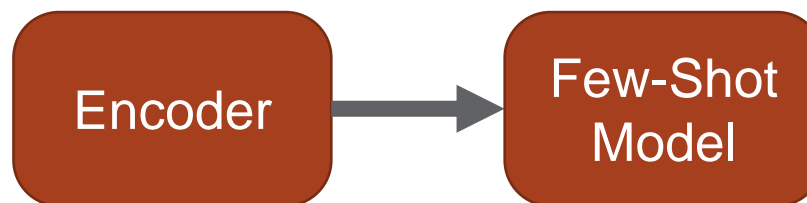




Questions from  
day 1 and 2?

# Encoders vs. Few-Shot Models

- Encoder (e.g. ResNet-50, VGGish, BERT)
  - Modality specific
  - Converts raw data to numeric vector
- Few-Shot Model (e.g. ProtoNets, RCNet)
  - Combines support and query set
  - Makes classification decisions



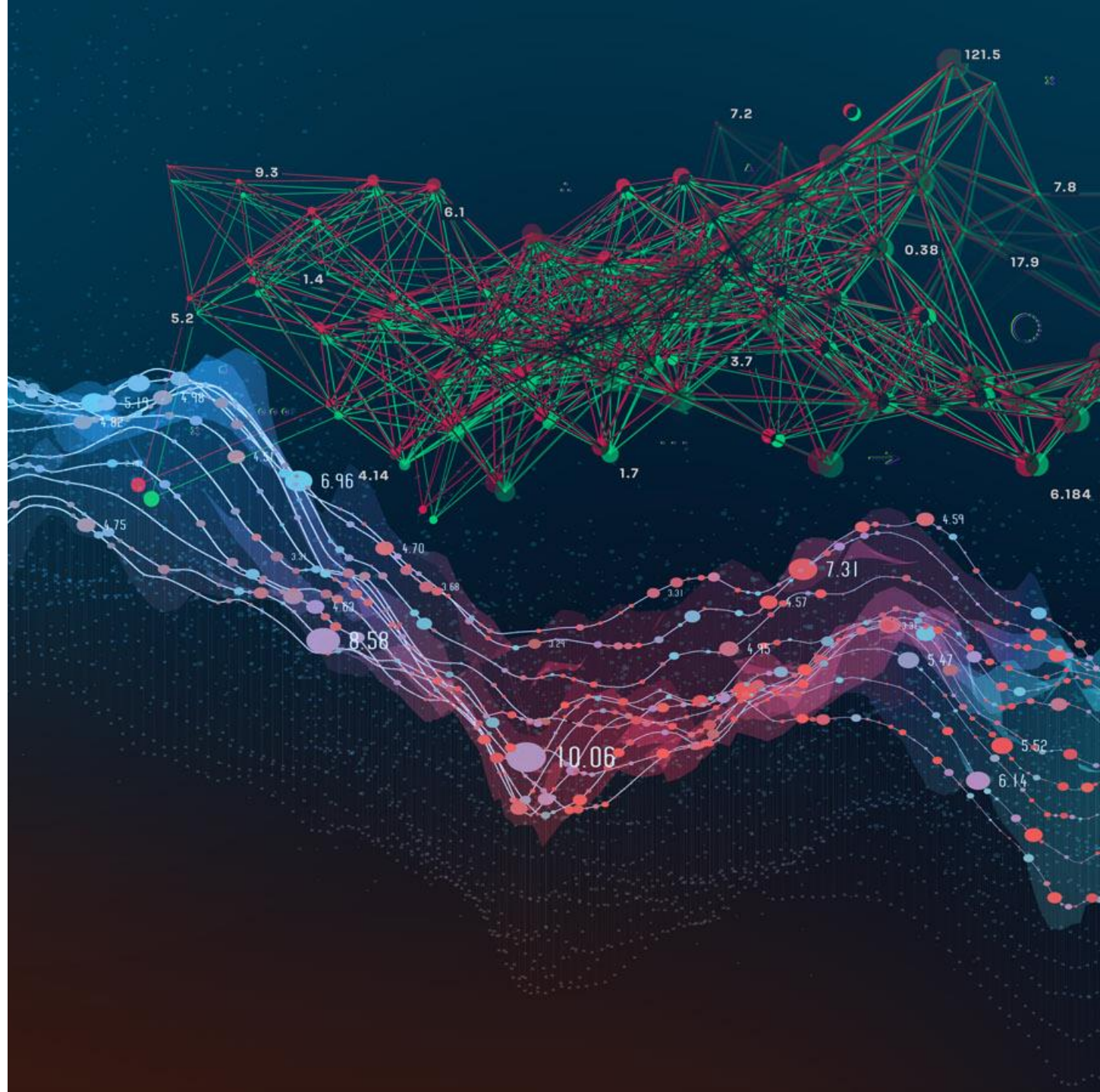
- To apply few-shot learning we only need:
  - Sufficient number of classes to train on
  - An encoder that converts raw data to vectors

# Applying to a new modality

- This allows us to apply few-shot in new modalities. And (minimally) requires us to modify the following:
- Data
  - Code to load individual datapoints
  - Any preprocessing functionality
- Encoder
  - An encoder architecture
  - Preprocessing functionality should match encoder's expected input

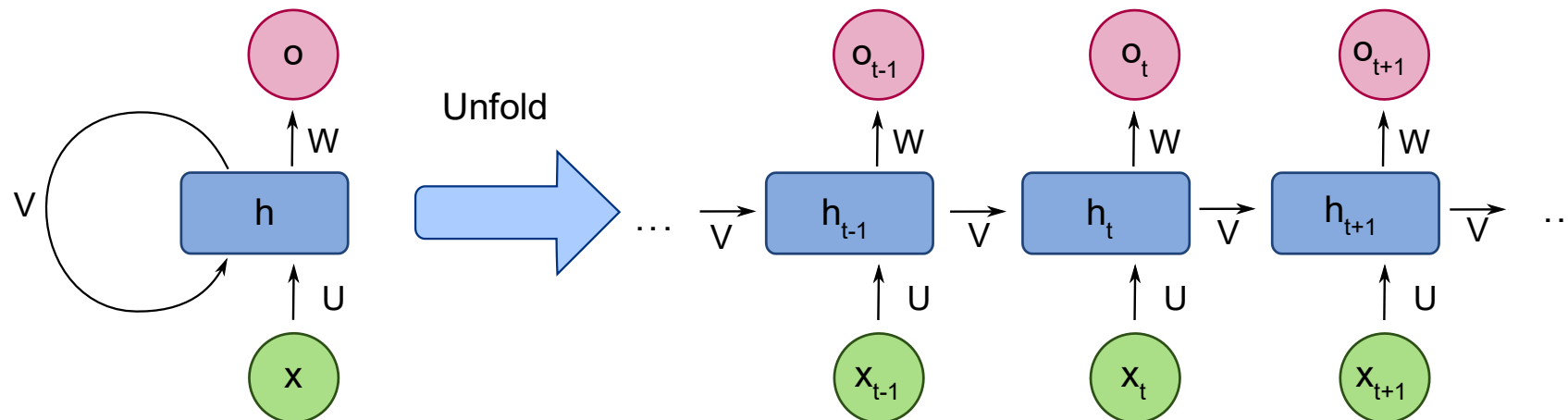


# Text Few-shot learning



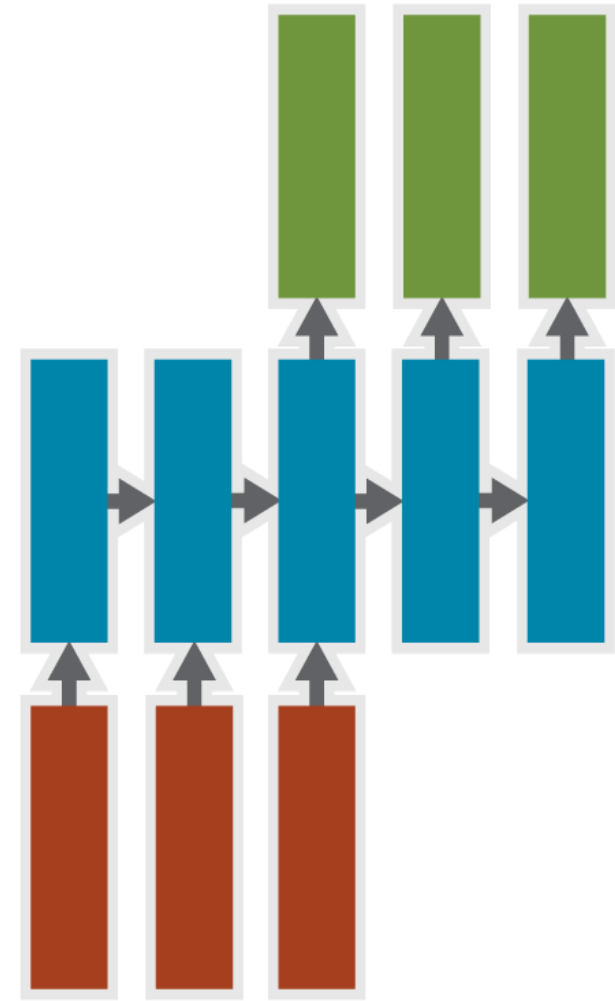
# Recurrent Neural Networks

- Fully connected networks with a hidden state
  - Incorporates information from previous timesteps to maintain context through time.
- Many different variations
  - Simple Recurrent Neural Network
  - Long Short-Term Memory
  - Gated Recurrent Unit



# Sequence → Sequence

- Seq2Seq
  - Predict future events from past sequence
  - Incredibly powerful with *attention mechanisms*
- Example: Machine Translation
  - Original sentence → Translated sentence
  - Paraphrase generation
  - Dialogue generation
- Example: Video Forecasting
  - Beginning of video → Predict what follows





# Transformers

- Traditional text models use Recurrent Neural Networks (RNNs)
  - Process text as a sequence
- Transformers are an ***attention-based*** model
  - First introduced in 2017
- Unprecedented text generation quality  
<https://transformer.huggingface.co/>

How should I explain what a transformer is to my colleagues? They're very interested in new developments to Natural Language Processing tasks!

You can also explain things very simply using mathematical notation. (It's not necessary.)

Transformers are used in a variety of fields. (I'm a professional mathematician myself.) These two are examples of what transformer is in other domains of science, including:

In the field of mathematics, we use many different mathematical languages to describe our transformations of numbers into symbols. For instance, to get a symbol from one number to another, we use the "add" function (the number of numbers you add up to get one number). And to get the same symbol, we use the "subtract" function.

You can also say that a transformer can produce a symbol based on a particular context, a category, or on the input to the transformer. For example, we can call a transformer any function that takes two variables,  $a$  and  $b$ , and applies a function  $f$  to the first variable only. Thus, the following is a function that takes two variables and applies one function  $g$  to the first and the second

# Transformers

- Not just for improving text generation
- State-of-the-art performance in almost all standard NLP tasks
- SQUAD 2.0 leaderboard
  - Question answering

Pre-transformers

Rank	Model	EM	F1
	Human Performance Stanford University (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Apr 06, 2020	SA-Net on Albert (ensemble) QIANXIN	90.724	93.011
2 Apr 05, 2020	Retro-Reader (ensemble) Shanghai Jiao Tong University <a href="http://arxiv.org/abs/2001.09694">http://arxiv.org/abs/2001.09694</a>	90.578	92.978
3 Mar 12, 2020	ALBERT + DAAF + Verifier (ensemble) PINGAN Omni-Sinitic	90.386	92.777
4 Jan 10, 2020	Retro-Reader on ALBERT (ensemble) Shanghai Jiao Tong University <a href="http://arxiv.org/abs/2001.09694">http://arxiv.org/abs/2001.09694</a>	90.115	92.580
5 Nov 06, 2019	ALBERT + DAAF + Verifier (ensemble) PINGAN Omni-Sinitic	90.002	92.425
6 Sep 18, 2019	ALBERT (ensemble model) Google Research & TTIC <a href="https://arxiv.org/abs/1909.11942">https://arxiv.org/abs/1909.11942</a>	89.731	92.215

58 Sep 17, 2018	Unet (ensemble) Fudan University & Liulishuo Lab <a href="https://arxiv.org/abs/1810.06638">https://arxiv.org/abs/1810.06638</a>	71.417	74.869
--------------------	--	--------	--------

SQuAD 2.0 leaderboard (4/27/2020)

# Self-Attention Mechanisms

- Allows model to directly connect words from the full sequence, ignoring distance

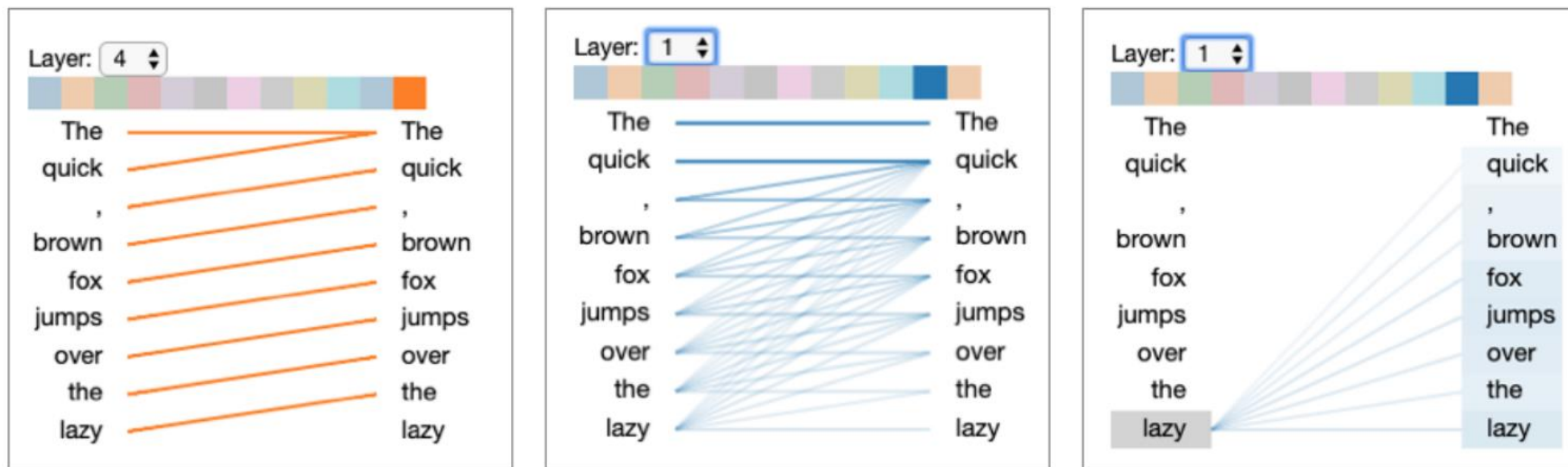


Figure 1: Attention-head view for GPT-2, for the input text *The quick, brown fox jumps over the lazy*. The left and center figures represent different layers / attention heads. The right figure depicts the same layer/head as the center figure, but with the token *lazy* selected.

<https://arxiv.org/pdf/1906.05714.pdf>



# Attention Mechanisms

- Attention can act as a window into what a model is “thinking”
- The weights that the attention mechanism produces show us how important each value is to the network at that time

Figure 3. Examples of attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word)



A woman is throwing a frisbee in a park.



A dog is standing on a hardwood floor.



A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.



A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

# Text Few-shot learning

- Few-shot generally relies on training with many classes (ideally  $>100$ )
  - But many NLP tasks focus on small sets of classes
- Open questions
  - Can we leverage pre-trained transformers as an encoder?
  - When talking about 5-shot performance, is 5 examples of text enough? Or should we expect worse performance?

# Few-shot Authorship Attribution

- Authorship attribution
  - Many potential classes (i.e. authors)
  - Very challenging problem requiring large data...
  - Could few-shot even be applied?
- Goals:
  - Assess viability
  - How much data is enough?
  - What techniques from traditional authorship ID can be applied (e.g. stylometrics?)



# Reddit Comment Authorship



- Reddit is a social media website where users can comment on posts
  - One month of comment history collected from pushshift.io
- From the most-commented 10K subreddits, identified top 10K users
  - Removed known bot accounts (AutoModerator, \*\*\*\*Bot)

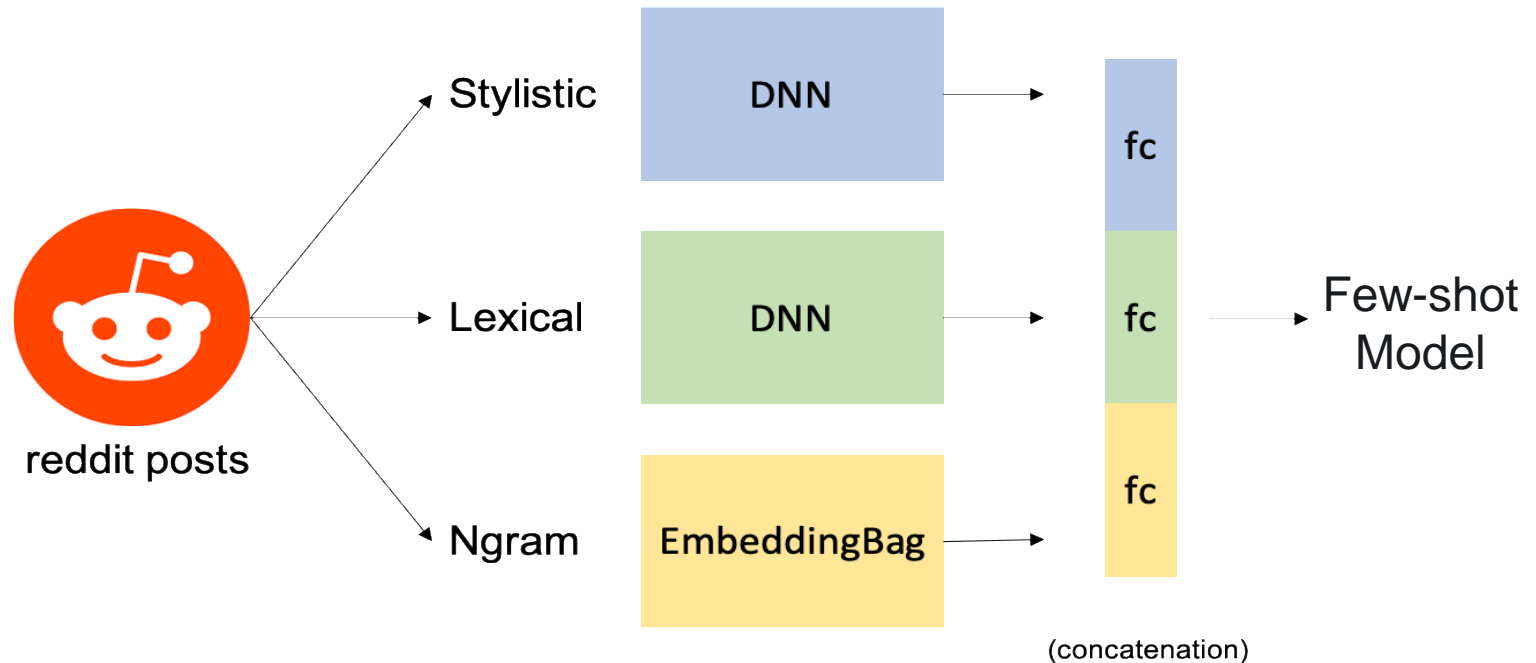
	Train	Val	Test
# authors	7977	917	1007
Avg # posts/author	1018	917	1003
Avg # chars/post	175	147	165

# Stylometrics

- Features intended to capture how a person writes, rather than the topics they choose to write about
- **Stylistic**
  - Punctuation usage, sentence length, upper/lowercase, whitespace
- **Character n-grams**
  - 10,000 features, computed over training dataset
- **Syntactic**
  - POS tags, POS n-grams calculated with Tweet NLP (Owoputi et al., 2013)
- **Lexical**
  - Normalized function word counts

# Text feature encoders

- Late Fusion concatenates separate feature types
  - N-gram features are represented as embeddings
  - Non-n-gram features processed through 4-layer MLP with ReLU activations





## Results – Stylometric comparison

- 5-shot, 5-way
  - Choose from 5 different authors
  - 5 comments per author
- Of individual features, DistilBERT performs best, but may be focusing on “topic” rather than identity
- Stylistic features perform best in combination

Feature Types	Test Acc.
Stylistic	52.4%
N-grams	33.4%
Syntactic	47.1%
Lexical	31.2%
DistilBERT	<b>70.6%</b>
Style + Lex	52.3%
Style + Lex + Syn	58.2%
All Stylometrics	57.4%
All + DistilBERT	<b>72.2%</b>

## Results – Increasing $k$ -shot

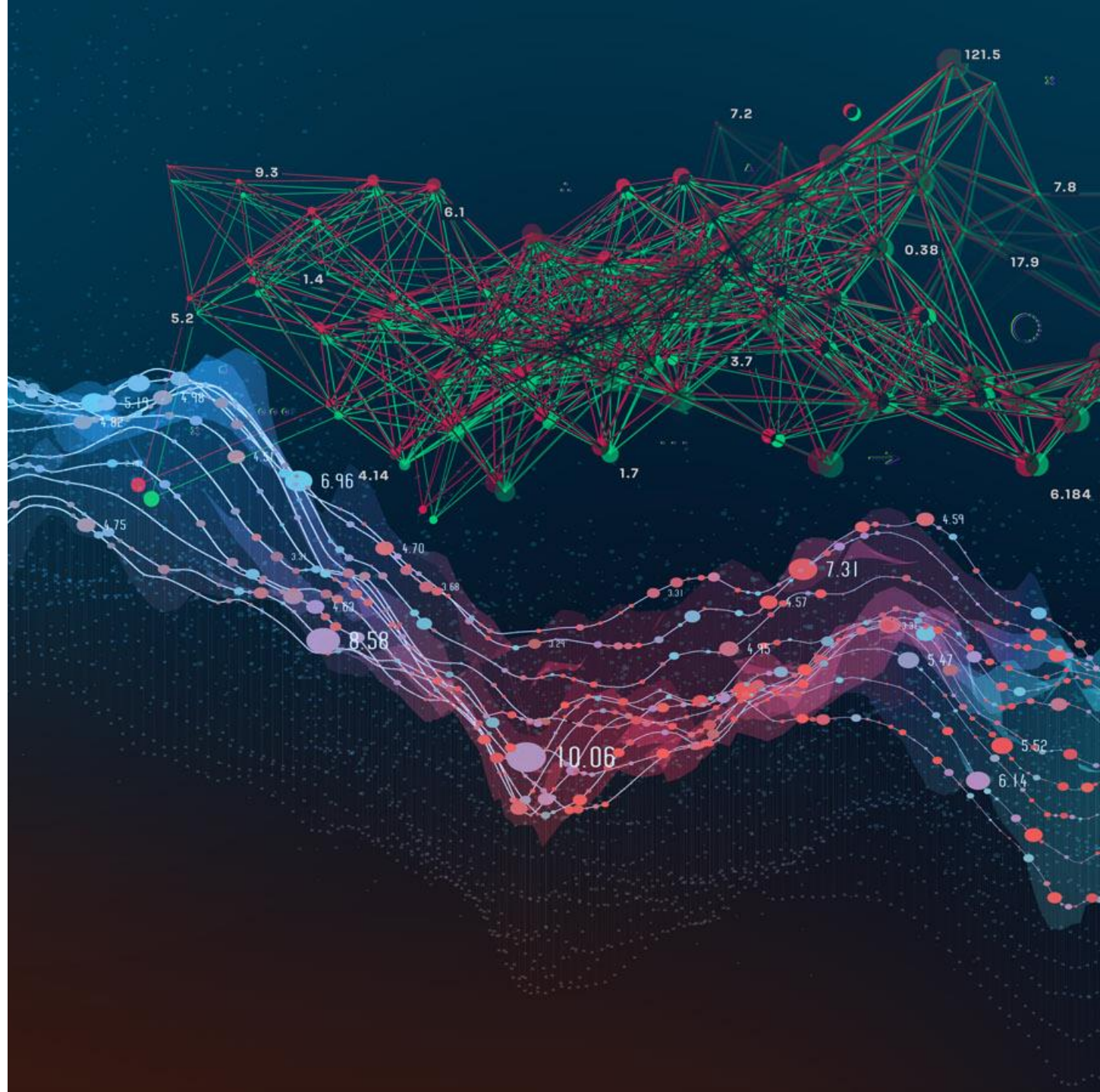
- How does performance scale with increased data?
  - Slight improvements (3-4%pts) for all models

	5-shot	10-shot	20-shot
DistilBERT	70.01%	72.19%	74.13%
All Stylometrics	43.10%	44.41%	46.20%
All + DistilBERT	70.19%	72.32%	74.15%

- Is DistilBERT just focusing on topic?
  - Easy set is 200 users who comment on least diverse set of subreddits
  - Hard set is 200 users who comment on most diverse set of subreddits

	Easy	Hard
DistilBERT	83.2%	61.3%
All Stylometrics	67.3%	52.0%
All + DistilBERT	83.8%	61.8%

# Video Few-shot learning





# Video Few-shot learning

- A few preprocessing considerations.
  - Target framerate?
  - Target height and width?
  - Optical flow?
- Videos can quickly consume too much memory.
  - Low framerates (1-5 fps) can be helpful concessions to make.
  - Loading entire episodes of data can be time-consuming.
  - Necessitates loading preprocessed data (at least for training)
- Once loaded, the goal is to classify query videos (frame sequences).

# Few-Shot Kinetics Data



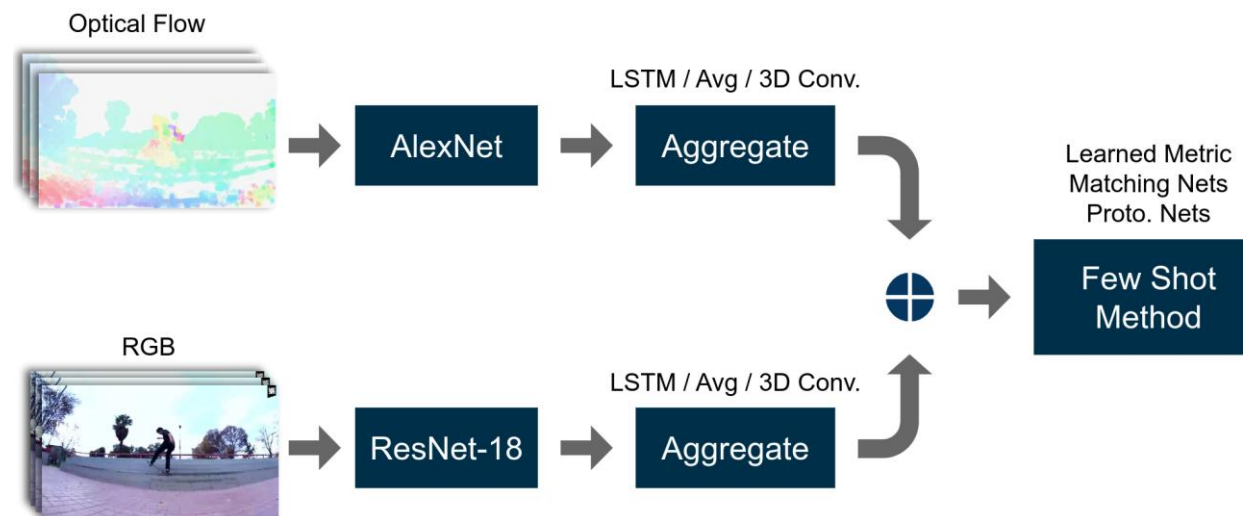
- Subset of the Kinetics dataset (10-second YouTube clips)
- Kinetics 600
  - 600 actions, at least 600 examples per action
  - Provides more examples / allows for “challenge set”
    - General test: standard, randomly sampled classes
    - Challenge test: “aggressive action” set (wrestling, headbutting, smashing, tackling, etc.)
- Kinetics 400
  - Few-shot splits from (Zhu and Yang, 2018)
  - Less data, no challenge set.
  - Allows for comparison to state-of-the-art.
- Extract RGB and optical flow.



Illustration of optical flow features from <https://ghassanalregib.com/>.

# Video Model Architecture

- CNN Architectures:
  - AlexNet, ResNet18, ResNet34
  - Applied to each frame of the video.
- Aggregation:
  - Avg. Pooling, LSTM, 3D-Convolutions.
  - Combine frame embeddings over time.
- Few-shot method:
  - Prototypical Nets, Matching Nets, Relation Net.
  - All simple and similar, ProtoNets outperforms others.
- “Two-stream” when using RGB + Flow, “one-stream” using just RGB



# Experimental Results

## Kinetics 600

Input	Encoder	General Test	Challenge Test
RGB (1 fps)	One-Stream, LSTM	82.7	58.0
RGB + Flow	Two-Stream, LSTM	84.2	59.4

After a thorough model search, LSTM or averaging, with prototypical nets yield best results.

Flow improves performance by 2%.

Challenge test set involves non-random split

- Train on "non-violent" actions.
- Test on "violent" actions

## Kinetics 400

Method	CNN	Acc
Baseline: CMN	RN-50	78.9
Baseline: TAM (SOTA)	RN-50	85.8
Two-Stream + LSTM	RN-34 + RN-18	78.6
Two-Stream + LSTM (PT)	RN-34 + RN-18	86.7

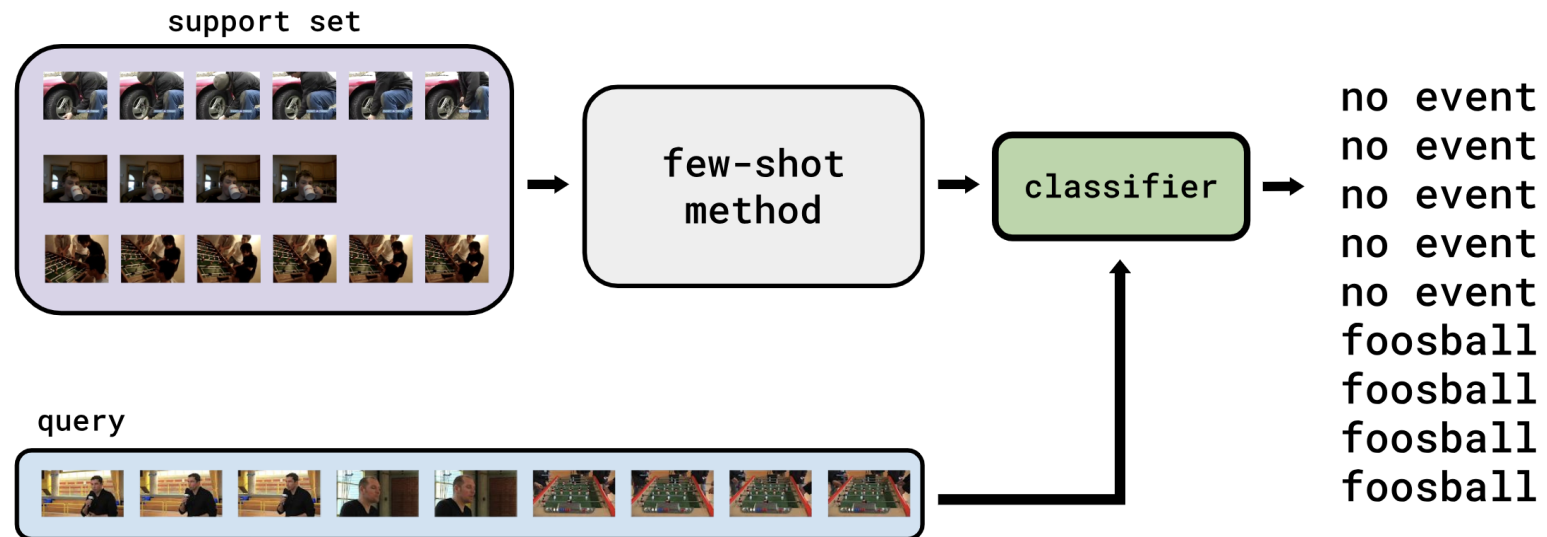
More difficult dataset, due to less data.

Pre-training yields substantial boost.



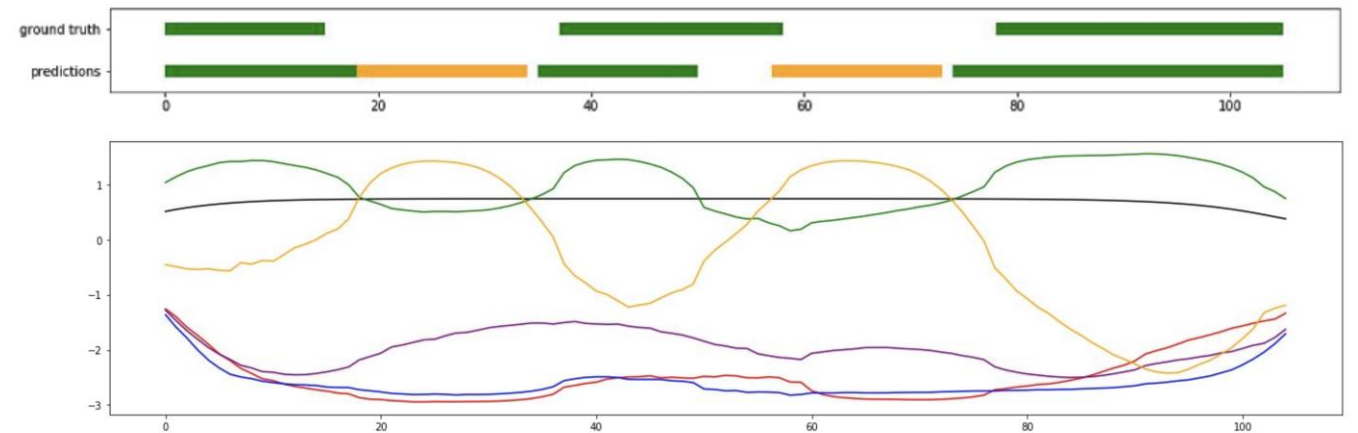
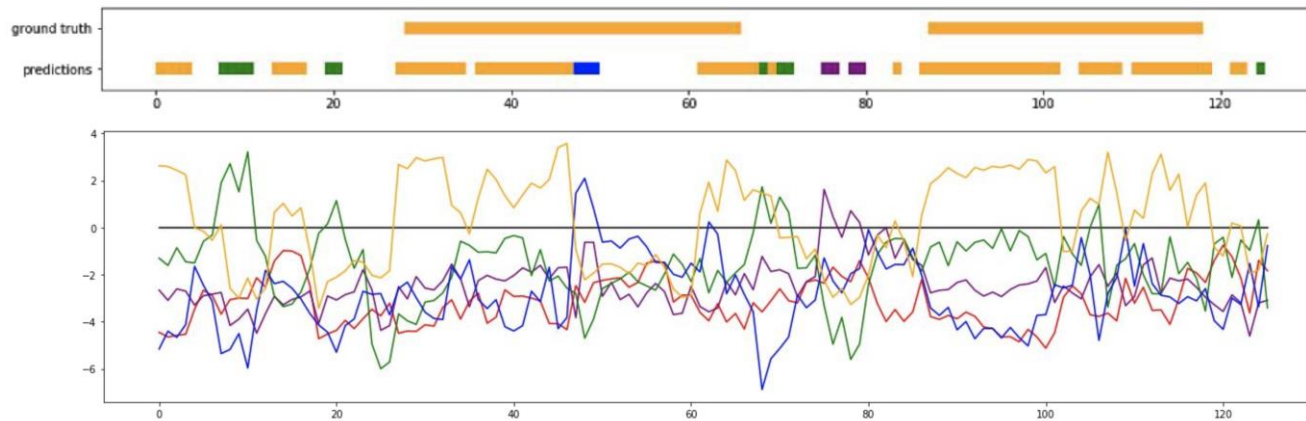
# Localization

- Moving beyond short videos, can we identify where in a sequence an action is taking place?
  - Window-based prediction cuts long sequence into manageable chunks





# Localization

- Challenge is predictions can be very noisy, so smoothing is required



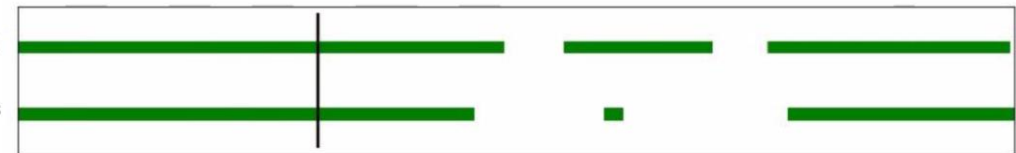
# Video - Localization



volleyball =   
crunches = 

ground truth

predictions



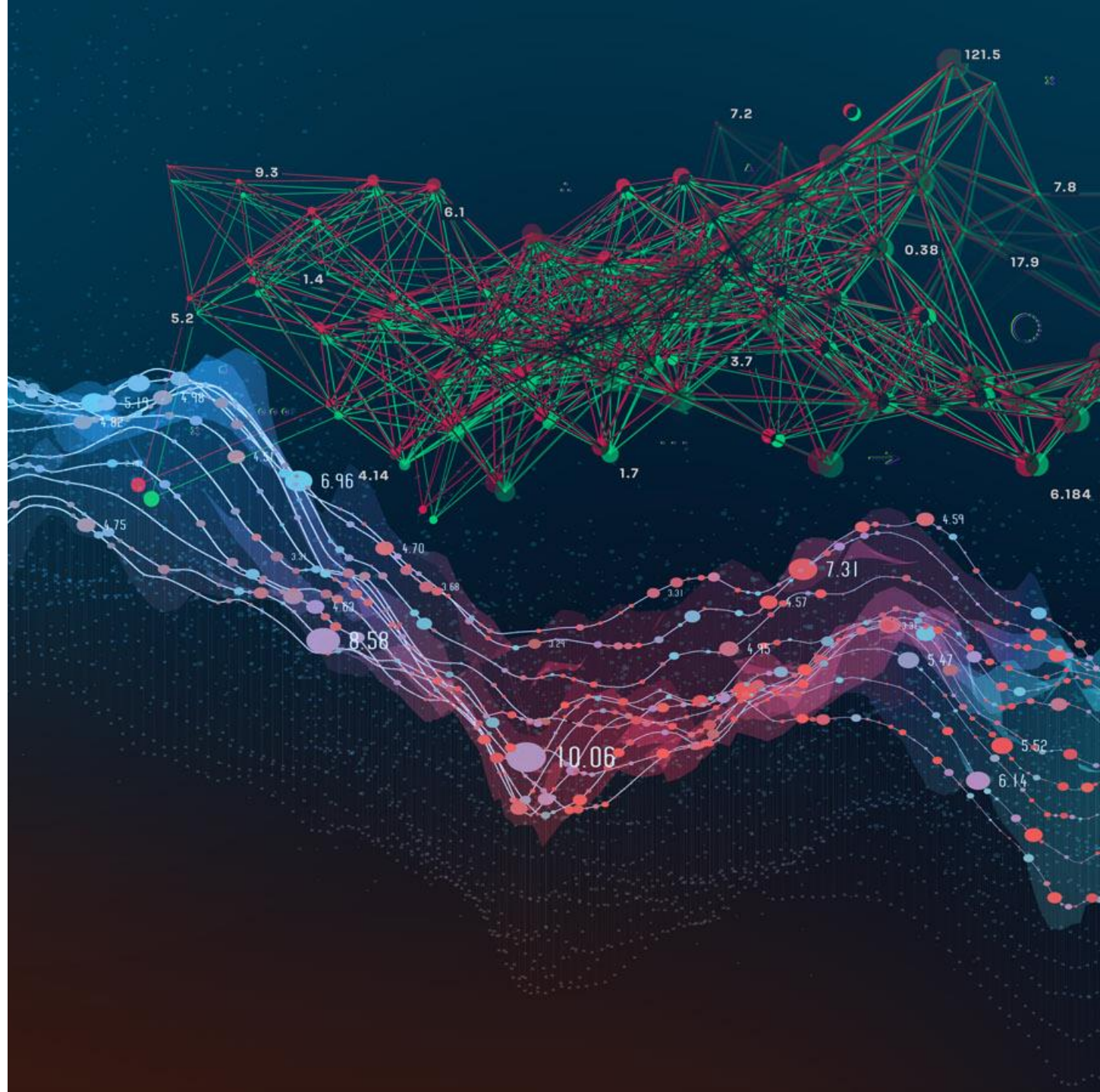
ground truth

predictions





# Audio Few-shot learning



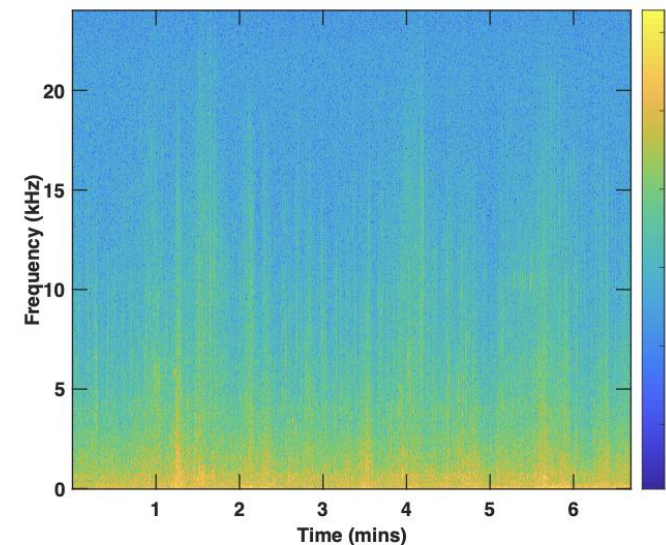
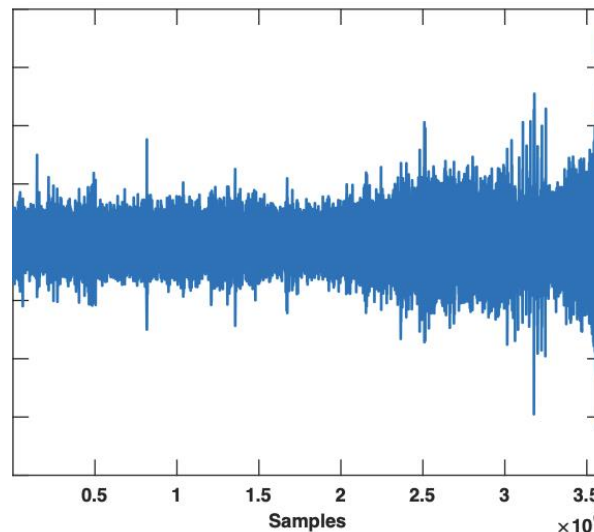


# Audio Few-Shot Learning

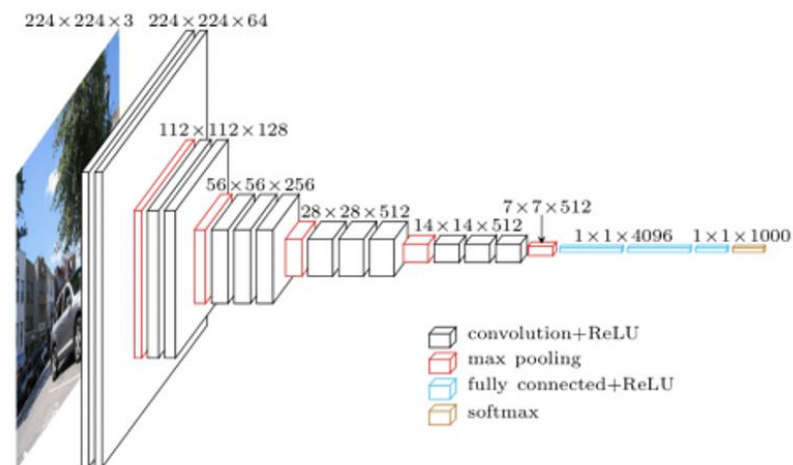
- Multiple parameters for preprocessing audio.
  - Sample rate?
  - Length of processing windows?
  - How much do processing windows overlap?
- Might need to chunk up large audio sequences to save memory.
  - For short clips (~30 sec), we can frequently process them all at once.
- Once processed, audio and video can be handled classified in similar fashion
  - E.g. Localization

# Audio Few-Shot Learning

- Processing audio clips relies on transforming raw signals into Mel-Spectrogram representations.
- CNNs then operate on the spectrogram images.



Amplitude and spectrogram of conversation on top of a street scene (Holland)



Macro-architecture of the VGG16 Network

## Audio – Speaker Identification

- VoxCeleb speaker identification
  - Near 80% accuracy from a single 30-second clip

1-shot / 5-way	5-shot / 5-way
79.9%	93.5%

- Without any additional training, up to 66% accuracy on ICSI meeting corpus
  - Long audio clips with multiple speakers

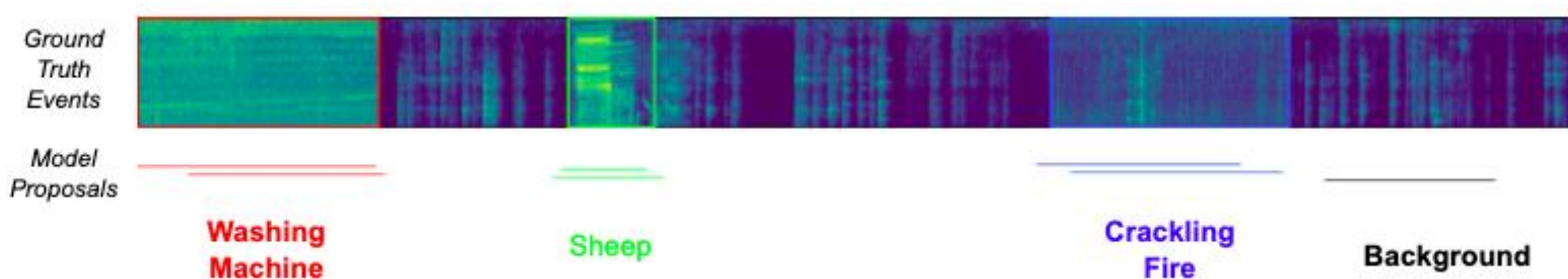
# speakers	4	5	6	7	8	9
Avg. Acc.	66.4%	54.0%	56.5%	53.1%	59.3%	36.9%

## Audio – Audio Event Detection

- Kinetics: Same as video experiments, but relying solely on audio clues

1-shot / 5-way	5-shot / 5-way
35.3%	51.5%

- HACS temporal action localization
  - Identify when action is taking place based on audio
  - 0.4 mean average precision (mAP)



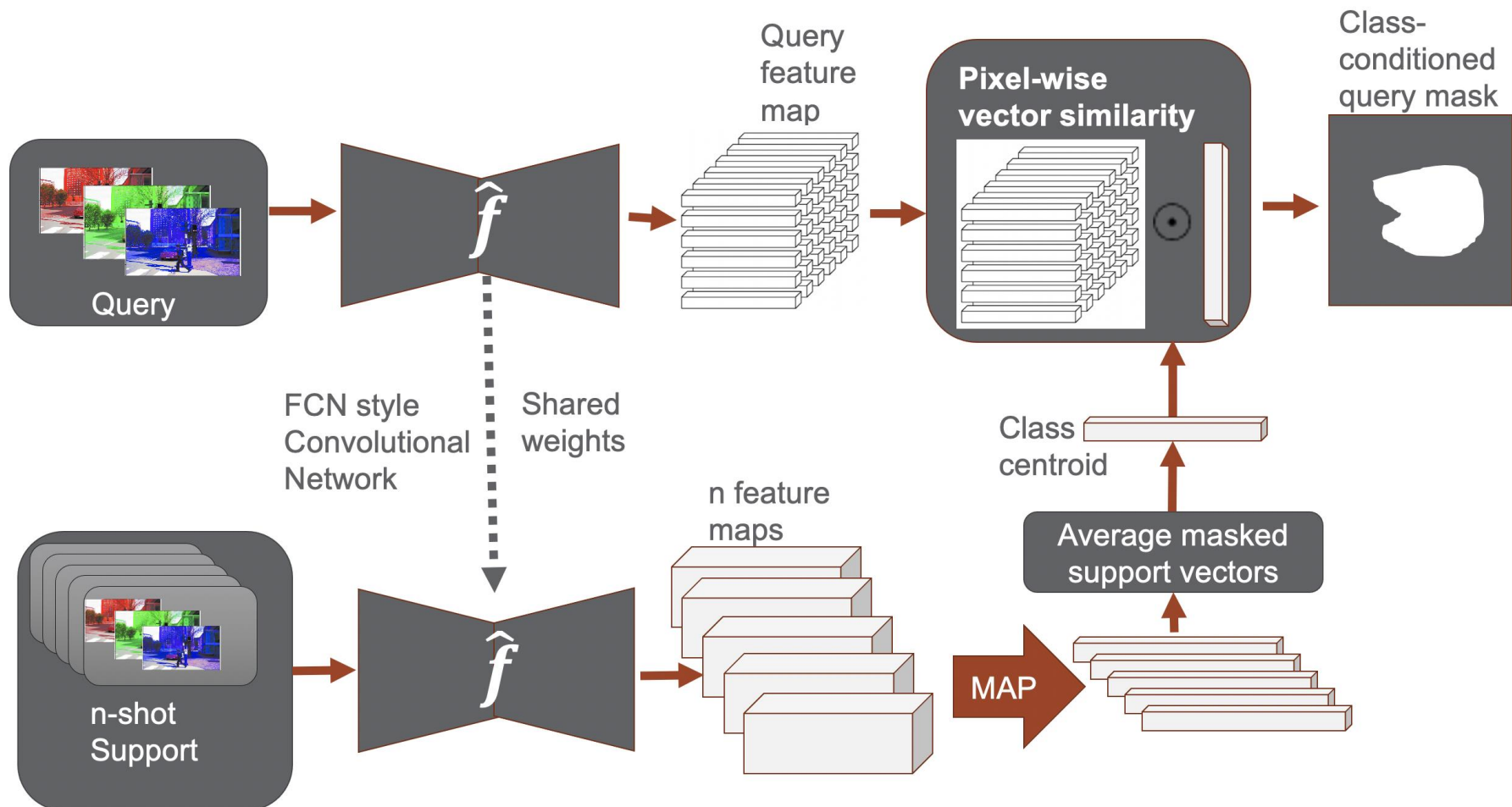




# Few-shot Localization Tasks

- For many problems, knowing *where* an entity is can be as crucial as *what* it is.
  - Disambiguates images with many entities of the same class.
  - Avoids scrubbing through video to find a potential event of interest.
  - Or listening to long audio files!
- Few-shot learning was already a hard problem!
  - We're adding a whole new dimension in the same, small-data regime.
  - Labeling can also be harder: we need location *and* class label.

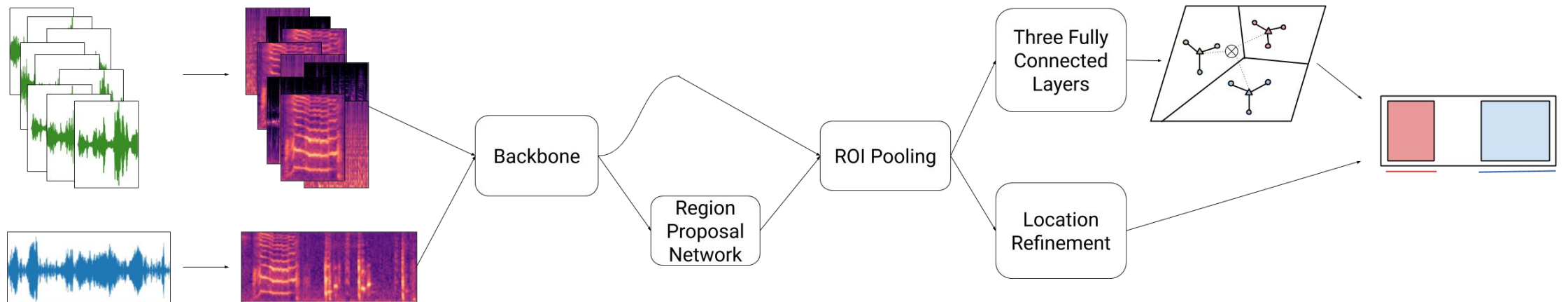
# Few-shot Localization Tasks: Image





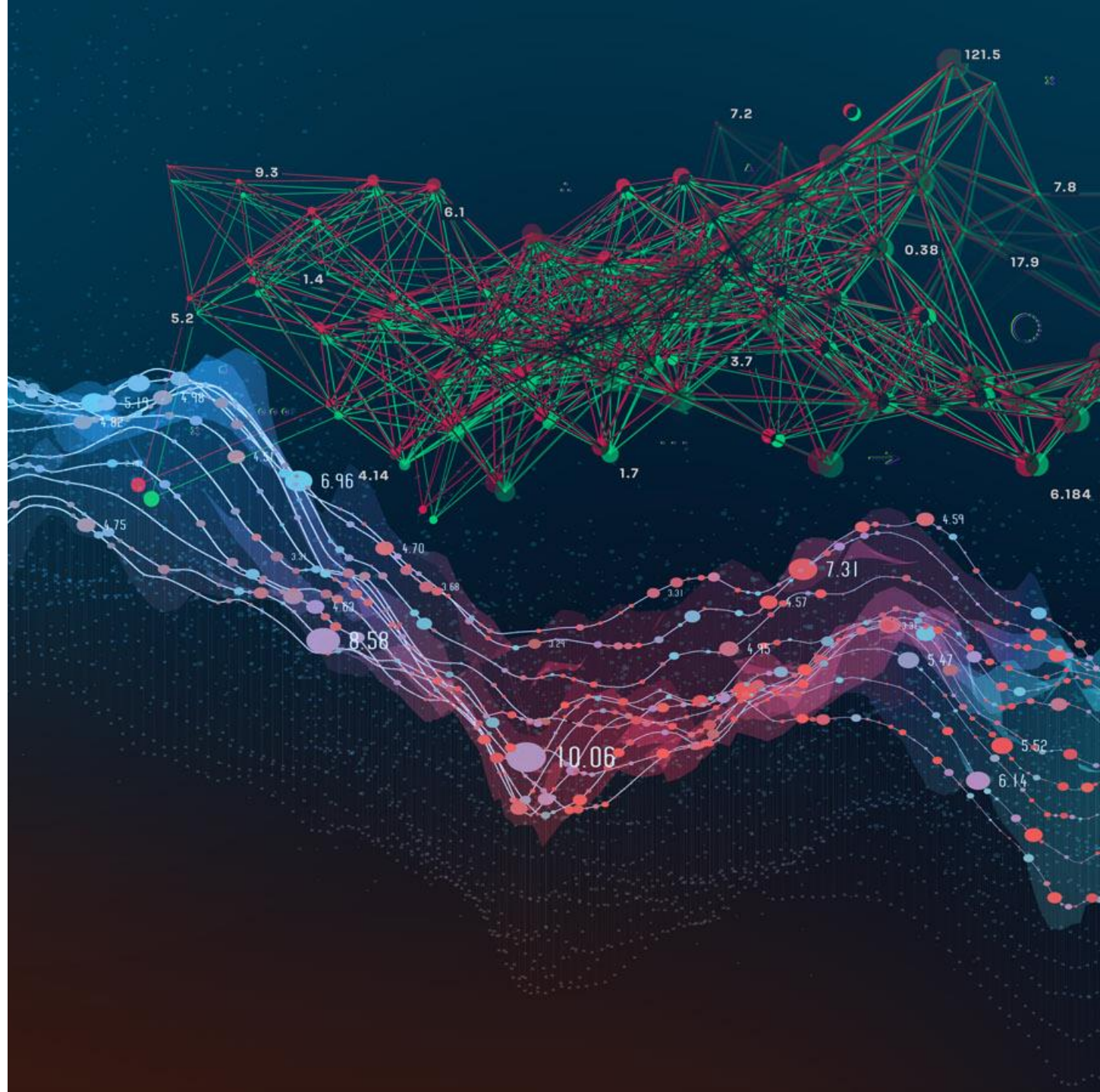
# Few-shot Localization Tasks: Audio / Video

- Localization can be integrated with a typical event detection framework.
  - Video or audio "frames" are encoded with a spatiotemporal backbone.
  - Predict start and end times for event proposals.
  - Proposals of interest are transformed to a fixed size.
  - Perform few-shot classification on proposals of interest.





# Some Extensions to Few-shot

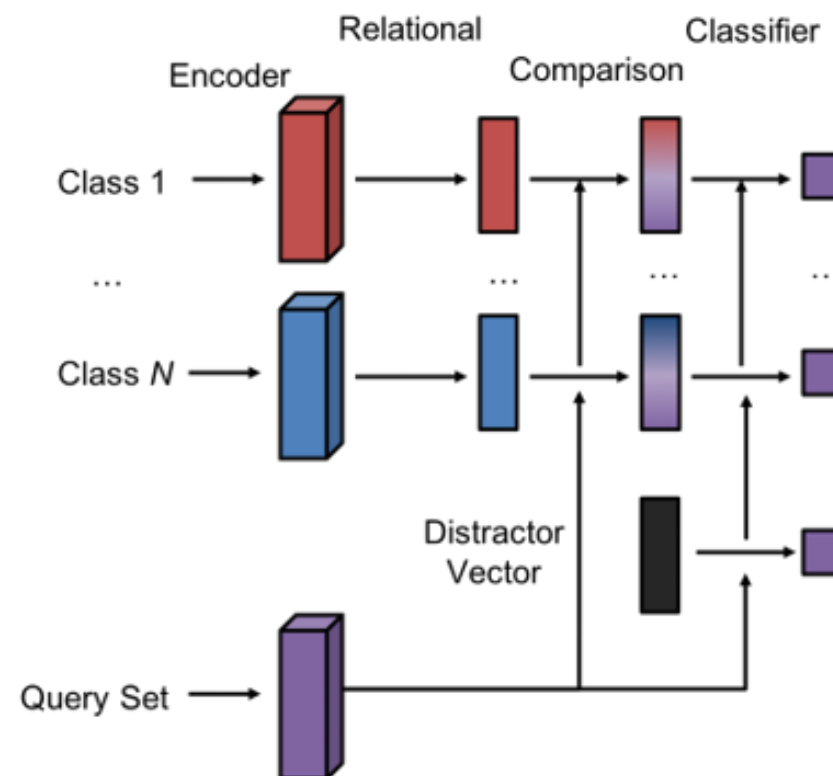


## None-of-the-above, or “Distractors”

- In practice, real-world datasets are made up of individual datapoints that may belong to any number of possible classes
  - In many cases it will be impossible to assign every datapoint to a class
  - Or some classes may just not be very interesting and we’d like to ignore
- Many few-shot models lack the ability to place a datapoint into *no* class
  - E.g. ProtoNets inherently assigns all data to a class
- This is a substantial limitation, especially when applying few-shot models to user interfaces

# Solutions to Distractors

- RCNet solves this problem by creating a vector to represent the distractor class
  - Distractors are still somewhat more difficult than other datapoints
- RCNet distractor performance
  - Precision = 0.79
  - Recall = 0.69
  - Overall acc. = 0.88





# Mistakes as Negative Examples

1. User provides positive examples
2. Model makes decisions
3. User corrects model mistakes
4. Model makes decisions incorporating feedback

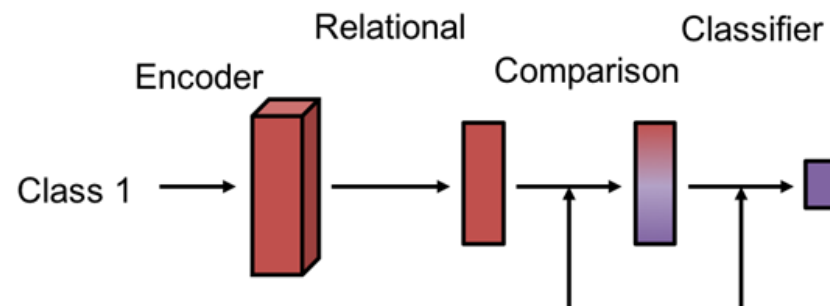




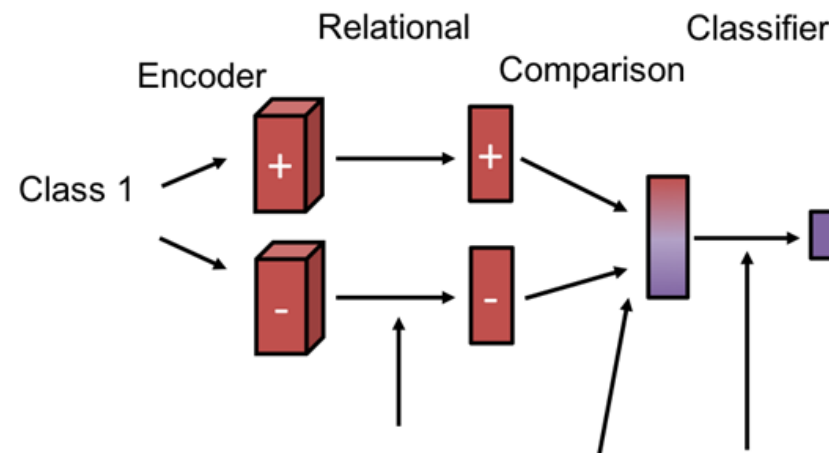
## Negative examples

- Negative examples should inform what a class *doesn't* look like
- Relational network looks for similarity between queries and negatives
- To train the model, we train each episode twice
  - Once with only positives
  - Then a second time with negatives as well

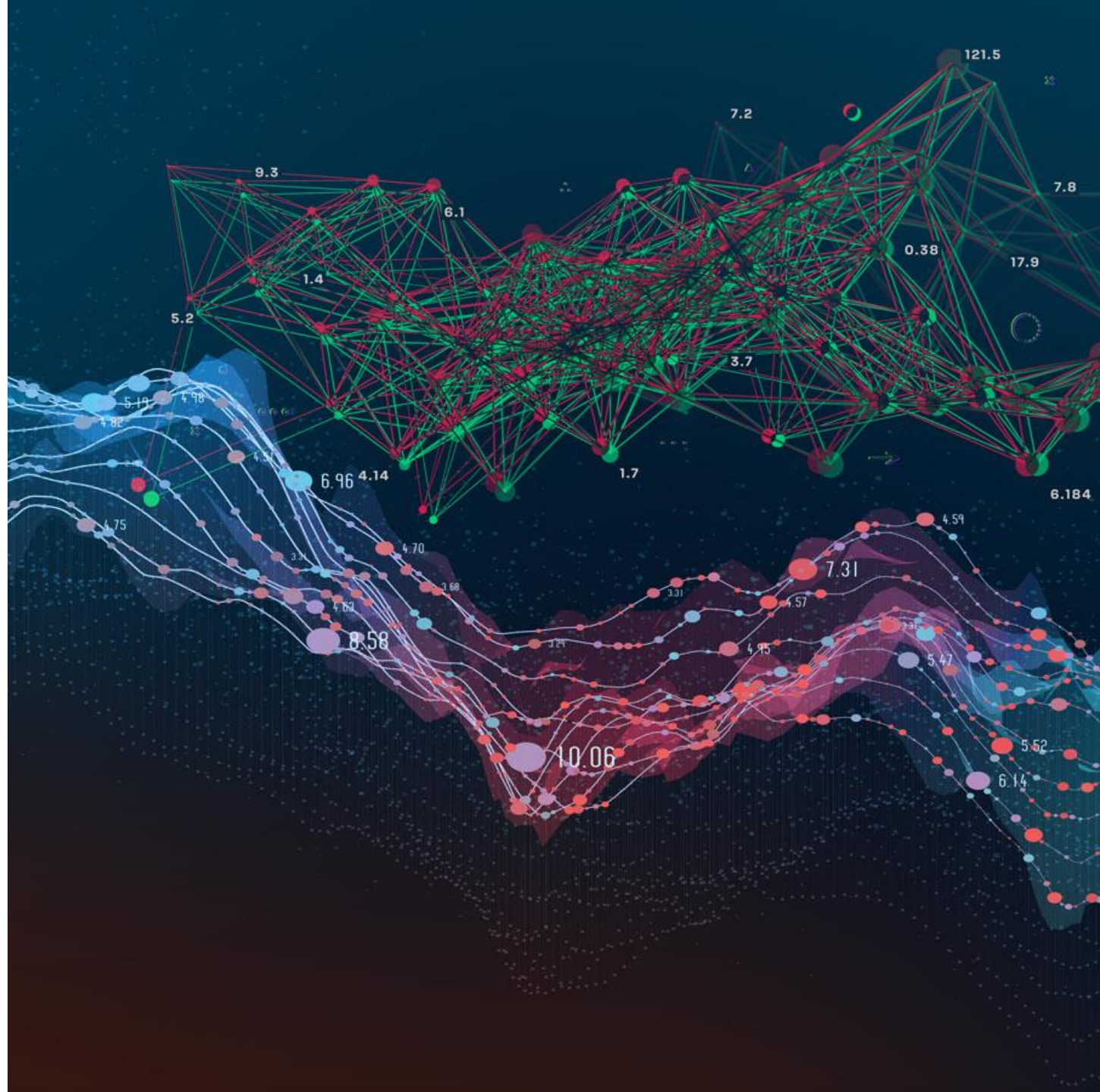
### Positives Only



### With Negatives



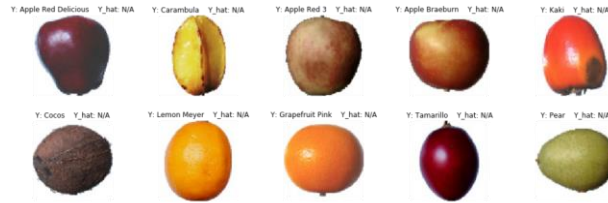
# Day 3 Practical





# Baseline Datasets

- A few pictures highlighting the variety of images in the datasets



## Baseline Takeaways

- Baseline Experiments - *produced experiments on a wide variety of datasets to get a better snapshot of how well models generalize to new data*
- Baseline pipeline - *standardized the addition of new datasets to the pipeline so we can continue to add new datasets with minimal effort*