A Transformer-Based Metagenomic Viral Classification Tool

Brian Li
MIT Primes

Introduction: What is Metagenomics?

- Study of genetic material from environmental samples
- Enables study of microbial communities directly
- Essential for pathogen detection and viral discovery
  - Ex: novel viruses that can cause pandemics
  - Need to distinguish viral sequences within all the metagenomic data

- Challenge: limited viral references and high mutation rates
  - Traditional tools (e.g., BLAST) are slow and perform poorly on divergent viral sequences

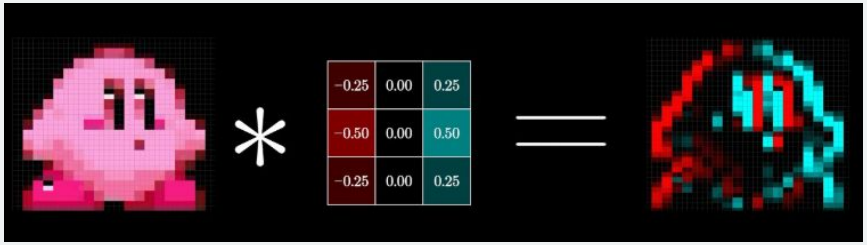
AI in metagenomics

- VirFinder (2017): logistic regression on k-mers
- Bzhalava et al. (2018): random forests improve accuracy
- VirSorter2 (2021): 5-random forest

- Older models
- Most remain mostly binary classifiers
- Limited ability to identify related species

CNNs

- VirDetect-AI (Zárate et al., 2025) introduced CNNs for multiclass viral detection
- CNNs detect local patterns using filters (kernels) that are run through the ML
- Allows identification of local sequence features by the ML



Credit: 3blue1brown: Diagram of kernels in CNNs

- Allows classification of many classes & performs better than other tools
  - 900+ classes, in comparison to < 5 for previous models

Problems with CNNs

- CNNs are good at local pattern recognition through kernels
- Limited receptive field misses long range dependencies
- Fails to model global genomic context
    - Ex: non-coding introns between exons, structures that fold together from an amino acid chain

Transformers

- New architecture introduced in 2017
- Basis of all LLMs
- Self-attention mechanism captures long-range relationships efficiently

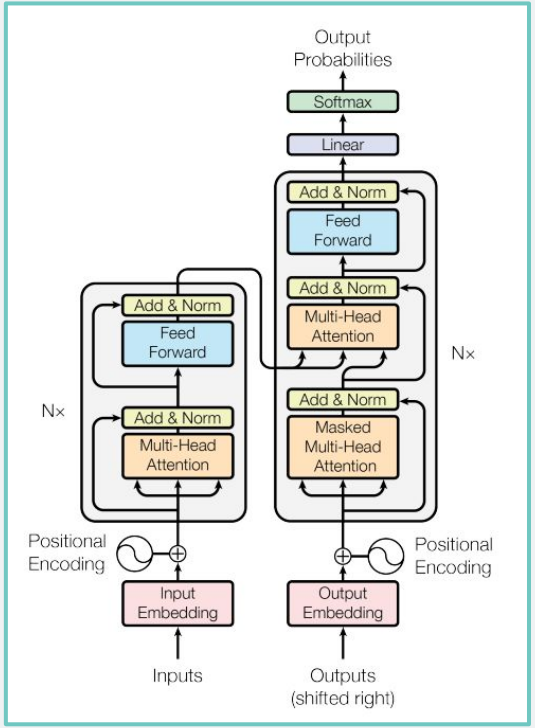


Diagram of transformer architecture (Vaswani et al.)

Model Architecture Details

#	Step	Purpose
1	Input (One-hot DNA)	Encoded nucleotide matrix (A, T, C, G)
2	Conv1D – 64 filters, kernel = 7, stride = 1	Learns short local motifs and converts bases to embeddings efficiently
3	ReLU Activation	Adds non-linearity for motif detection
4	Transformer Encoder × 4, 4 heads per layer, dim=64	Models long-range dependencies via self-attention
5	Mean Pooling	Pools sequence features
6	Dropout (0.2)	Prevents overfitting
8	Softmax	Outputs probability distribution for classes

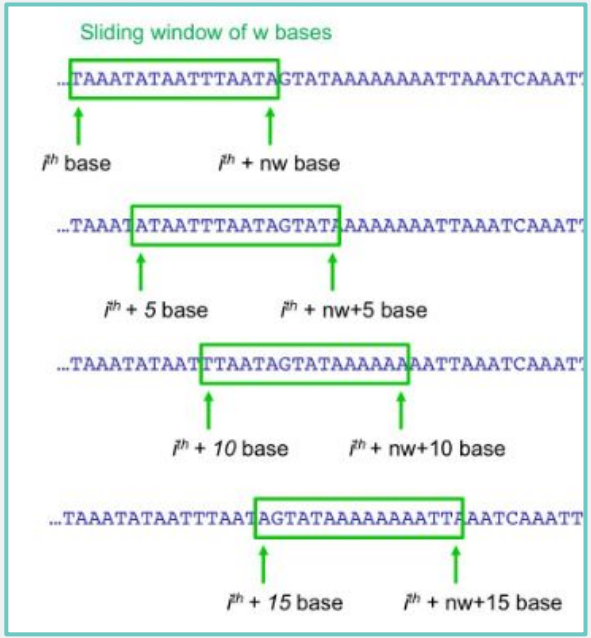
Training Setup

- Data: NCBI Virus Nucleotide Database
  - Training set: **Nucleotide** sequences found before September 2020
  - Test dataset: ~9000 random sequences found after September 2020
- Non-viral class was taken from random samples of the human + bacterial genomes
- Clustering samples based on sequence similarity with mmseqs2
  - 1.6 million sequences after filtering out marginal classes
  - 176 classes

Fragmentation

- The output of metagenomic data is DNA fragments
- Model needs to be trained on these DNA fragments

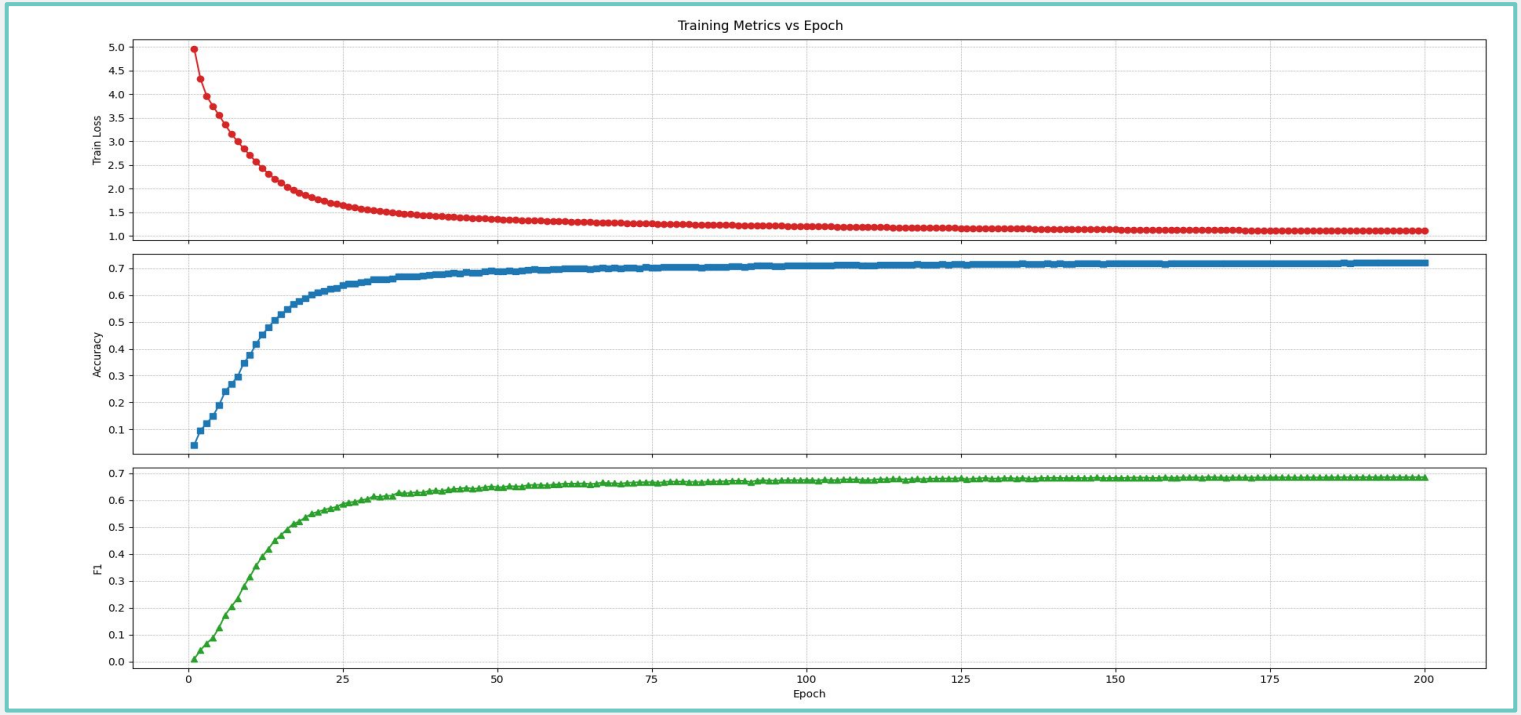
- 300bp sliding window with a 20bp step



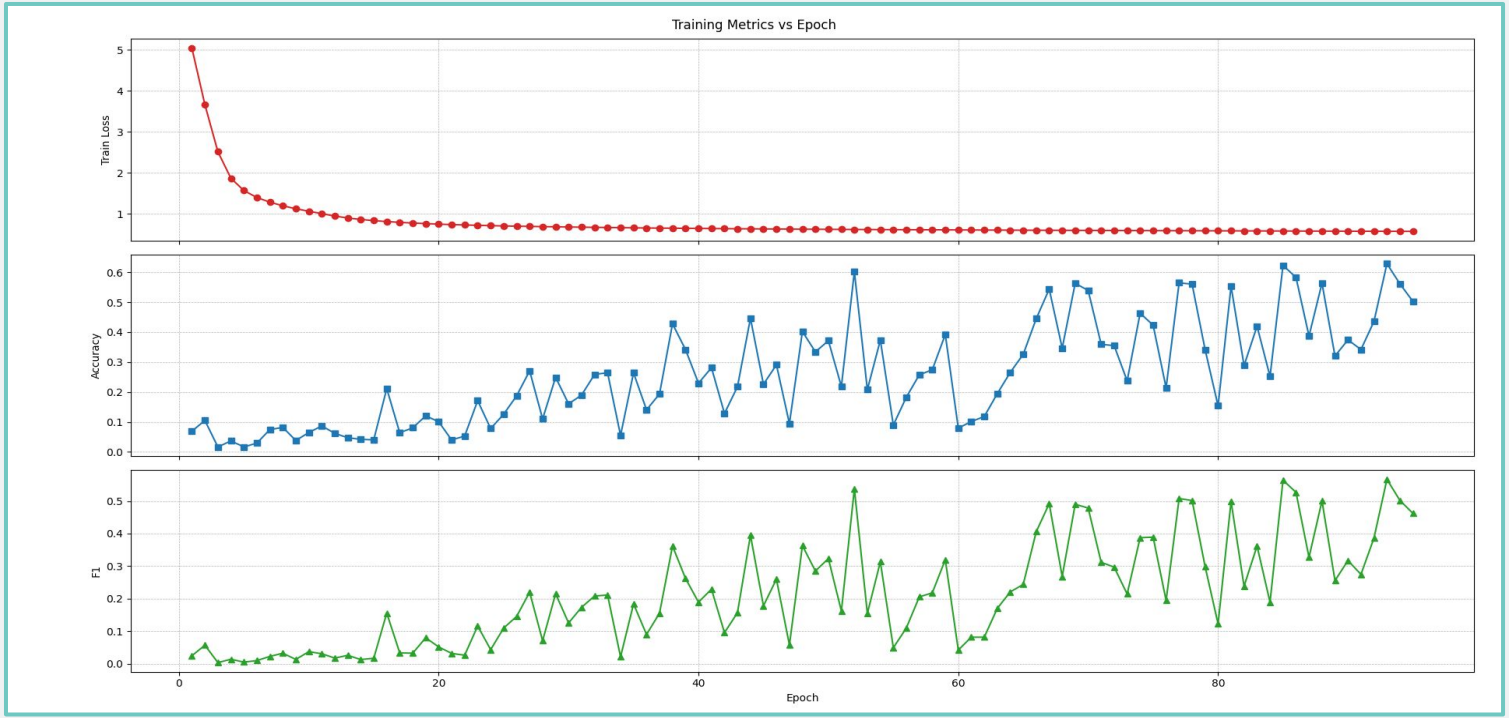
Results

- VirDetect-AI architecture was replicated and evaluated on the same dataset
- Accuracy, loss, and F1 score was used as metrics

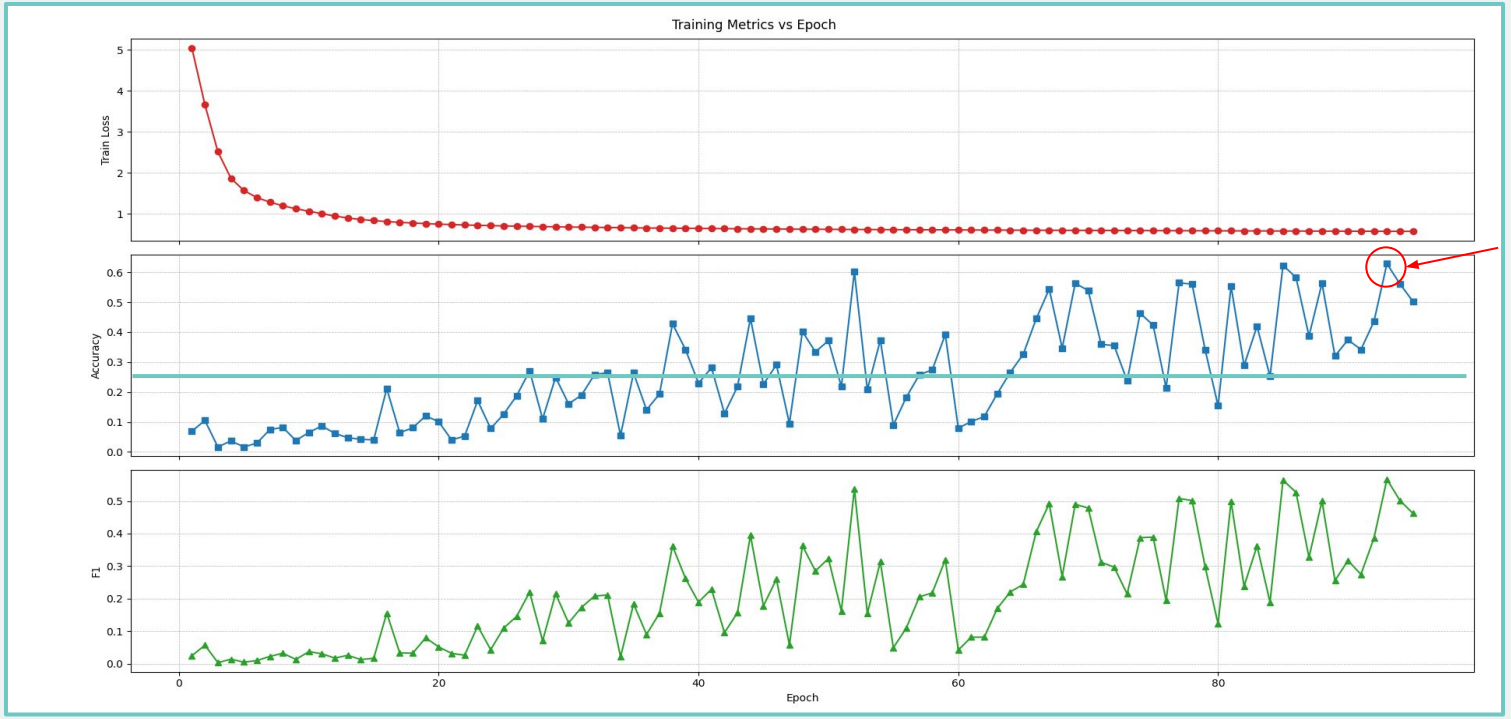
Results


Training Metrics vs Epoch
Train Loss
Accuracy
F1
Epoch
0
25
50
75
100
125
150
175
200

VirDetect-AI architecture



VirDetect-AI architecture



Future work

- Add more to the NV dataset
  - Fungal, archeal data
- Optimize transformer parameters
- Experiment with variable window sizes and k-mer embeddings

Conclusion

- Transformer model converges a lot more stronger generalization
- Achieves higher accuracy and F1-score with smoother training curves than VirDetect-AI
- Demonstrates better generalization to unseen viral data
- Strong foundation for next-generation metagenomic virus classifiers

Acknowledgements

- Thank you to Dr. Gil Alterovitz, Dr. Ning Xie, and all members of the Alterovitz lab for guiding me through the research

Thank you!