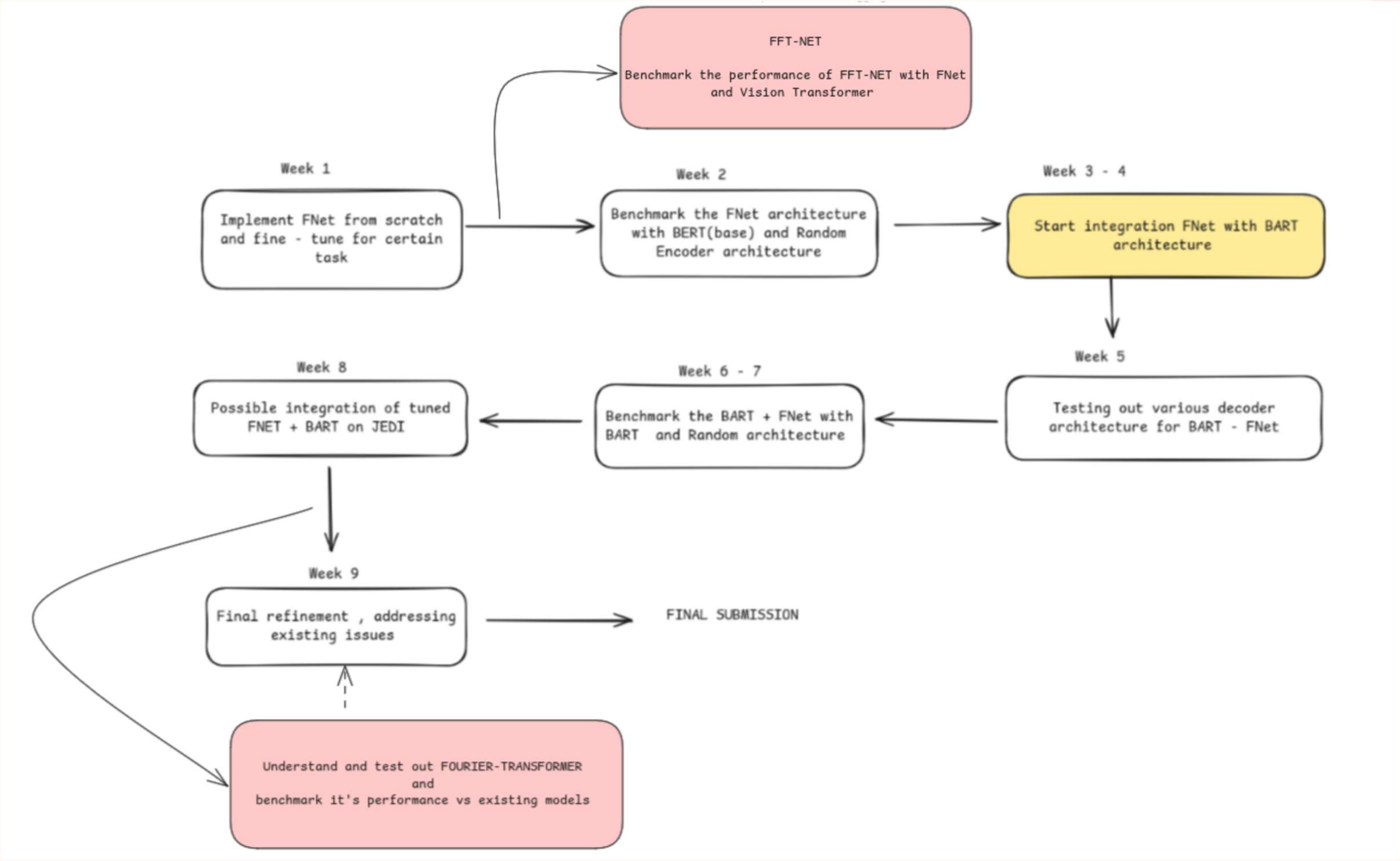
FNET + BART : IS FOURIER ALL YOU NEED ?

Samkit Jain , Aryan Garg , Aniruth Suresh

Project Outline



FFT-NET

Benchmark the performance of FFT-NET with FNet and Vision Transformer

Week 1

Implement FNet from scratch and fine - tune for certain task

Week 2

Benchmark the FNet architecture with BERT(base) and Random Encoder architecture

Week 3 - 4

Start integration FNet with BART architecture

Week 5

Testing out various decoder architecture for BART - FNet

Week 6 - 7

Benchmark the BART + FNet with BART and Random architecture

Week 8

Possible integration of tuned FNET + BART on JEDI

Week 9

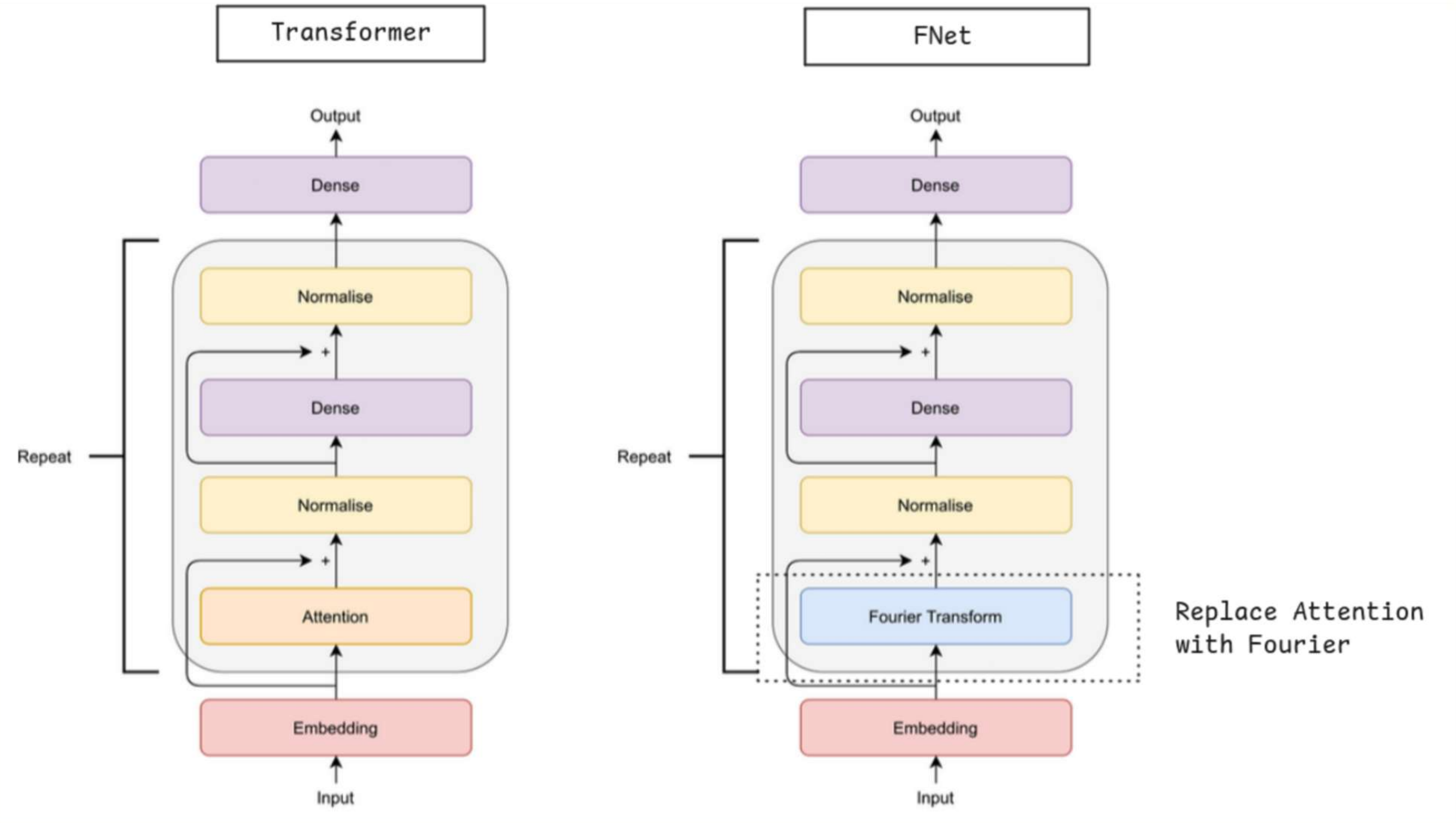
Final refinement , addressing existing issues

FINAL SUBMISSION

Understand and test out FOURIER-TRANSFORMER and benchmark it's performance vs existing models

F-Net Architecture

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

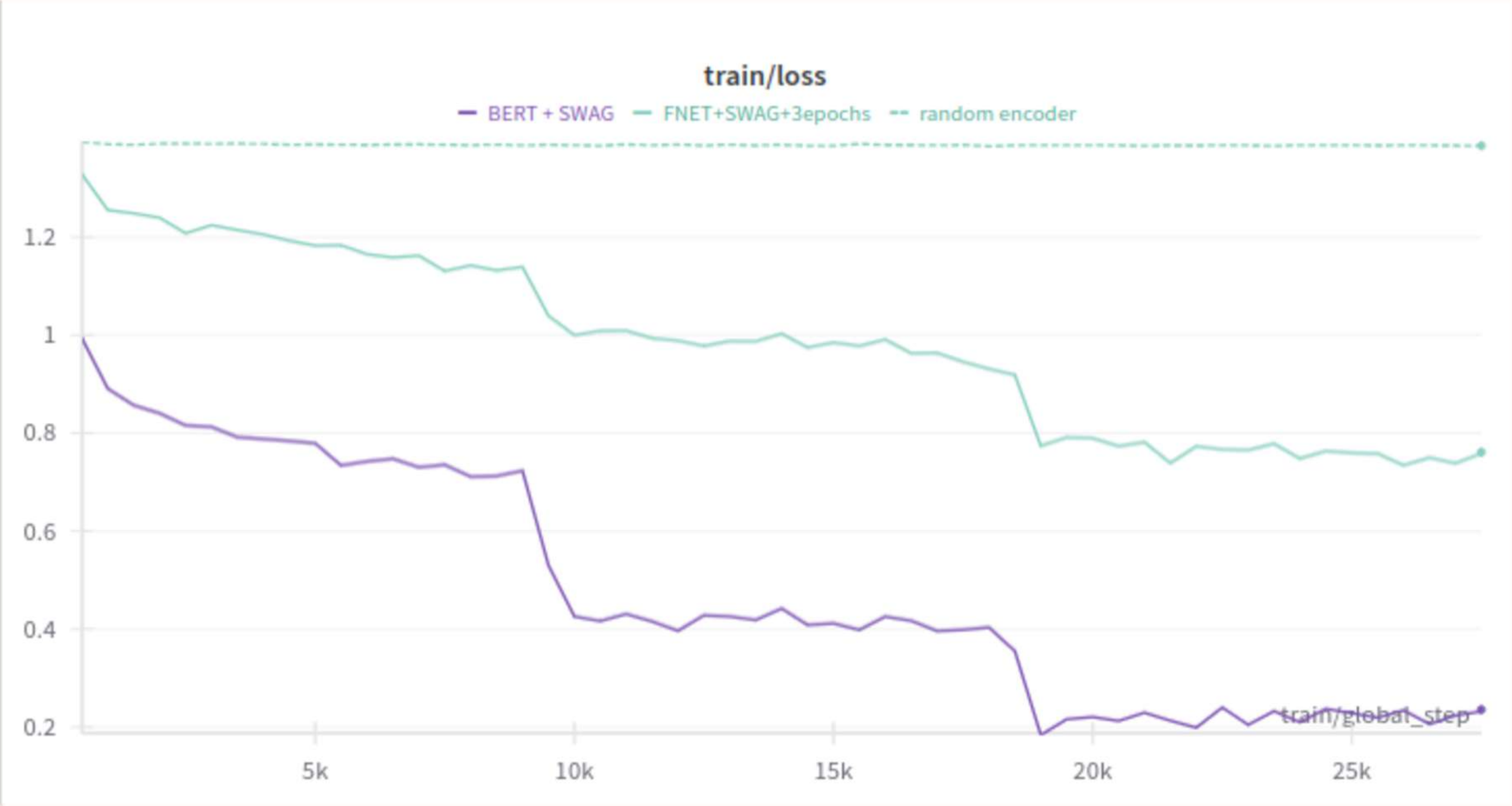


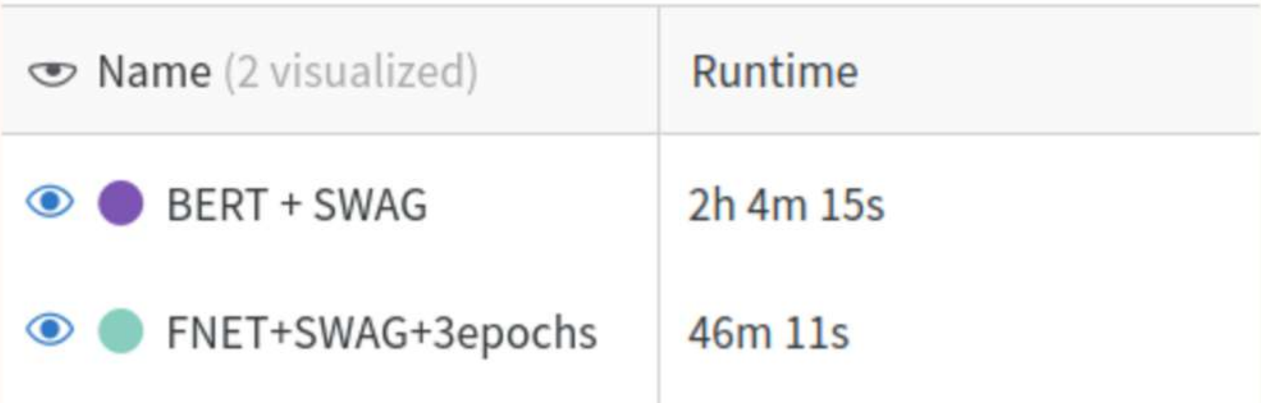
F-Net swaps the standard, computationally heavy self-attention layers found in Transformers.

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

3 models :

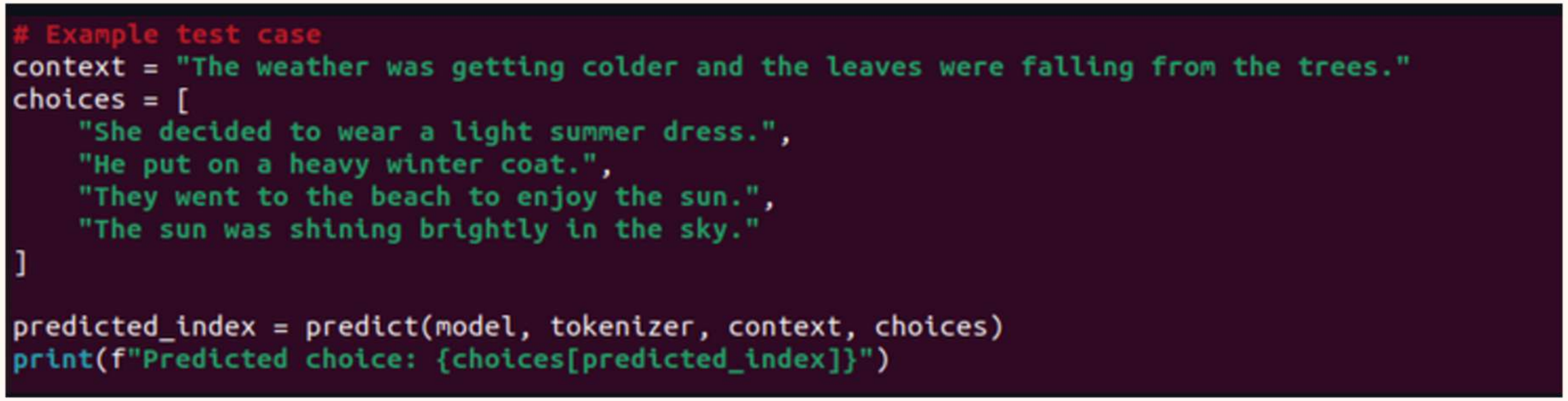
1. Fnet + SWAG
2. BERT + SWAG
3. Random (replaces attention by fixed matrices)

OBSERVATIONS :

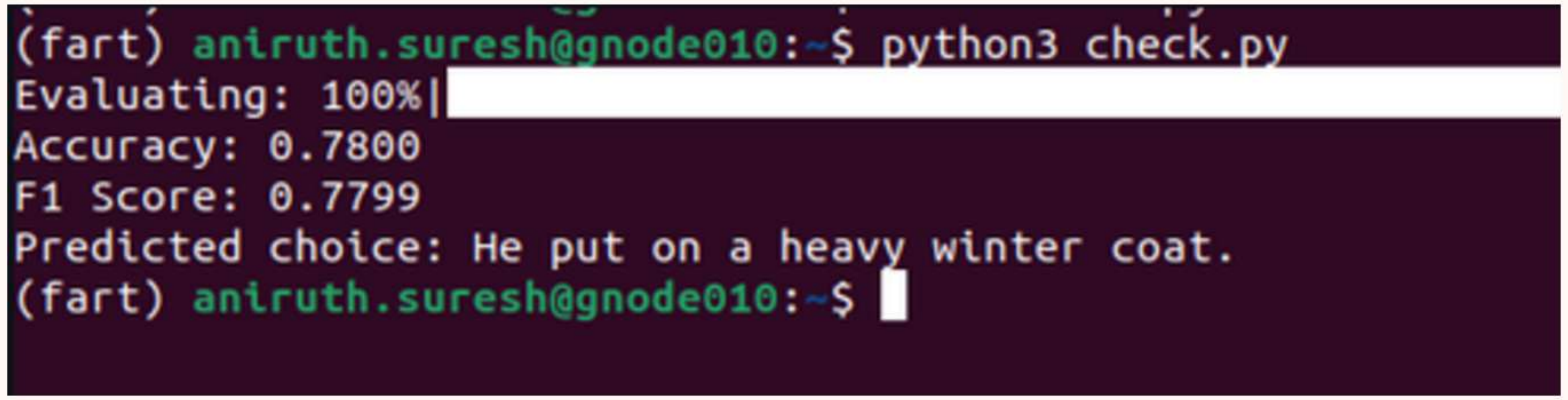
1. Clearly from the above graph , we observe that we can't just replace attention by any random fixed matrix .
2. This clearly signifies the importance using **Fourier Transform** .

Remember : FT has no learnable parameters !

FNet Benchmarking



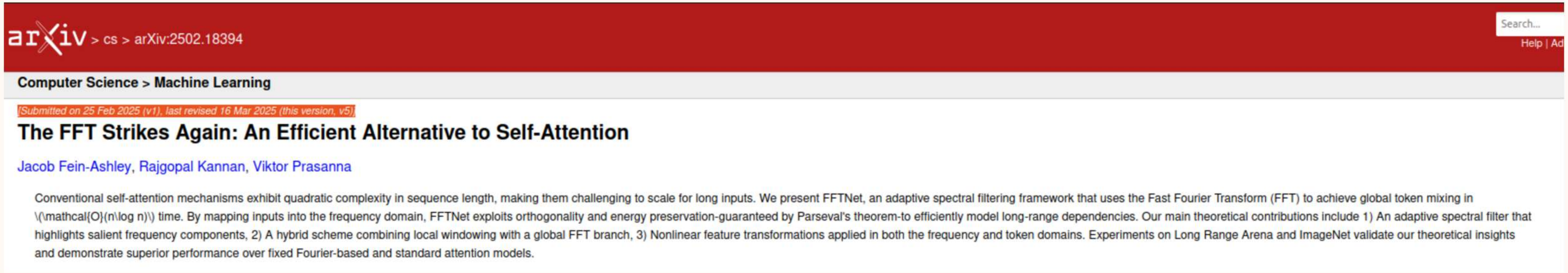
```
# Example test case
context = "The weather was getting colder and the leaves were falling from the trees."
choices = [
    "She decided to wear a light summer dress.",
    "He put on a heavy winter coat.",
    "They went to the beach to enjoy the sun.",
    "The sun was shining brightly in the sky."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

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 ?



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.

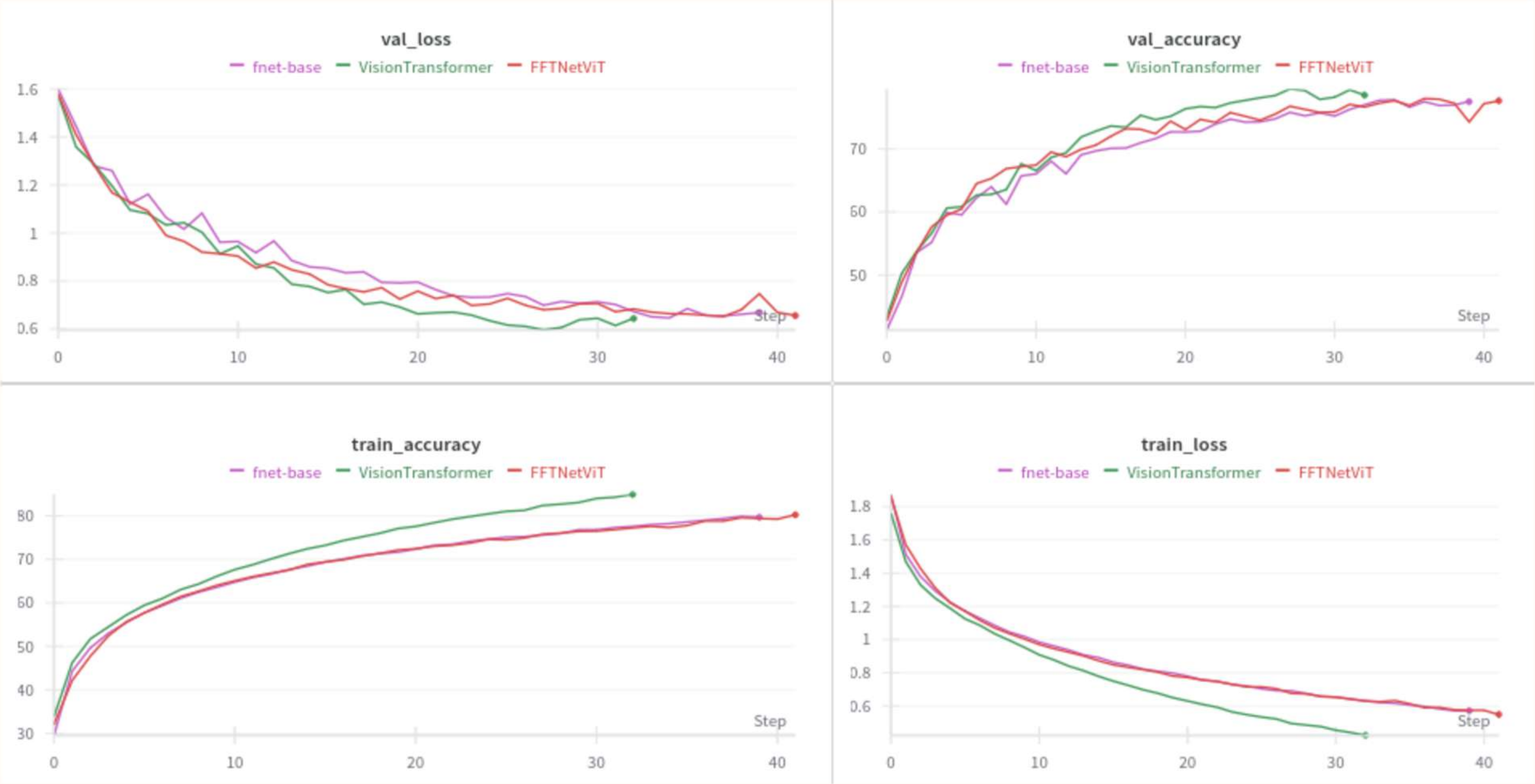
It introduces a learnable, context dependent filter in the frequency domain to dynamically emphasize or attenuate required frequency components, enhancing its performance over the fixed parameters of F-Net.

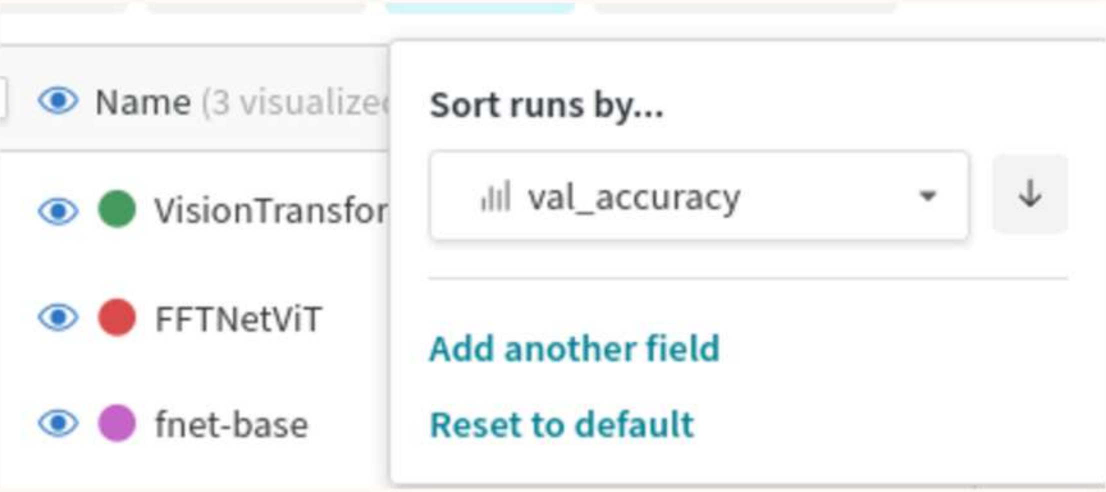
It also applies a non-linear activation function (modReLU) directly to the complex FT coefficients after filtering, resulting in better representation of token dependencies.

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




Name (3 visualize

VisionTransfor

FFTNetViT

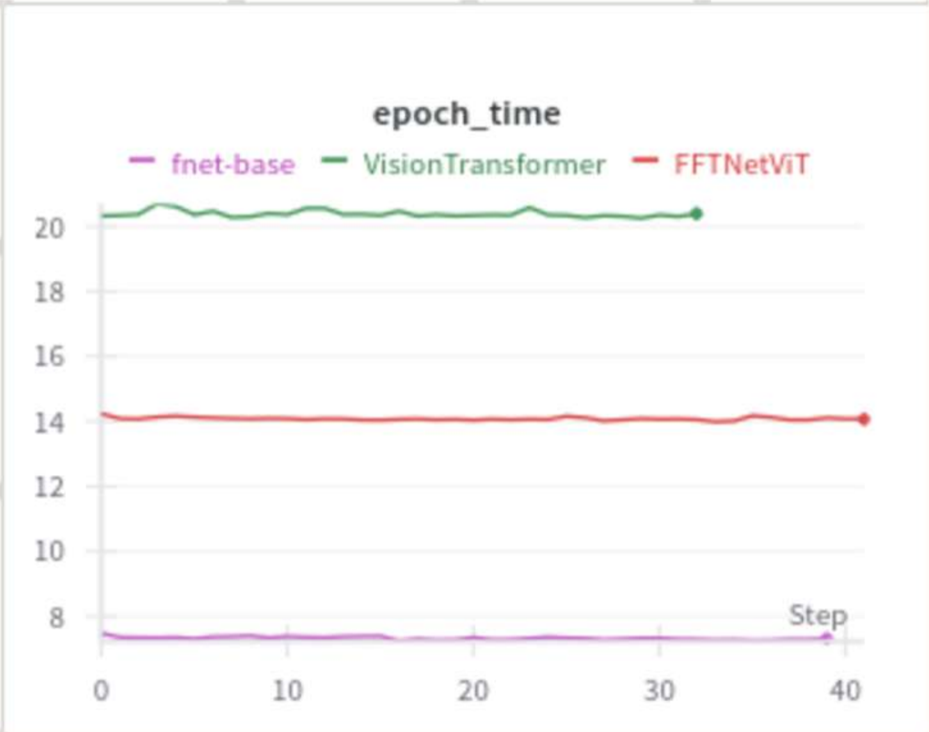
fnet-base

Sort runs by...

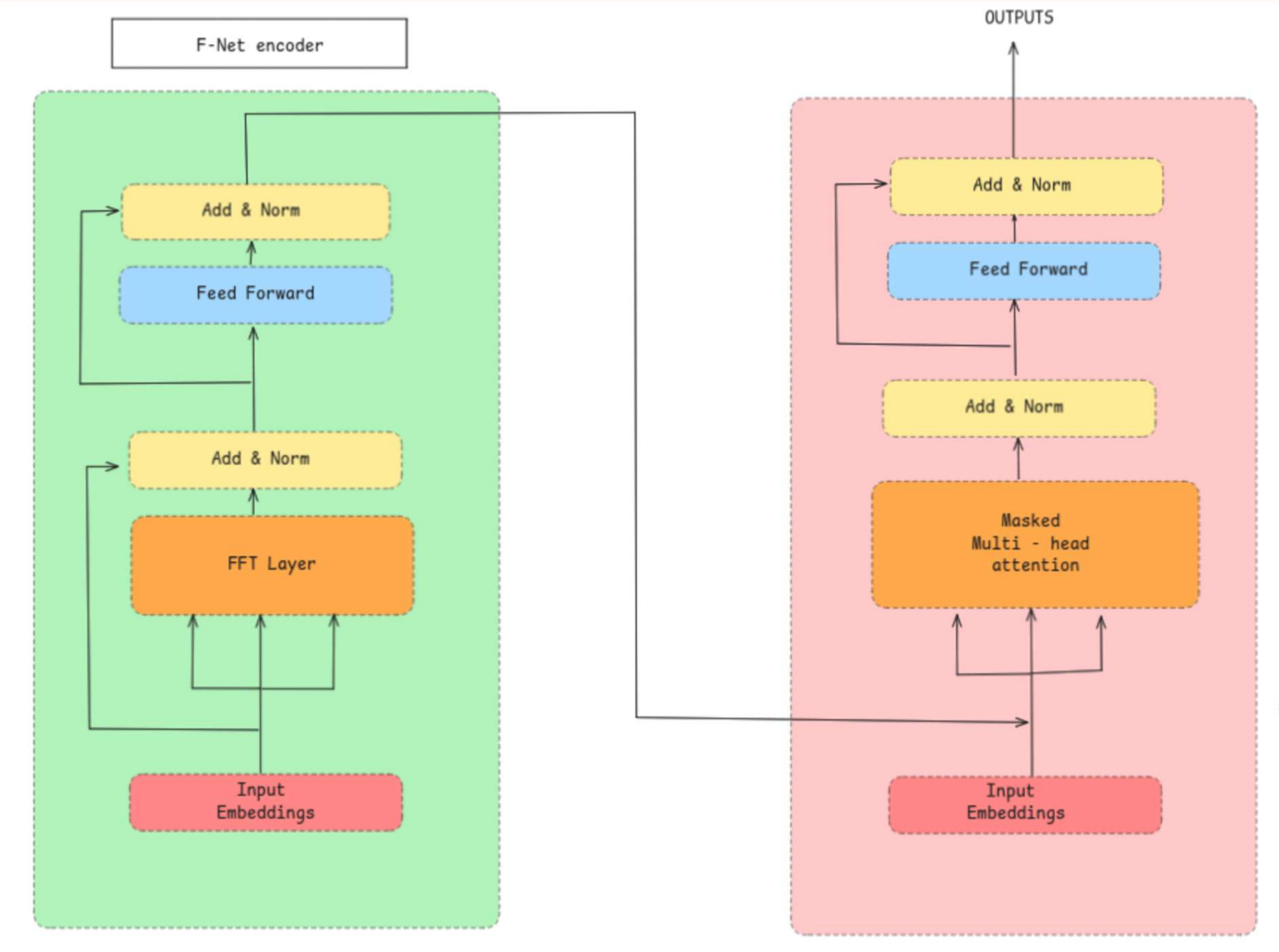
val_accuracy

Add another field

Reset to default

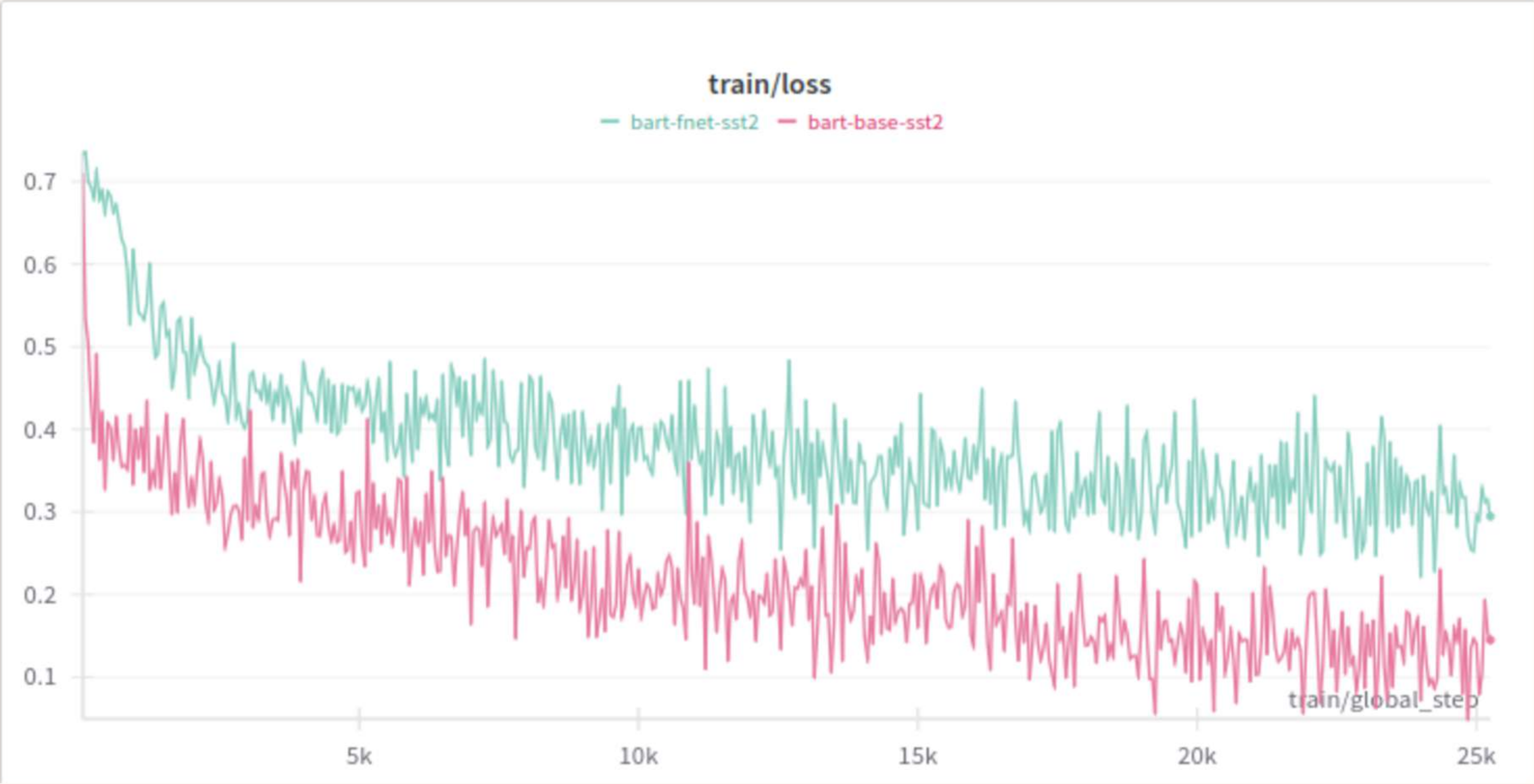


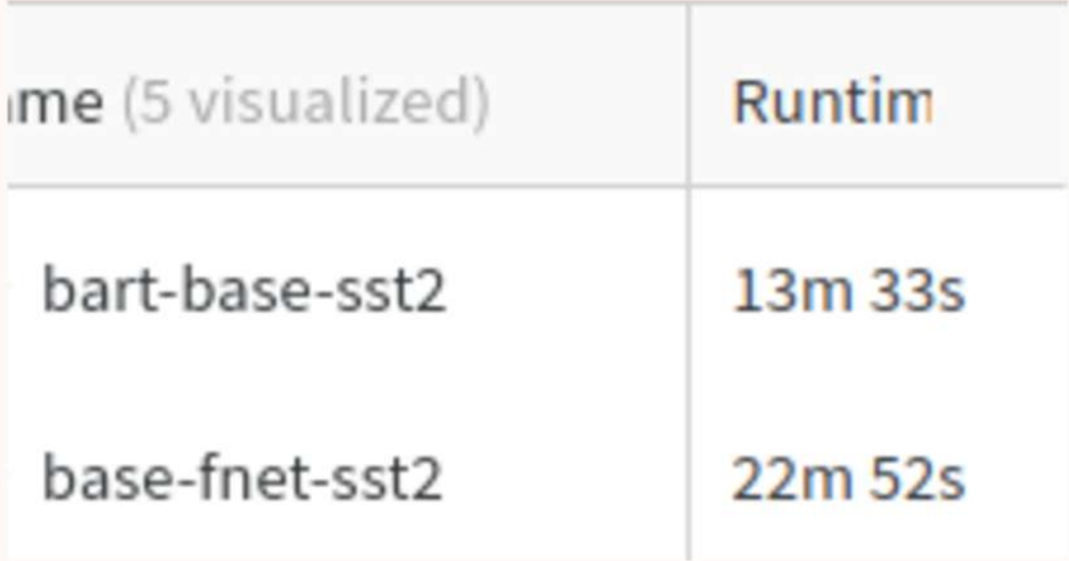
FNet + BART



FFT mixing incorporated in BART by replacing the attention heads by FFT Layers. (currently , we have tested replacing it in encoder . We further plan to extrapolate it to decoder stage as well) .

FNet + BART





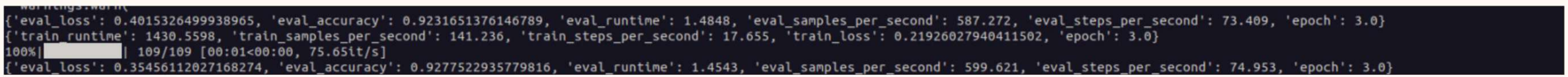
me (5 visualized)	Runtim
bart-base-sst2	13m 33s
base-fnet-sst2	22m 52s

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

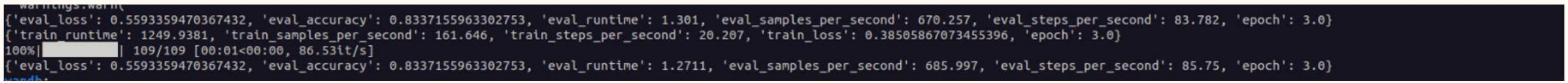
Time - Accuracy Tradeoff

FNet + BART

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

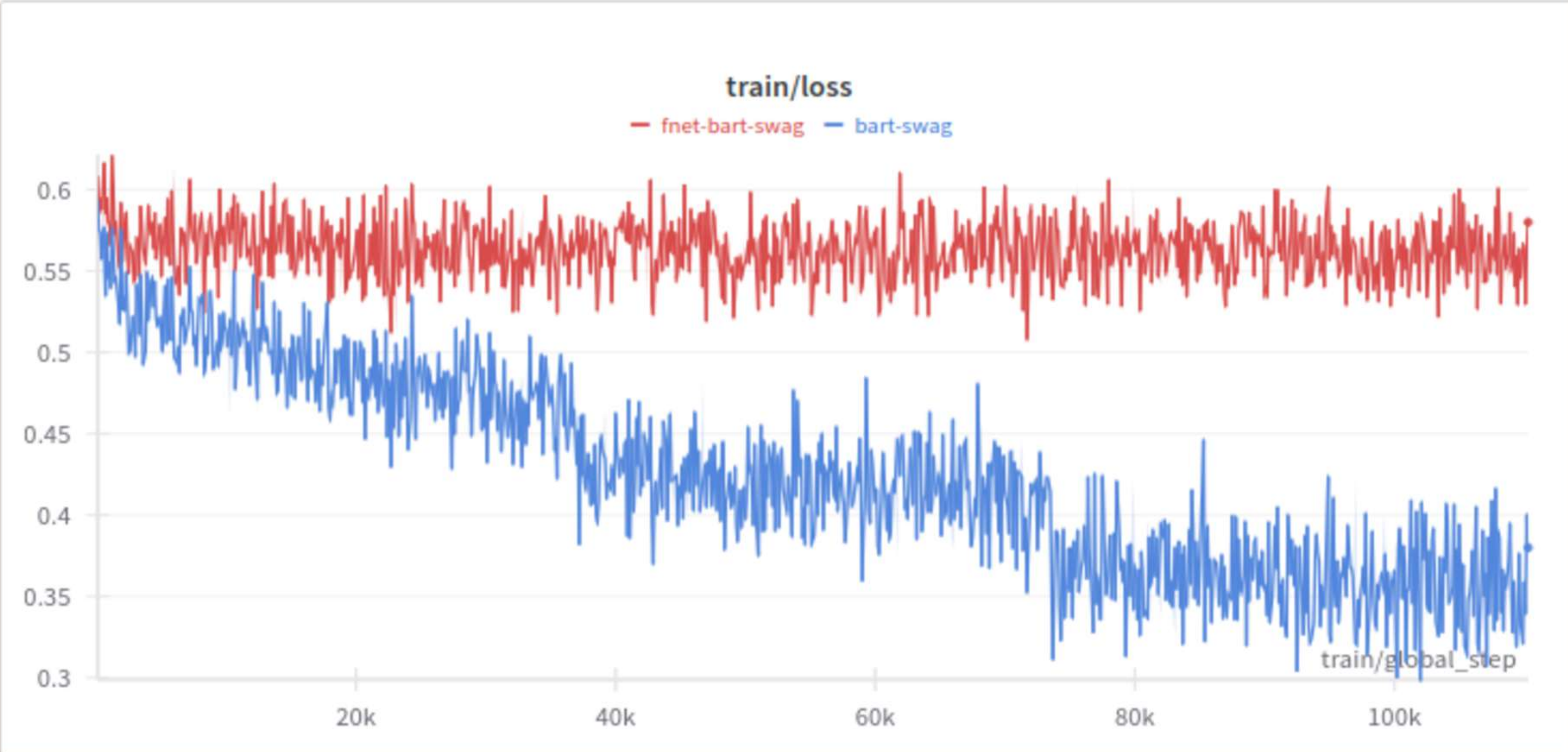


```
  warnings.warn(
{'eval_loss': 0.4015326499938965, 'eval_accuracy': 0.9231651376146789, 'eval_runtime': 1.4848, 'eval_samples_per_second': 587.272, 'eval_steps_per_second': 73.409, 'epoch': 3.0}
{'train_runtime': 1430.5598, 'train_samples_per_second': 141.236, 'train_steps_per_second': 17.655, 'train_loss': 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}
```



```
  warnings.warn(
{'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}
wandb:
```

FNet + BART

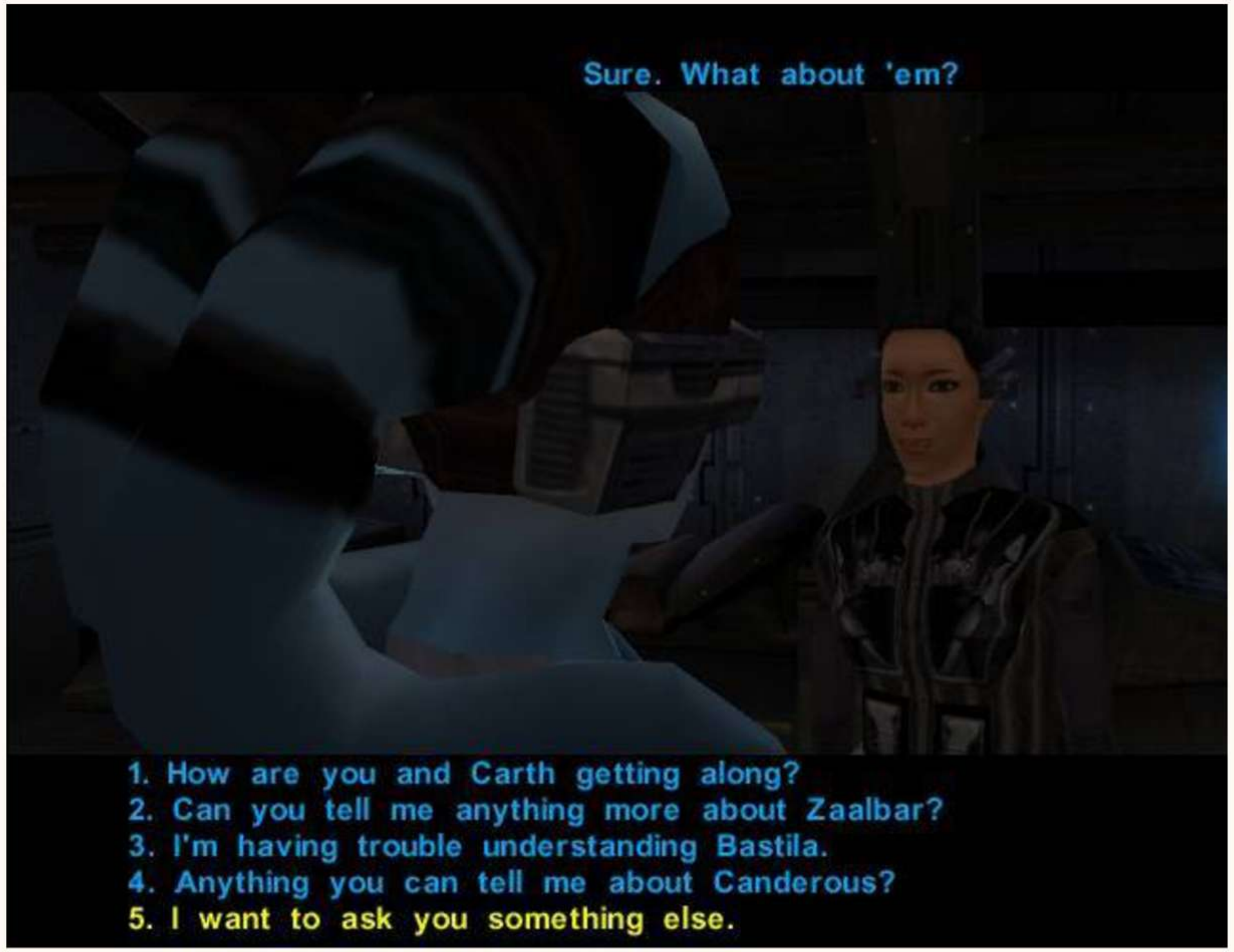


Reasons for poor performance : SWAG depends heavily on understanding context and making inferences, which often benefits from attention. By replacing the self-attention layer with Fourier Transforms the model may be **losing important semantic information** required for accuracy.

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 !!

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)

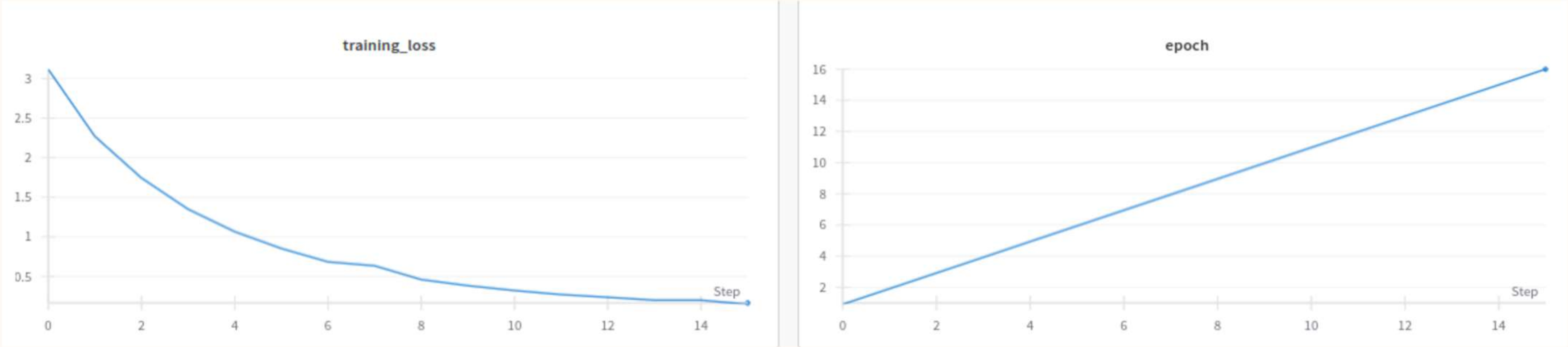


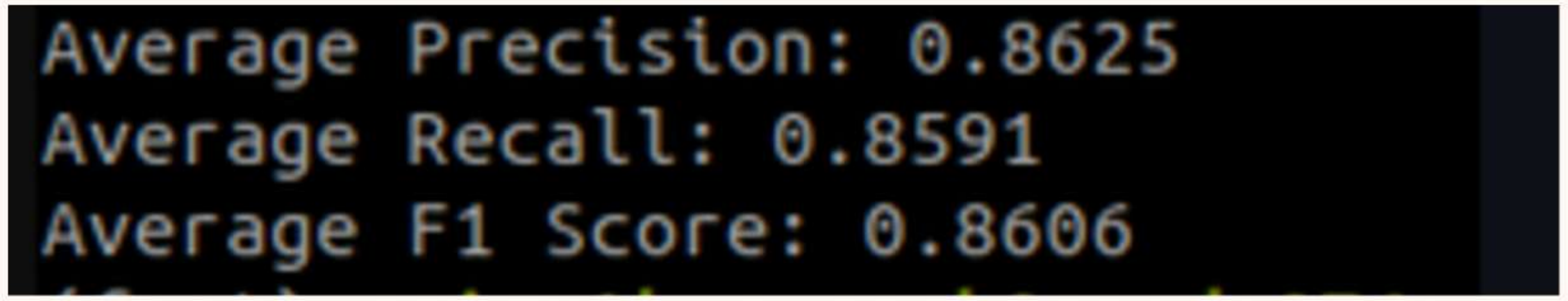
Sure. What about 'em?

1. How are you and Carth getting along?
2. Can you tell me anything more about Zaalbar?
3. I'm having trouble understanding Bastila.
4. Anything you can tell me about Canderous?
5. I want to ask you something else.

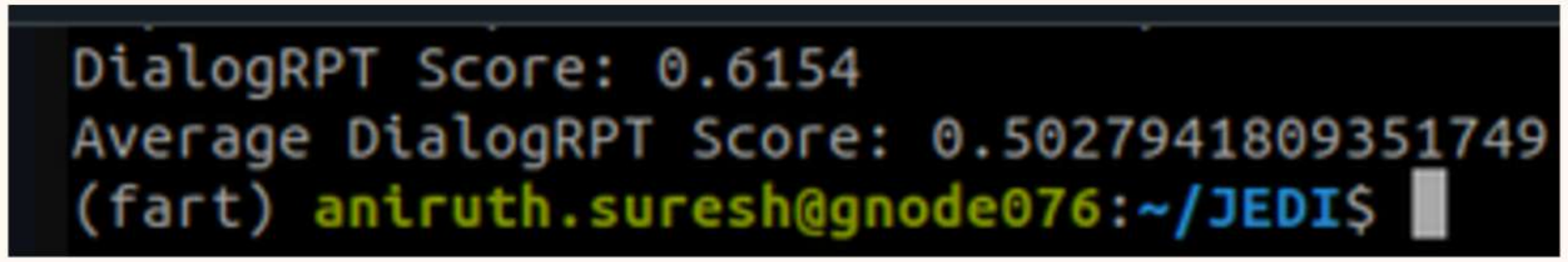
Key idea : Represent “Dialogue as a graph”

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 !





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



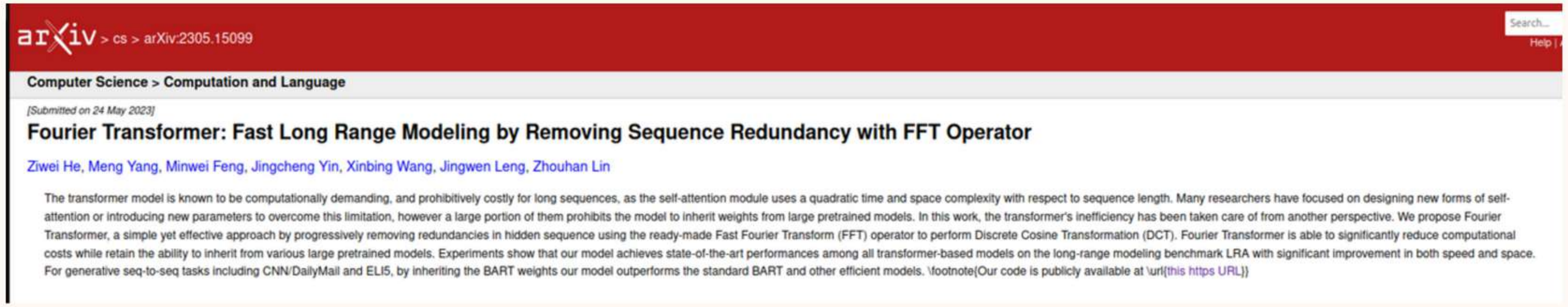
```
DialogRPT Score: 0.6154
Average DialogRPT Score: 0.5027941809351749
(fart) aniruth.suresh@gnode076:~/JEDI$ 
```

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

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)

Future Implementations

1. Plan on implementing **adaptive filtering techniques** similar to that used in FFTNet to make use of complex information from FFT instead of just taking real part .
2. Setup and benchmark Fourier Transformer which uses spectral filtering using Fourier Transform .



Computer Science > Computation and Language

[Submitted on 24 May 2023]

Fourier Transformer: Fast Long Range Modeling by Removing Sequence Redundancy with FFT Operator

Ziwei He, Meng Yang, Minwei Feng, Jingcheng Yin, Xinbing Wang, Jingwen Leng, Zhouhan Lin

The transformer model is known to be computationally demanding, and prohibitively costly for long sequences, as the self-attention module uses a quadratic time and space complexity with respect to sequence length. Many researchers have focused on designing new forms of self-attention or introducing new parameters to overcome this limitation, however a large portion of them prohibits the model to inherit weights from large pretrained models. In this work, the transformer's inefficiency has been taken care of from another perspective. We propose Fourier Transformer, a simple yet effective approach by progressively removing redundancies in hidden sequence using the ready-made Fast Fourier Transform (FFT) operator to perform Discrete Cosine Transformation (DCT). Fourier Transformer is able to significantly reduce computational costs while retain the ability to inherit from various large pretrained models. Experiments show that our model achieves state-of-the-art performances among all transformer-based models on the long-range modeling benchmark LRA with significant improvement in both speed and space. For generative seq-to-seq tasks including CNN/DailyMail and ELI5, by inheriting the BART weights our model outperforms the standard BART and other efficient models. \footnote{Our code is publicly available at \url{this https URL}}

3. Plan to modify the decoder architecture of BART and analyze the performance .
4. Integrate the FNet BART models on JEDI and compare and analyze the results .

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

(As of mid-submission, there are 6 active branches.)

AniruthSuresh/FART---INLP: Repo to maintain all the codes for INLP Project

Repo to maintain all the codes for INLP Project. Contribute to AniruthSuresh/FART---INLP development by creating an account on GitHub.

GitHub

THANK YOU

FOR YOUR ATTENTION:)