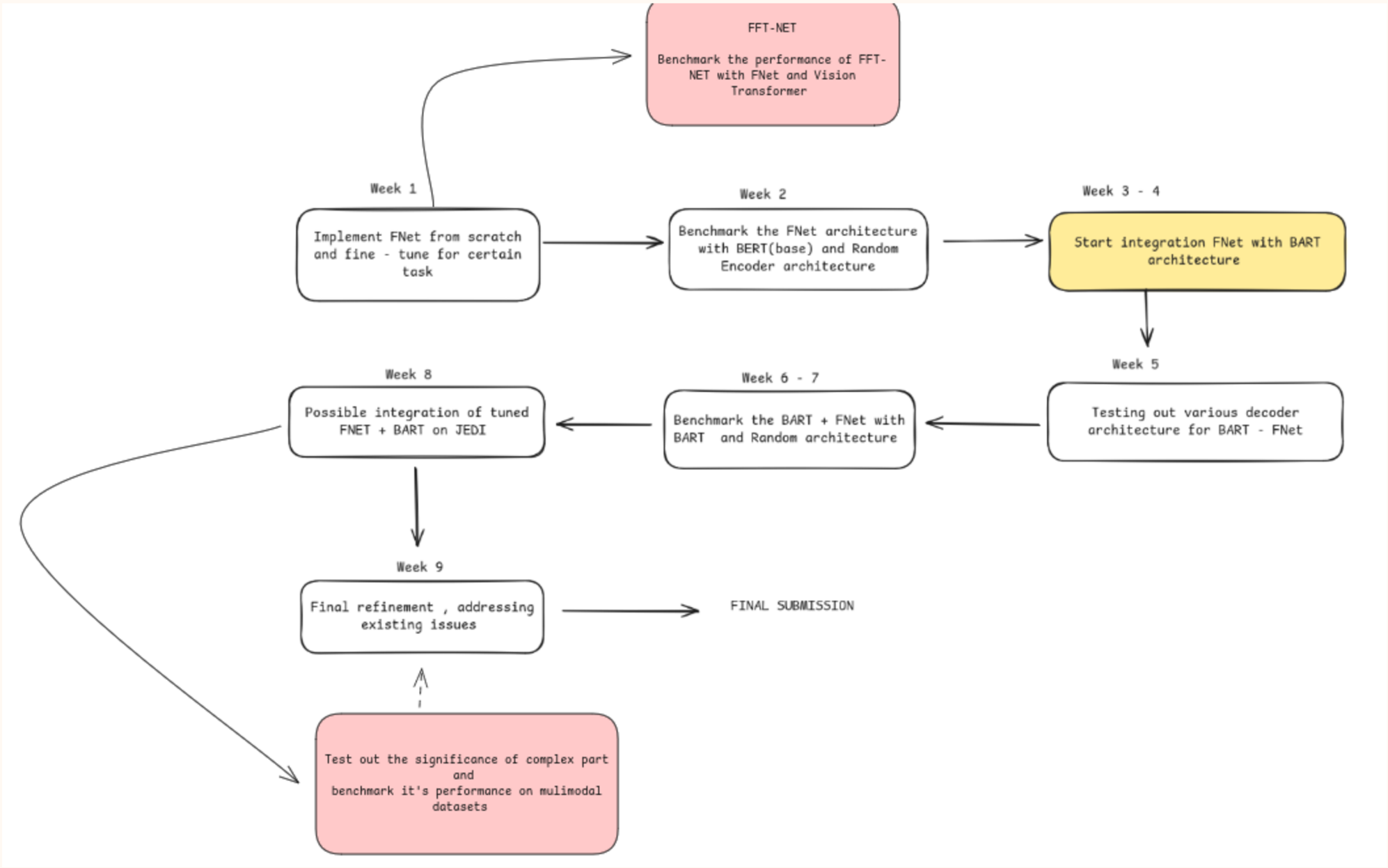
FNET + BART : IS FOURIER ALL YOU NEED ?

Samkit Jain , Aryan Garg , Aniruth Suresh

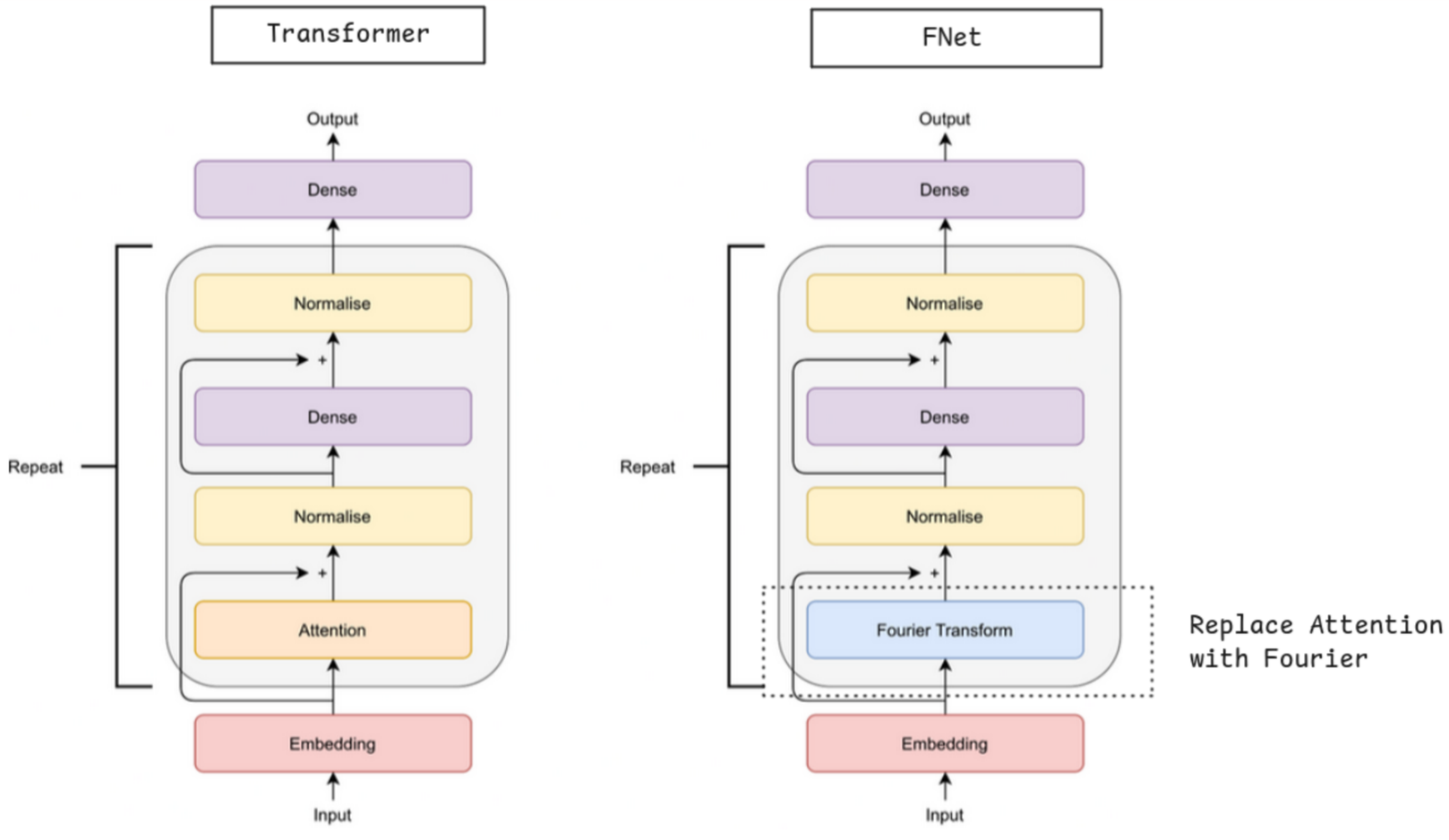
GitHub Repo: Link

Project Outline



F-Net Architecture

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi f t} dt$$

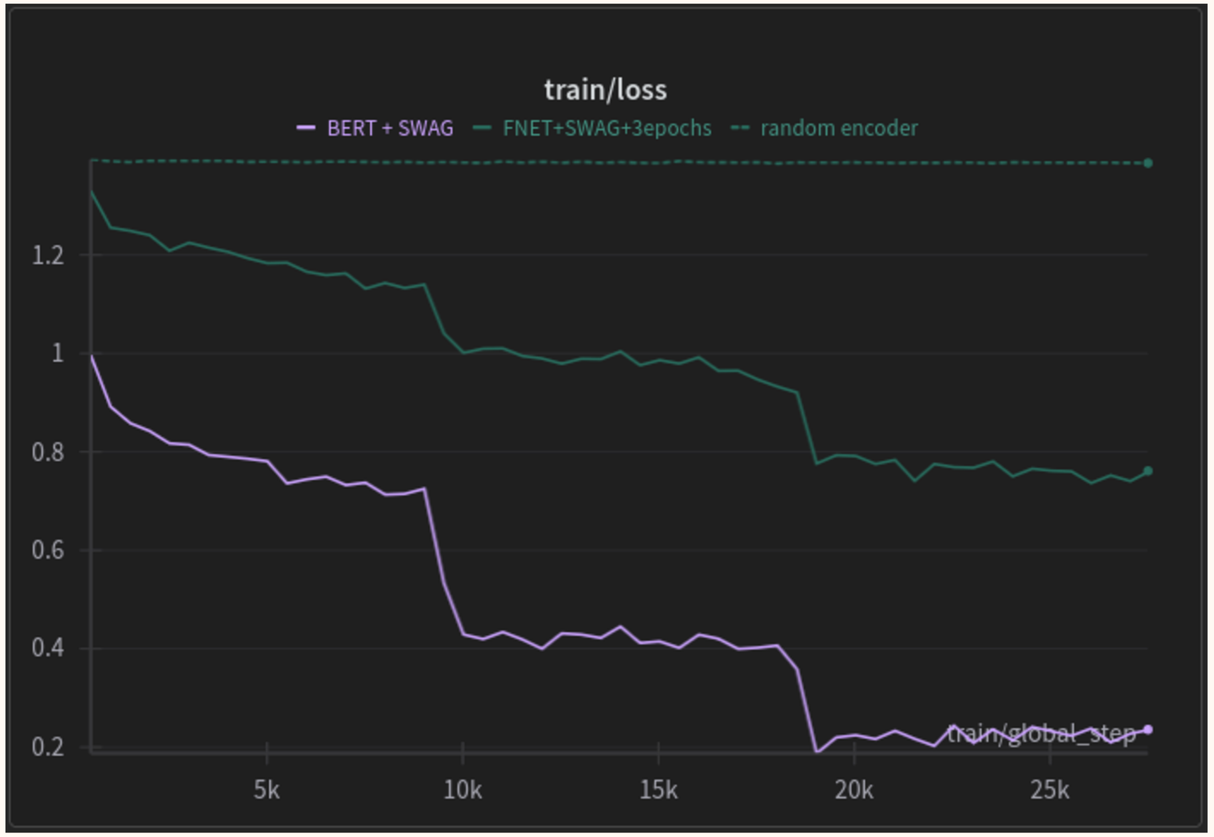


F-Net swaps the standard, computationally heavy self-attention layers found in Transformers.

Uses DFT as the core operation for mixing of information. Capable of performing this mixing without the need for learnable parameters.

Reduces computational complexity from O(n*n) to O(nlogn)

F-Net Benchmarking



Name (3 visualized)	Runtime
BERT + SWAG	2h 4m 15s
FNET+SWAG+3epochs	46m 11s

3 models :

1. Fnet + SWAG
2. BERT + SWAG
3. Random (replaces attention by fixed matrices)

OBSERVATIONS :

1. Clearly from the above graph , we observe that we can't just replace attention by any random fixed matrix .
2. This clearly signifies the importance using **Fourier Transform** .

Remember : FT has no learnable parameters !

FNet Benchmarking

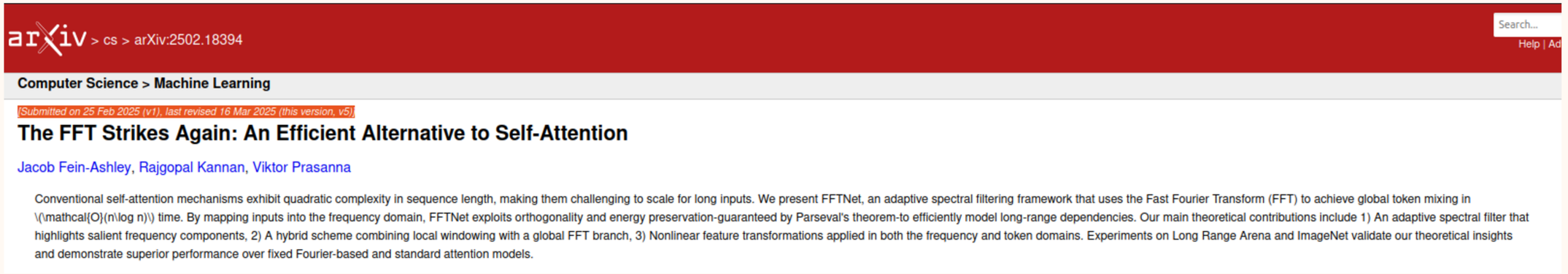

```python
# Example test case
context = "The weather was getting colder and the leaves were falling from the trees."
choices = [
    "She decided to wear a light summer dress.",
    "He put on a heavy winter coat.",
    "They went to the beach to enjoy the sun.",
    "The sun was shining brightly in the sky."
]

predicted_index = predict(model, tokenizer, context, choices)
print(f"Predicted choice: {choices[predicted_index]}")
```

```
(fart) aniruth.suresh@gnode010:~$ python3 check.py
Evaluating: 100%|
Accuracy: 0.7800
F1 Score: 0.7799
Predicted choice: He put on a heavy winter coat.
(fart) aniruth.suresh@gnode010:~$
```

Given a context and set of four options , F-Net predicted the most appropriate one which around 78% accuracy and F1 -score !

FFT-Net ?



arXiv > cs > arXiv:2502.18394

Search... Help | Ad

Computer Science > Machine Learning

[Submitted on 25 Feb 2025 (v1), last revised 16 Mar 2025 (this version, v5)]

The FFT Strikes Again: An Efficient Alternative to Self-Attention

Jacob Fein-Ashley, Rajgopal Kannan, Viktor Prasanna

Conventional self-attention mechanisms exhibit quadratic complexity in sequence length, making them challenging to scale for long inputs. We present FFTNet, an adaptive spectral filtering framework that uses the Fast Fourier Transform (FFT) to achieve global token mixing in \(\mathcal{O}(n\log n)\) time. By mapping inputs into the frequency domain, FFTNet exploits orthogonality and energy preservation-guaranteed by Parseval's theorem-to efficiently model long-range dependencies. Our main theoretical contributions include 1) An adaptive spectral filter that highlights salient frequency components, 2) A hybrid scheme combining local windowing with a global FFT branch, 3) Nonlinear feature transformations applied in both the frequency and token domains. Experiments on Long Range Arena and ImageNet validate our theoretical insights and demonstrate superior performance over fixed Fourier-based and standard attention models.

Paper published on **16th March 2025** that builds on the idea of F-Net.

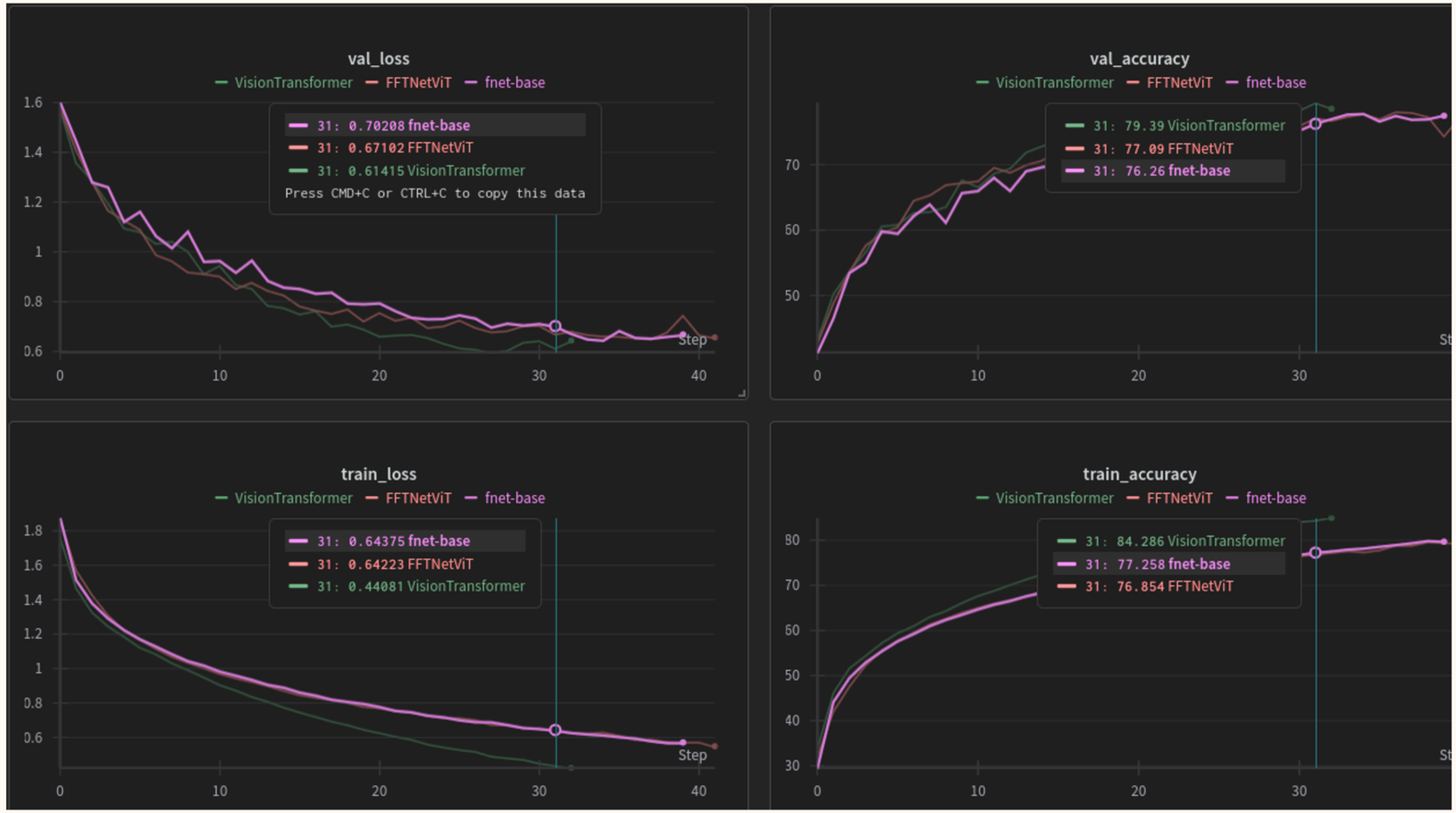
It introduces a learnable, context dependent filter in the frequency domain to dynamically emphasize or attenuate required frequency components, enhancing its performance over the fixed parameters of F-Net.

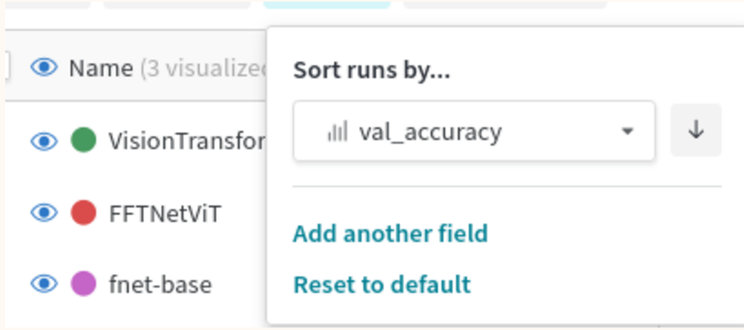
It also applies a non-linear activation function (modReLU) directly to the complex FT coefficients after filtering, resulting in better representation of token dependencies.

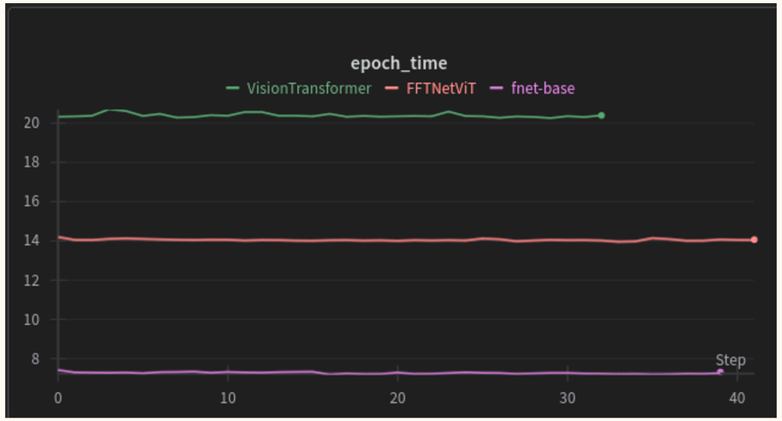
Even though this adds some computational overhead compared to the base F-Net, the complexity is still about O(nlogn). Better benchmarks on datasets are also observed on finetuning.

FFT-Net ?

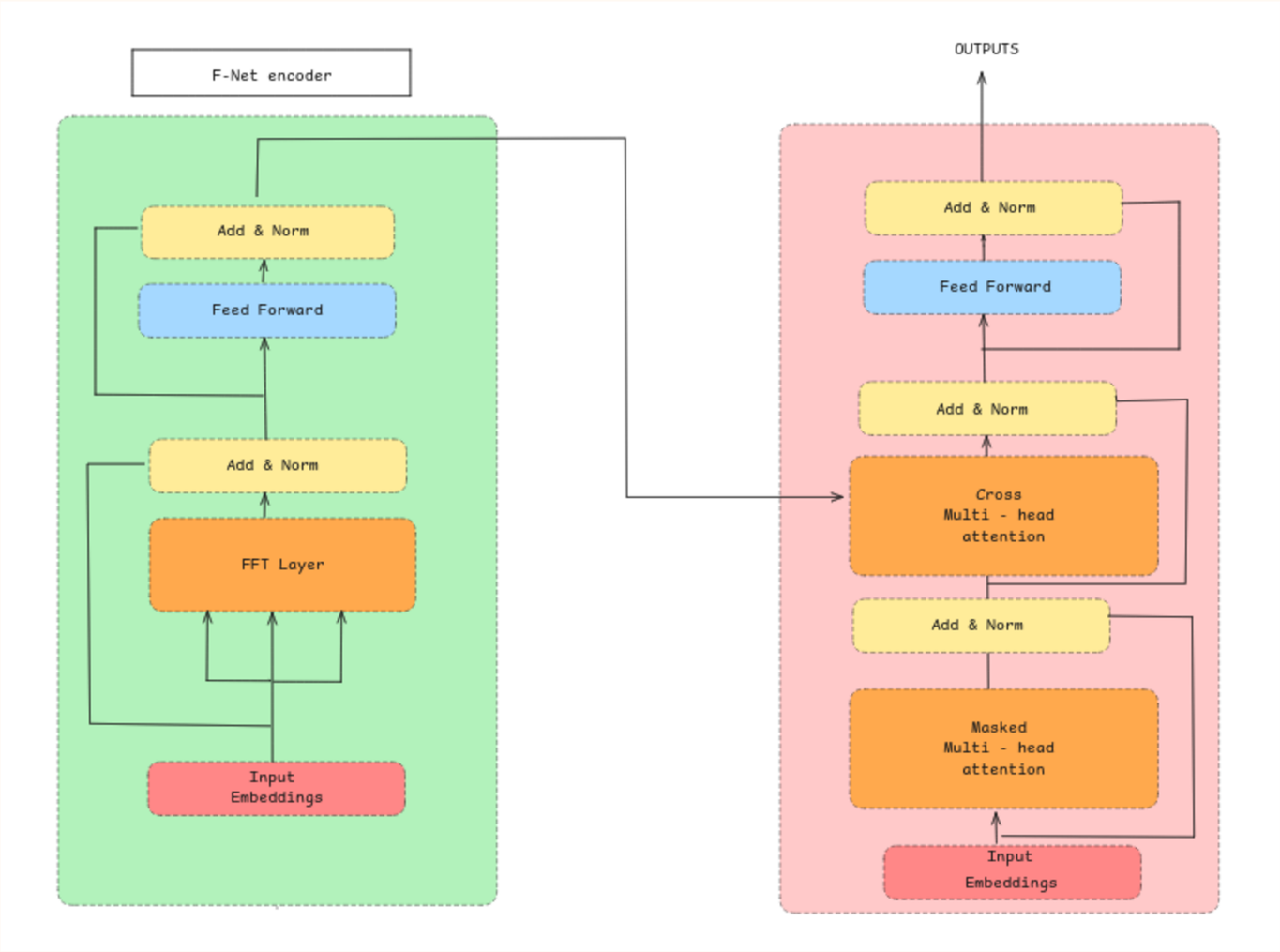
Task : CIFAR - 10 classification





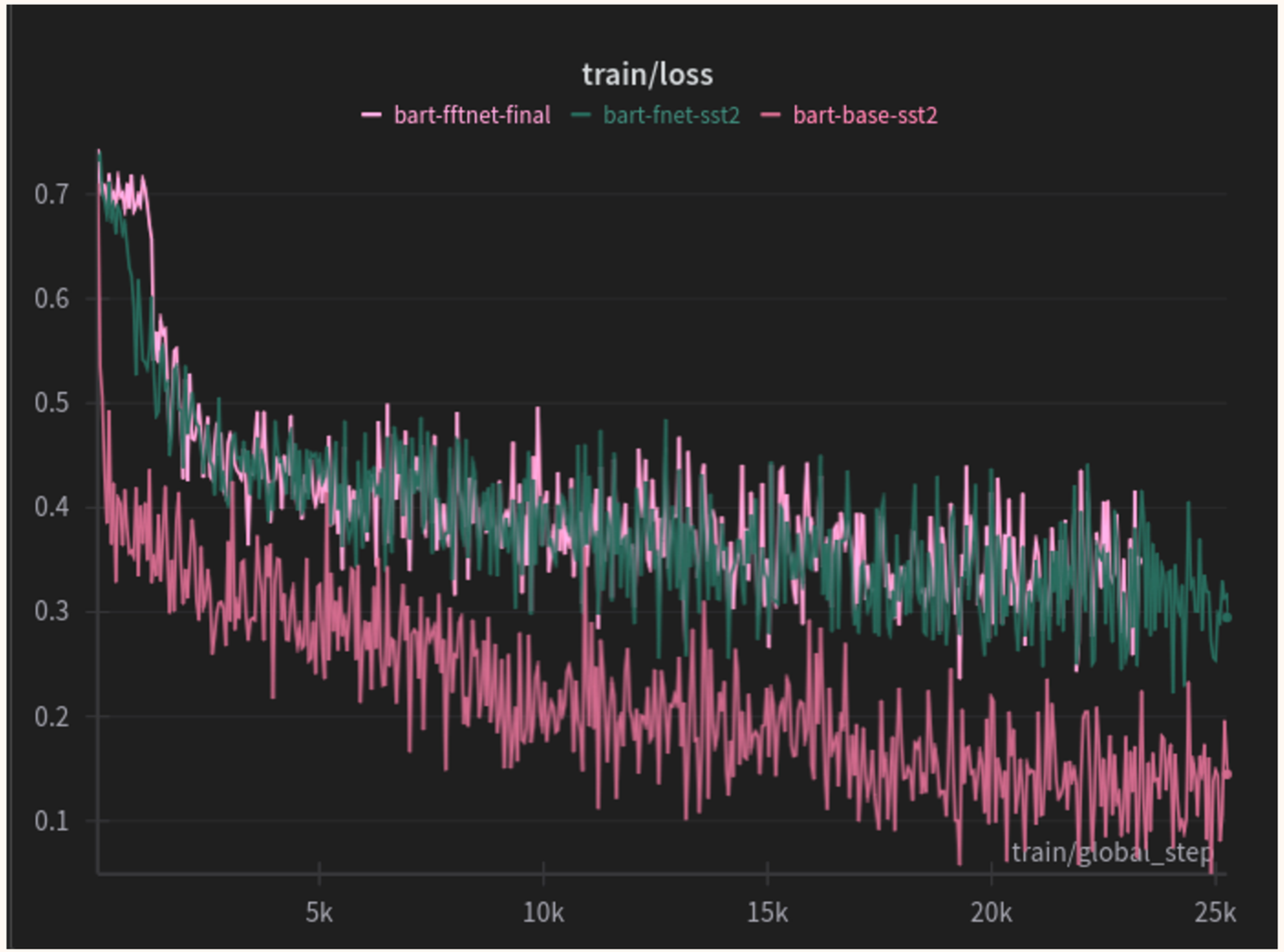


FNet + BART Encoder Only



FFT mixing incorporated in BART by replacing the attention heads by FFT Layers.

1. Encoder Only



me (5 visualized)	Run
bart-fnet-sst2	10m 48s
bart-fftnet-sst2	13m 33s
base-BASE-sst2	22m 52s

We train on the **SST-2 Dataset** which performs well for both regular BART , BART with FNet layers , FFTNET

Time - Accuracy Tradeoff

1.Encoder Only

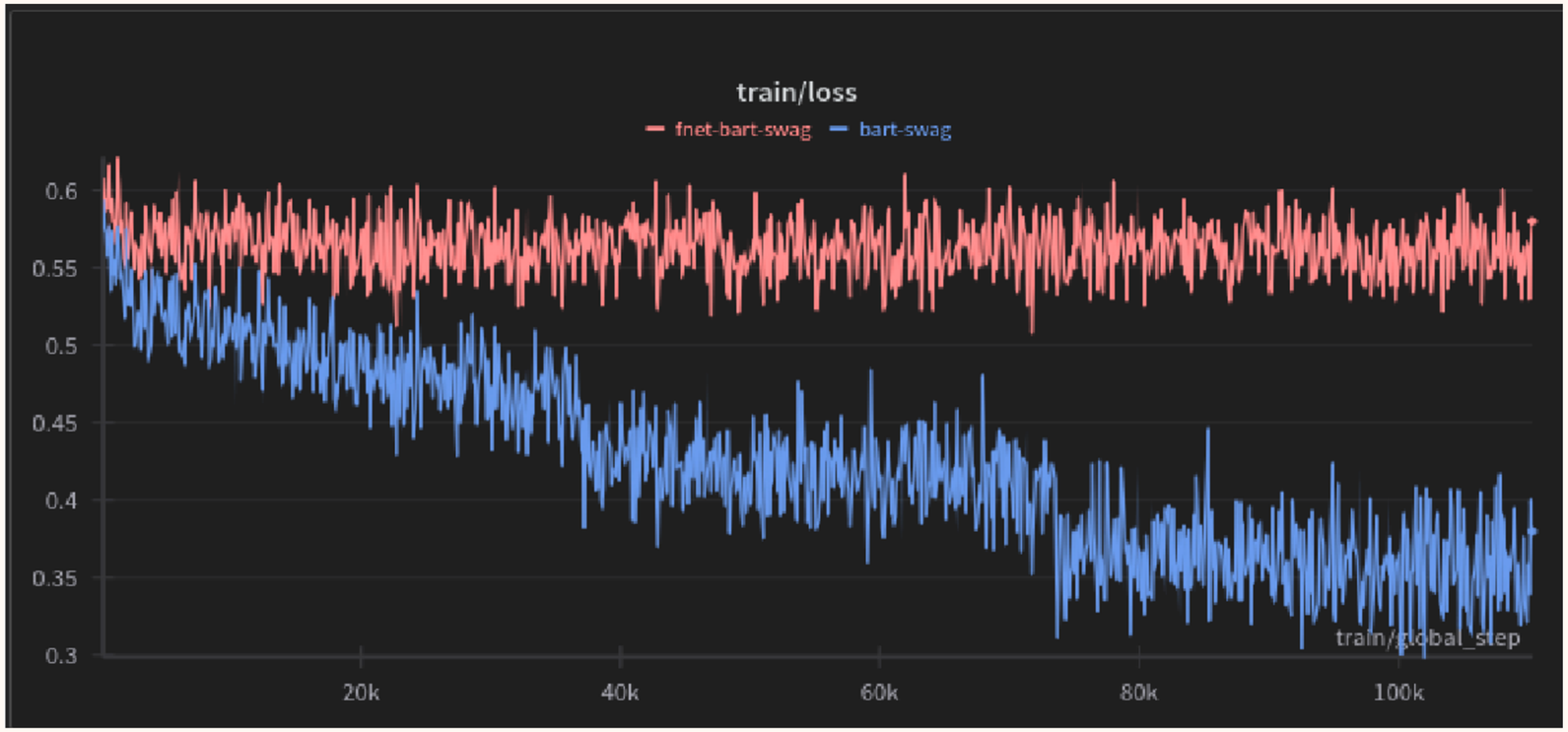
Model	Eval Accuracy	Eval Loss	Total time taken
Base BART	92.231%	0.40153	22m 52s
BART + FNet	83.37%	0.559	**10m 48s**
BART + FFTNET	82.22%	0.492101	13m 33s

```
{'eval_loss': 0.4015326499938965, 'eval_accuracy': 0.9231651376146789, 'eval_runtime': 1.4848, 'eval_samples_per_second': 587.272, 'eval_steps_per_second': 73.409, 'epoch': 3.0}
{'train_runtime': 1430.5598, 'train_samples_per_second': 141.236, 'train_steps_per_second': 17.655, 'train_loss': 0.21926027940411502, 'epoch': 3.0}
100%|██████████| 109/109 [00:01<00:00, 75.65it/s]
{'eval_loss': 0.35456112027168274, 'eval_accuracy': 0.9277522935779816, 'eval_runtime': 1.4543, 'eval_samples_per_second': 599.621, 'eval_steps_per_second': 74.953, 'epoch': 3.0}
```

```
{'eval_loss': 0.5593359470367432, 'eval_accuracy': 0.8337155963302753, 'eval_runtime': 1.301, 'eval_samples_per_second': 670.257, 'eval_steps_per_second': 83.782, 'epoch': 3.0}
{'train_runtime': 1249.9381, 'train_samples_per_second': 161.646, 'train_steps_per_second': 20.207, 'train_loss': 0.38505867073455396, 'epoch': 3.0}
100%|██████████| 109/109 [00:01<00:00, 86.53it/s]
{'eval_loss': 0.5593359470367432, 'eval_accuracy': 0.8337155963302753, 'eval_runtime': 1.2711, 'eval_samples_per_second': 685.997, 'eval_steps_per_second': 85.75, 'epoch': 3.0}
```

Epoch	Training Loss	Validation Loss	Accuracy
1	0.385100	0.491956	0.794725
2	0.374400	0.492101	0.822248

1. Encoder Only

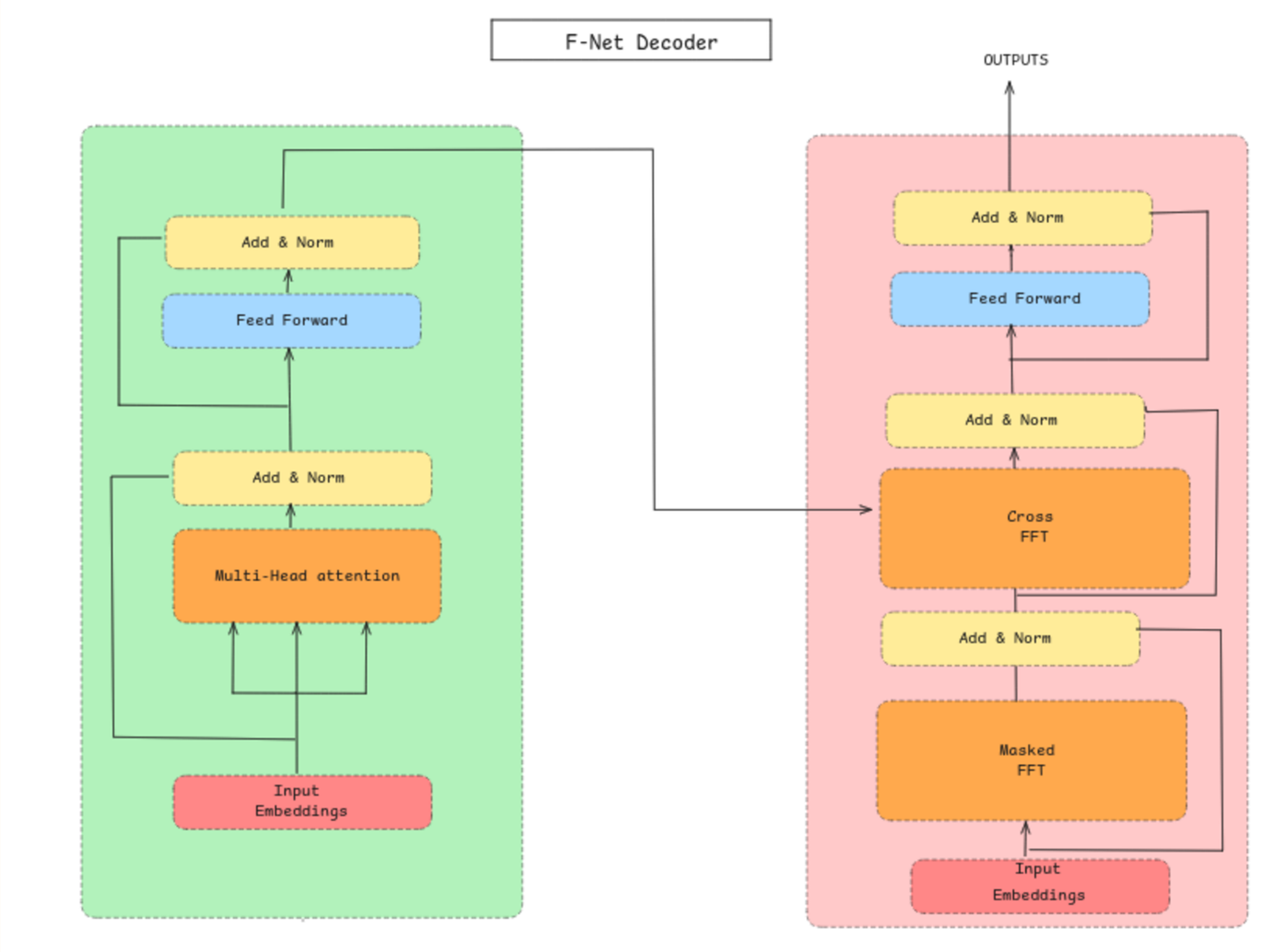


Reasons for poor performance :
SWAG depends heavily on understanding context and making inferences, which often benefits from attention. By replacing the self-attention layer with Fourier Transforms the model may be **losing important semantic information** required for accuracy.

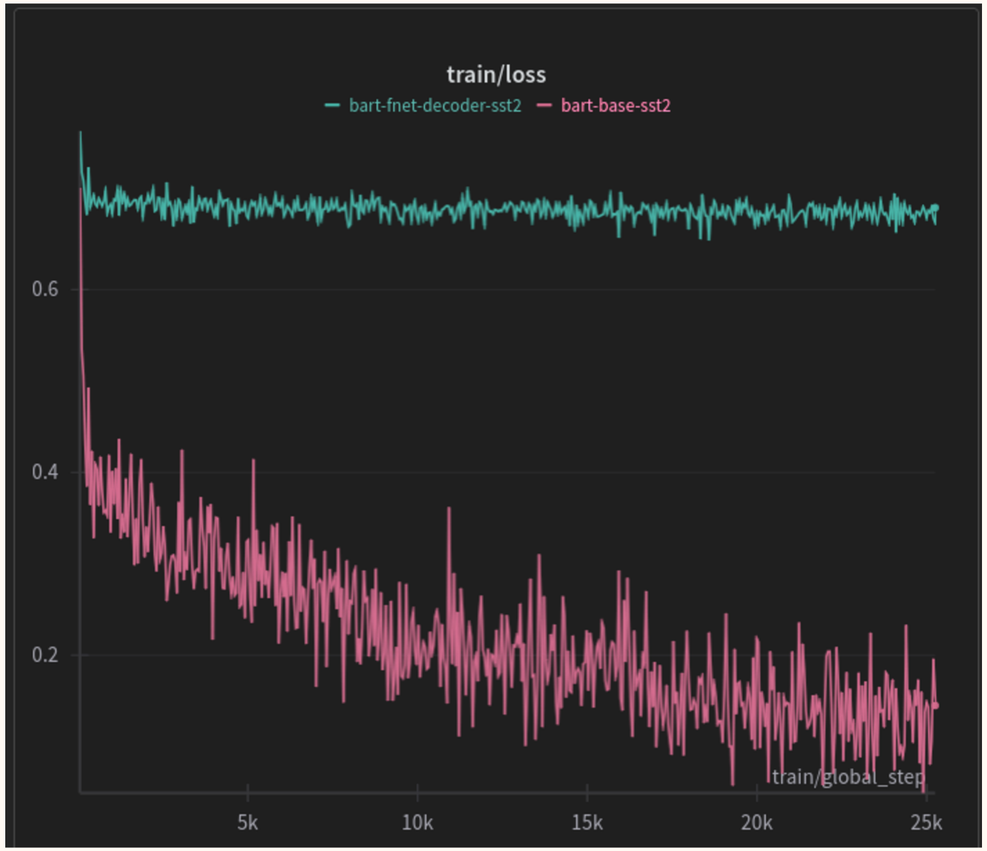
CLAIM : Implementing F-Net on BART encoder seems to perform well on simple tasks like sst-2 , MNIST and CIFAR but fails to generalize to heavy complex task like SWAG which requires context !!

2. Decoder Only

We replace both the self attention and the cross attention in decoder by FT and let the encoder have attention blocks !



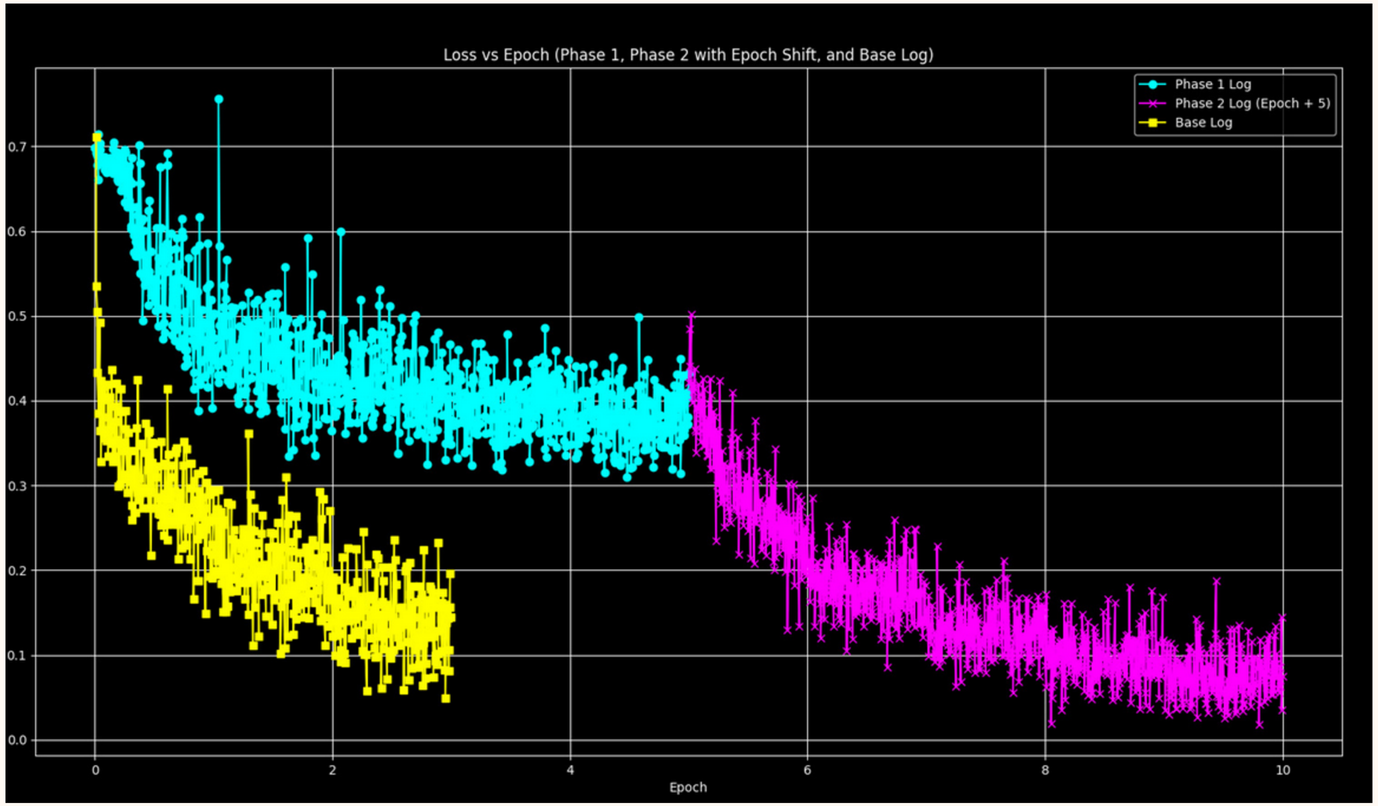
Attempt 1 : Directly replacing the self and cross attention by FT (sst-2 dataset)



ISSUES :

1. Gradients were tending to zero => Vanishing gradients
2. Concatenating all the token embeddings and then projecting it => too many dims and time taking !

Attempt 2: 2 Stage training Approach



SOLUTIONS:

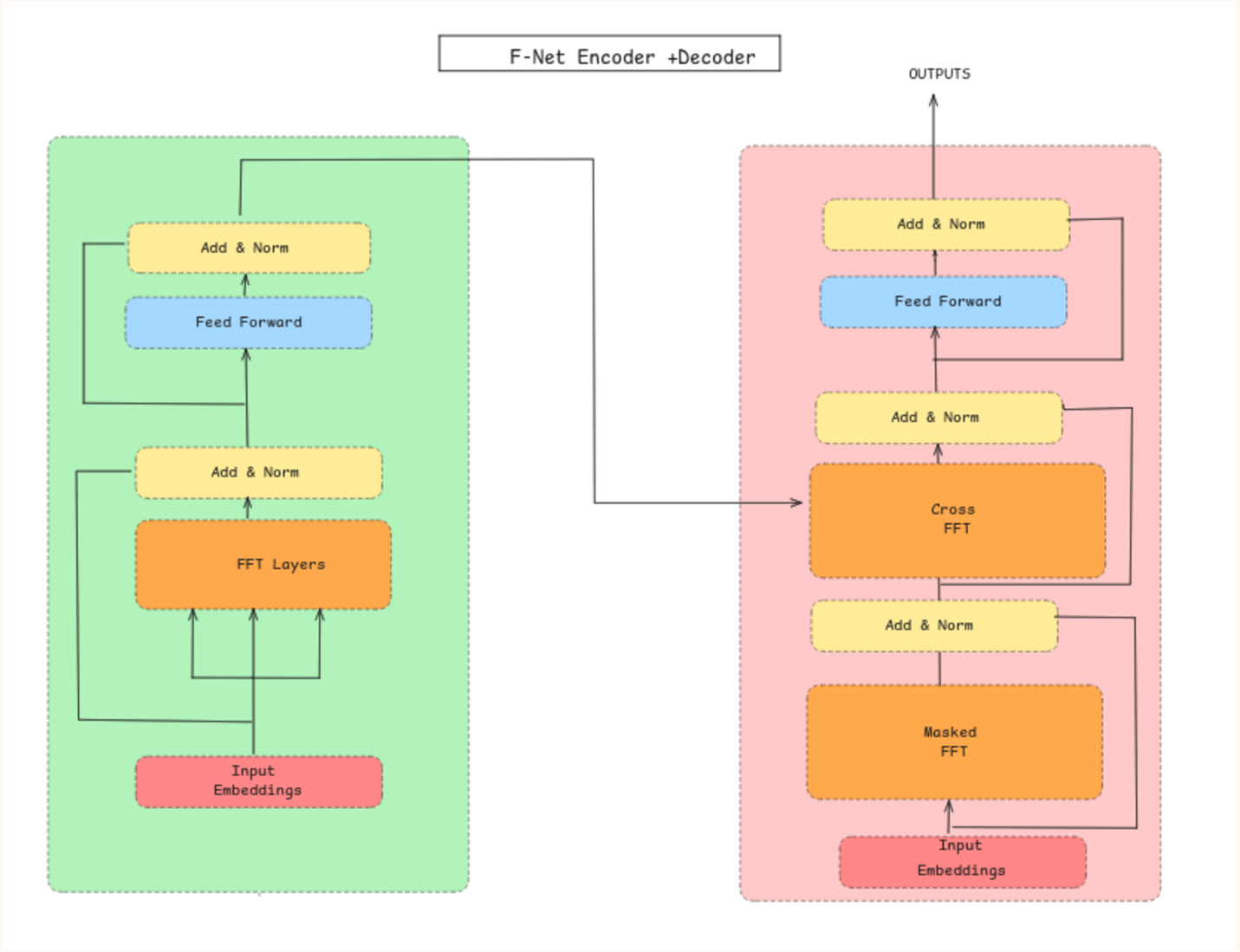
1. Added residual connections in multiple parts of the decoder to tackle Vanishing Gradients.
2. Added token embeddings instead of concatenating.

2-Stage training approach:

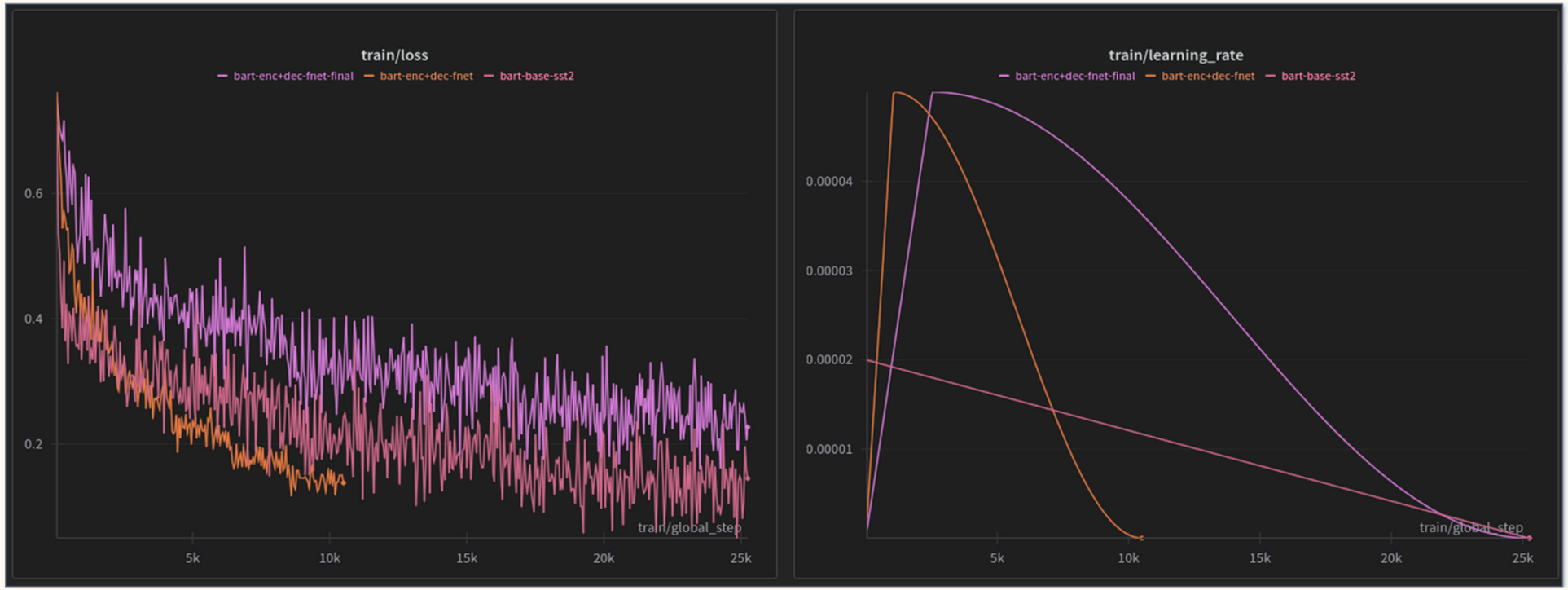
1. Freeze the encoder parameters and train only the decoder.
2. Train both the encoder and decoder parameters.

3. Encoder+Decoder

We replace all the self attention and the cross attention in both decoder and encoder !



3. Encoder+Decoder



2 Training Strategies

1. Epochs = 5, BS = 16, Gradient Accumulation = 2
2. Epochs = 3, BS = 1, Gradient Accumulation = 1

Gradient Accumulation:

Instead of updating model weights after each batch, the gradients are accumulated and over multiple batches and updates are made at once.

Complete evaluation details

sst -2 dataset metrics

Model Setting	Eval Loss	Eval Accuracy	Time per Epoch (s)
Base (BART)	0.3546	92.78%	17m 13s
Decoder-Only* (FT)	0.3090	92.20%	12m 33s
Encoder-Only (FT)	0.5593	83.37%	14m 17s
Encoder + Decoder (FT)	0.3939	**83.49%**	**9m 17s**

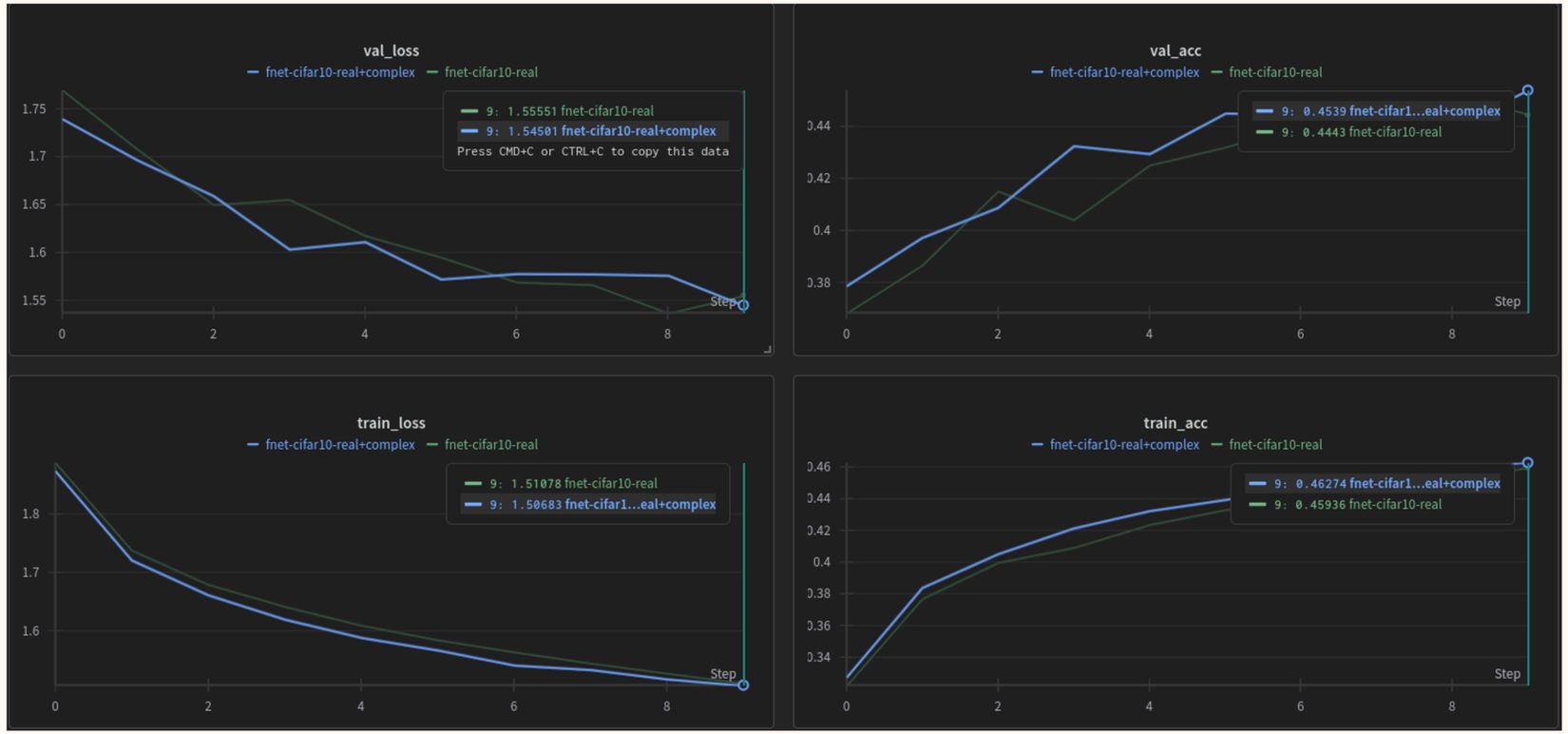
SIGNIFICANT IMPROVEMENT IN TIME : Base BART takes about 17m for a 92% acc , whereas , replacing both encoder and decoder self and cross attention , we achieve a 84% acc with almost half the time (9 min) !!!

Complex info ?

All the implementations - Fnet , FFTNet and FourierTransformer all uses only real part of the fourier transform !

We attempt to include the complex part as well : **concat([real , complex])** => **MLP overhead** => **project to d dim (rest same architecture)**

Dataset 1 : Image = CIFAR-10



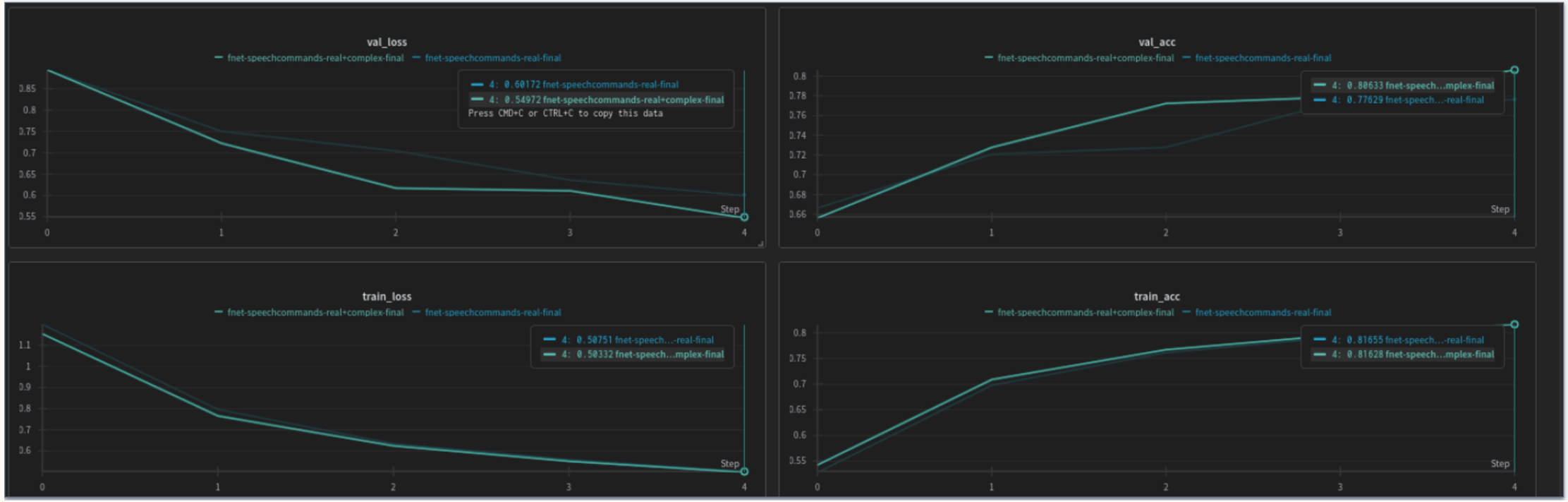
1.03% increase in validation accuracy . Image datasets might not be heavily influenced by complex features !

Complex info ?

Dataset 2 : Audio = SpeechCommands

Considered a smaller set of the complete dataset - only 6 classes : ***down, left, no, right, up, yes***

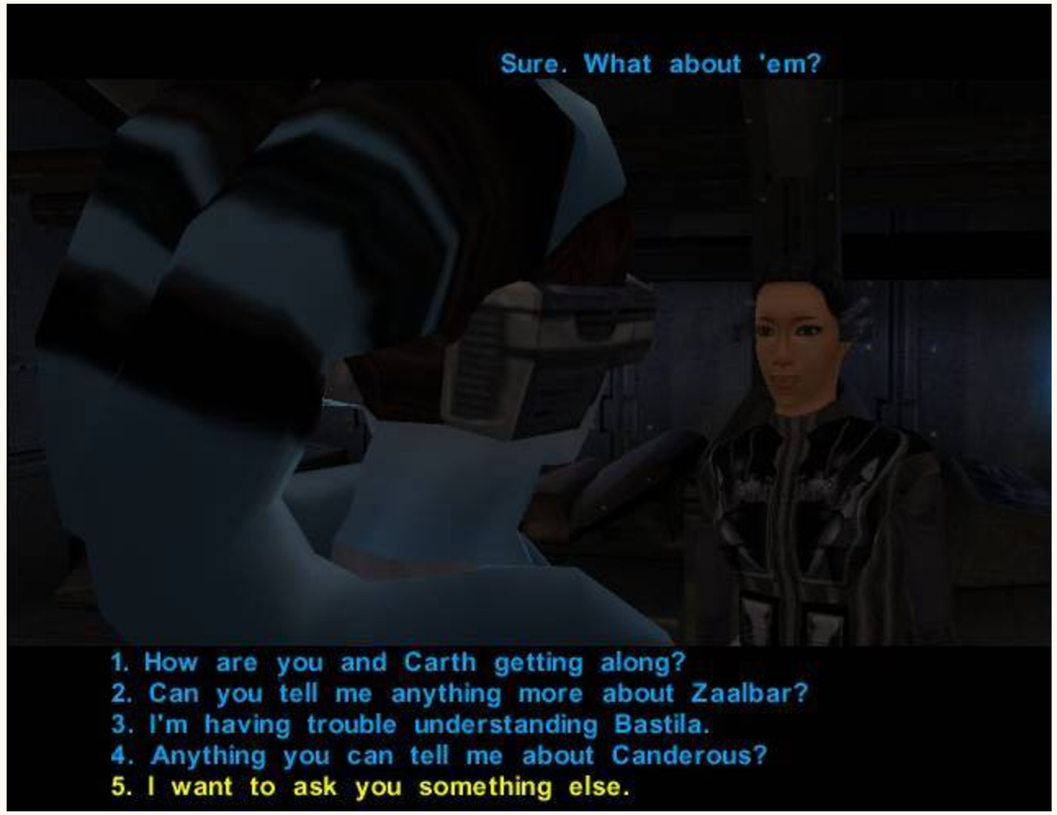
(computational reasons)



After extracting the mel features and feeding it , we achieve a clear 3% improvement on the val set .

Not a simple one - to - one fixed conversations => it depends on player choices, character stats, and game states

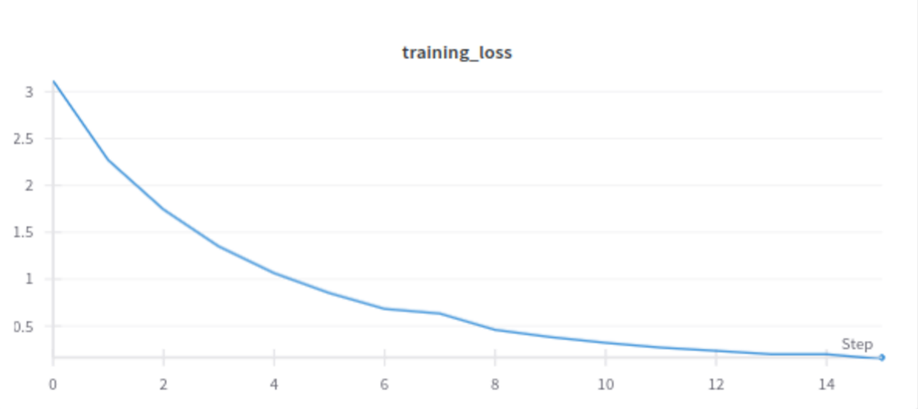
Star Wars: "Knights of the Old Republic" (KOTOR)

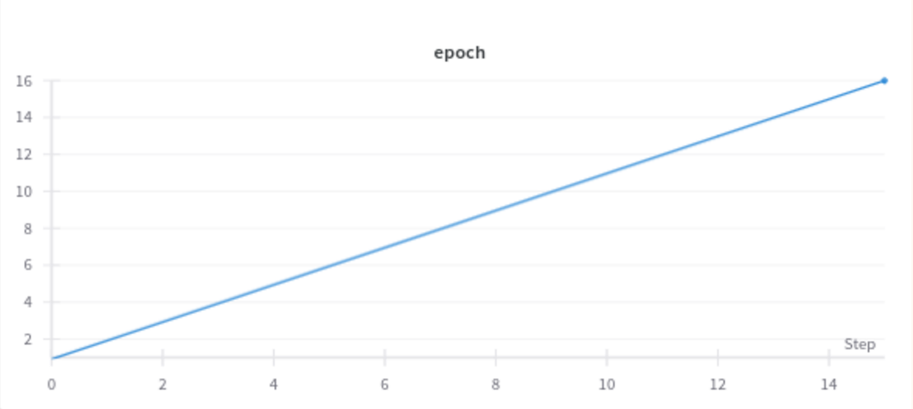


Key idea : Represent "Dialogue as a graph"

1. **Nodes** = individual dialogue utterances and **Edges** = transitions between utterances, which are determined by the game state
2. Similar dialogue nodes are grouped using clustering algorithms (A basic threhsold F1 score based algo is implemented) .
3. Graph is linearized
4. During training, one utterance is masked at a time within this sequence. The model is asked to predict the masked line given the other lines in the cluster and the current game state !

JEDI (base): Justifiable End-dialogue Driven Interaction





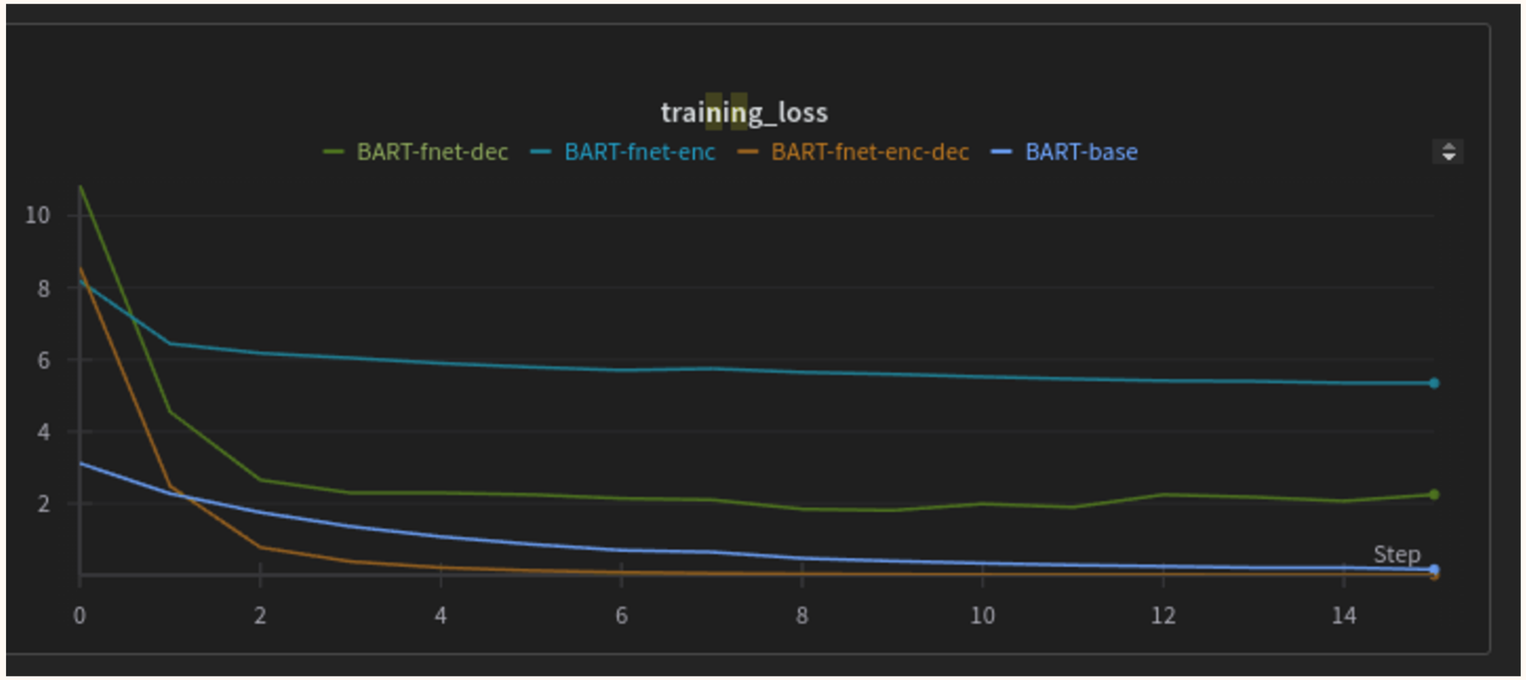

```
Average Precision: 0.8625
Average Recall: 0.8591
Average F1 Score: 0.8606
```

```
DialogRPT Score: 0.6154
Average DialogRPT Score: 0.5027941809351749
(fart) aniruth.suresh@gnode076:~/JEDI$
```

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

1. Q = current dialogue input
2. K , V = representation of the game state
3. QK^T = computes how well each element in the dialogue input (query) matches each element in the game state (similarity score)

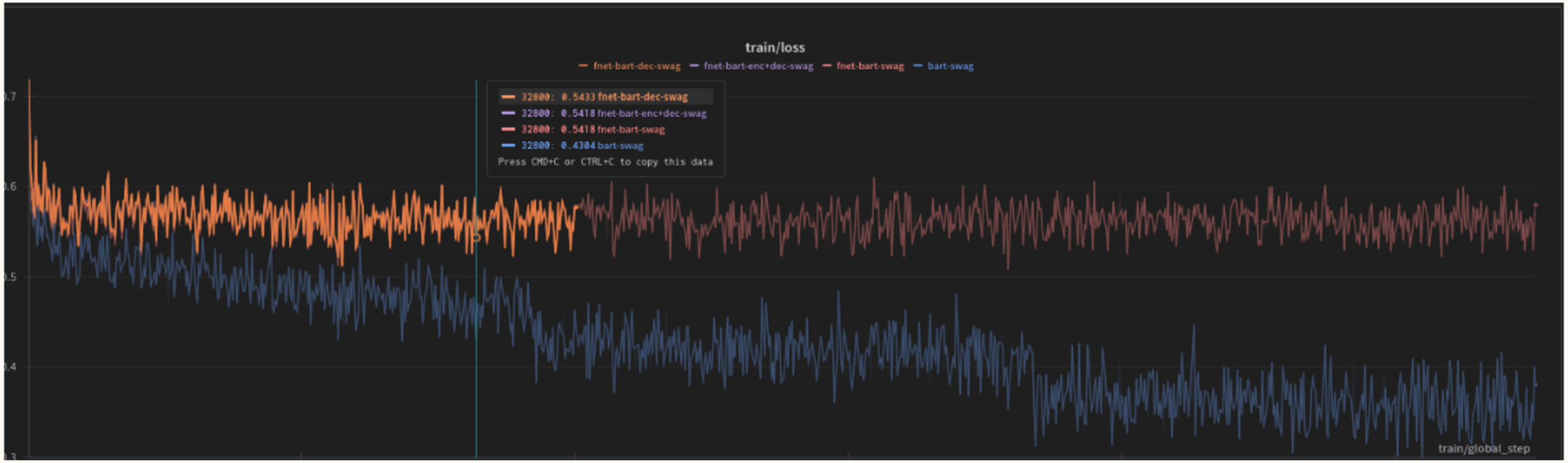
JEDI : All combinations



Model	Precision	Recall	F1-Score	DialogueRPT
Base BART	0.86	0.86	0.86	0.50
Encoder+Decoder FNet BART	0.78	0.84	0.81	0.50
Encoder Only Fnet BART	0.63	0.65	0.64	0.41
Decoder Only FNet BART	0.71	0.74	0.72	0.46

Further Notable Attempts

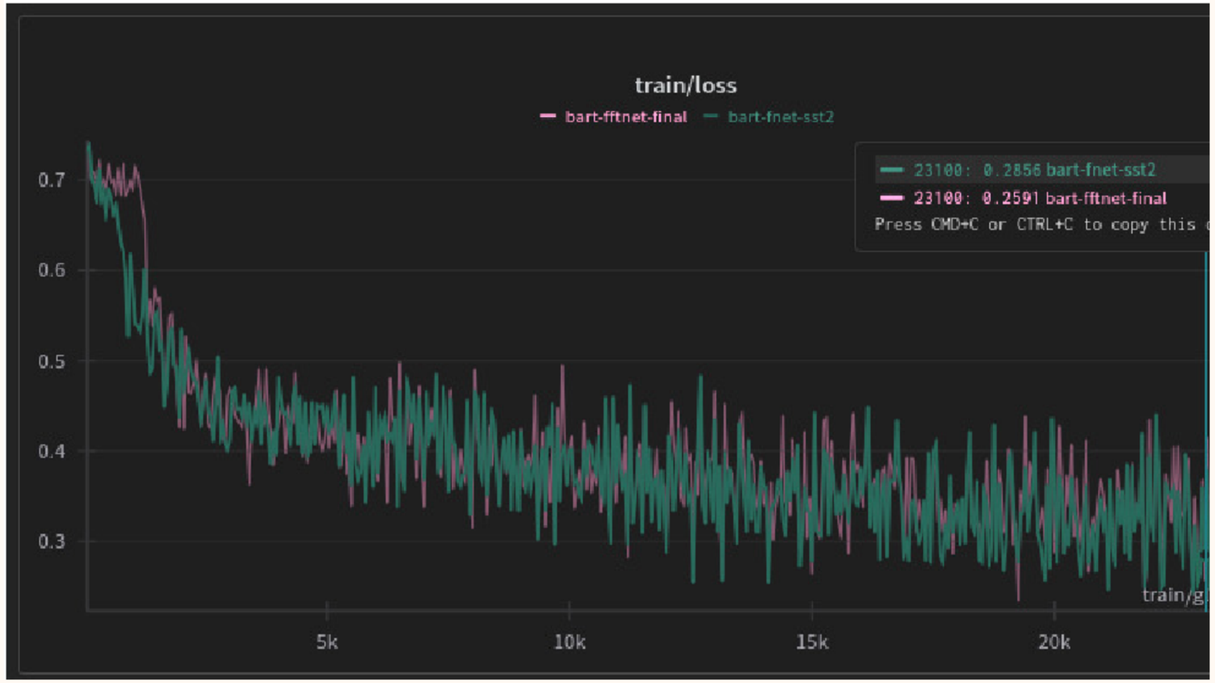
1. Benchmarking all the combinations on SWAG Dataset



- All models failed to produce satisfactory results .
- **Learning**: Attention can't be directly replaced with Fourier when the task demands strong contextual understanding.

2. Extending FFTNet to SST2

We tried extending the spectral filter from FFTNet onto the fnet encoder to perform better on SST-2 dataset .

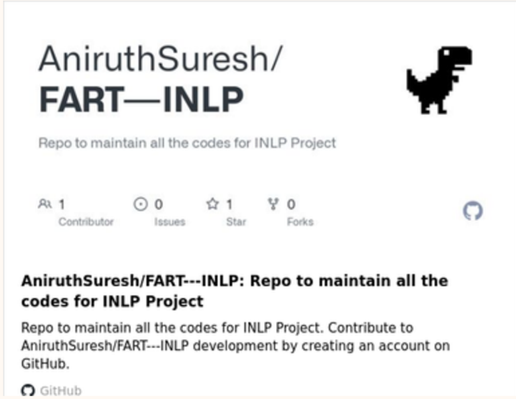


- It performed almost exactly same as the base FNET model
- **Possible explanation** : SST-2 is a sentiment analysis dataset, which may not benefit as much from the adaptive filtering technique

Codes and Submission

All active code, results, and run details are documented.

(As of end-submission, there are 7 active branches.)



THANK YOU

FOR YOUR ATTENTION:)