Exploring the Effect of Frequency Resolution in FNet

Greg Szumel and **Ghazal Khalighinejad** and **Rickard Stureborg** and **Sam Wiseman**
Duke University
{gks9, gk126, rs541, sam.wiseman}@duke.edu

Abstract

The recently introduced FNet model (Lee-Thorp et al., 2022) computes a two-dimensional discrete Fourier transform (DFT) of a sequence-length-by-hidden-dimension-sized representation of its input. Because it is equally efficient to compute the DFT of any reshaping of this input matrix, we investigate the extent to which increasing the frequency resolution in one dimension (at the expense of the other) affects task performance. We consider the LRA tasks (Tay et al., 2021) considered by Lee-Thorpe et al., as well as the more practical setting of using FNet as the encoder in a machine translation (MT) model. We find that frequency resolution has a marked task-dependent effect on performance, allowing us to largely outperform standard FNet on our tasks, and suggesting that resolution should be carefully tuned before deploying FNet.

1 Introduction

The FNet model, recently introduced by Lee-Thorp et al. (2022), is an encoder-only Transformer (Vaswani et al., 2017) that takes a major deviation from the Transformer's standard architecture. Mainly, FNet replaces the self-attention mechanism in a Transformer with a 2D Discrete Fourier Transform (DFT). The DFT is a deterministic operation which transforms an input vector x to output vector X, both of length N, by

$$X_k = \sum_{n=0}^{N-1} x_n * \exp(\frac{-i2\pi}{N} kn)$$

Notably, there are no learnable parameters within this sub-layer. Additionally, FNet represents an important innovation because the DFT scales sub-quadratically in the sequence length, unlike self-attention. FNet therefore offers the promise of a more efficient general-purpose transformer-like architecture.

We note that it is equally efficient to compute the 2D DFT of any reshaping of this matrix (i.e., increasing the number of columns at the expense of the rows, or vice-versa) and that reshaping will change the frequency resolution of the DFT in each dimension. We hypothesize that changing the frequency resolution may yield a more efficient token mixing than the standard input's mixing, particularly when flattening the input into a column of embedded tokens. The purpose of this paper is to explore whether there is any performance benefit to increasing or decreasing the frequency resolution in either dimension, a question not addressed by Lee-Thorp et al. (2022).

In addition to changing resolution by reshaping, which does not alter the number of elements in the matrix, we consider transformations that do alter the number of elements in the matrix, such as projecting up or down, or padding. We investigate the effect of these transformations on the FNet architecture as applied to the LRA tasks (Tay et al., 2021) considered in the original paper, and as an encoder in a standard transformer-based neural machine translation model. We find that reshaping has a marked effect on the performance of FNet on the tasks we consider, although no single reshaping appears to be optimal for all tasks. As such, we recommend that the reshaping be tuned per-task, as a hyperparameter. We also find that padding can improve performance for the Translation and short IMDB tasks, whereas projection tends to harm performance overall.

2 Methods

Below we outline several approaches to changing the dimensionality of the matrix consumed by the DFT in FNet, and thus the frequency resolution in at least one dimension. Whereas we are interested in changing the DFT's frequency resolution, we are not interested in changing the model's hidden dimension, which determines the size of the feed-

Proceedings of The Fourth Workshop on Simple and Efficient Natural Language Processing (SustaiNLP), pages 239–244

forward layers that follow the DFT. Accordingly, we always ensure that the DFT's output is mapped back to a sequence-length by hidden-dimension-sized matrix before being consumed by a feed-forward layer. Details are below.

Reshaping In FNet, the inputs to the DFT have dimension (excluding batch-size) $S \times H$, where S and H are sequence- and hidden-dimension respectively, and where S and H are powers of two. To reshape, we multiply S by a power of 2 (including negative powers), and divide H by that same power. We then contiguously reshape the matrix into one of dimension $S \cdot 2^i \times H \cdot 2^{-i}$. Note this transformation is implemented by the JAX/PyTorch library's `reshape` function. We choose all possible combinations of i such that $S \cdot 2^i$ and $H \cdot 2^{-i}$ are both integers. We include both fully flattened combinations (where H or $S = 1$) for completeness, although these transformations are equivalent.

Projection Reshaping maintains the same total number of elements in the input matrix, making it impossible to change the resolution in only one dimension. Projection, on the other hand, allows us to hold the resolution in one dimension constant while changing the the other's. In our experiments, we take the optimal power-of-2 reshaping as described in the previous paragraph, and then project one of its hidden or sequence dimension up or down. Specifically, we linearly project the pre-DFT input and then reshape it into the desired dimension. After taking the DFT, we undo the prior reshaping and project back to the original dimension. See Appendix A for more details.

Padding Rather than projecting, we may also pad the input matrix with zeros, along either the sequence or hidden dimension. In the case of padding the hidden dimension, we pad before taking the DFT and simply discard the extra columns after taking the DFT and before the subsequent feed-forward layer. In the case of padding the sequence-length, we simply pad the encoder input directly. See Appendix C for a discussion on time-complexities from padding.

3 Experiments

We evaluate on the Long Range Arena (LRA) tasks (Tay et al., 2021), also used in Lee-Thorp et al. (2022). Due to training instability, we only report on image-classification (CIFAR), text-classification (IMDB), and document matching (Matching). We report all LRA results as an average over three random seeds. In order to evaluate FNet in a more conventional NLP setting, we evaluate it as an encoder on the IWSLT14 English to German translation benchmark (Cettolo et al., 2014), and on a 'short' IMDB task. We modify the standard IWSLT14 task to only include examples shorter than 64 tokens, and pad all remaining examples to 64 tokens. See Appendix C for a discussion on time-complexity. To construct the short-IMDB task, we use the LRA's IMDB codebase but leverage the non-byte tokenization scheme, where each word is enumerated and is represented by its index. We also truncate and pad examples to have input length of 500 tokens. For IWSLT14 and the Short-IMDB tasks, we run experiments on a single seed and report the results directly. All results are reported as the optimal validation scores obtained during training.

Proj. dim. scale	D-Model	Seq-Length
Scale $\frac{1}{4}$	0.344	0.325
Scale $\frac{1}{2}$	0.390	0.318
Base (8, 4096)	0.422	0.322
Scale 2	0.409	0.314
Scale 4	0.419	0.313

(a) CIFAR projection experiments

Proj. dim. scale	D-Model	Seq-Length
Scale $\frac{1}{4}$	0.691	0.569
Scale $\frac{1}{2}$	0.702	0.575
Base (2048, 128)	0.693	0.568
Scale 2	0.566	0.572
Scale 4	0.567	0.593

(b) IMDB projection experiments

Proj. dim. scale	D-Model	Seq-Length
Scale $\frac{1}{4}$	0.624	0.623
Scale $\frac{1}{2}$	0.625	0.616
Base (2048, 256)	0.623	0.630
Scale 2	0.618	0.619
Scale 4	0.612	0.622

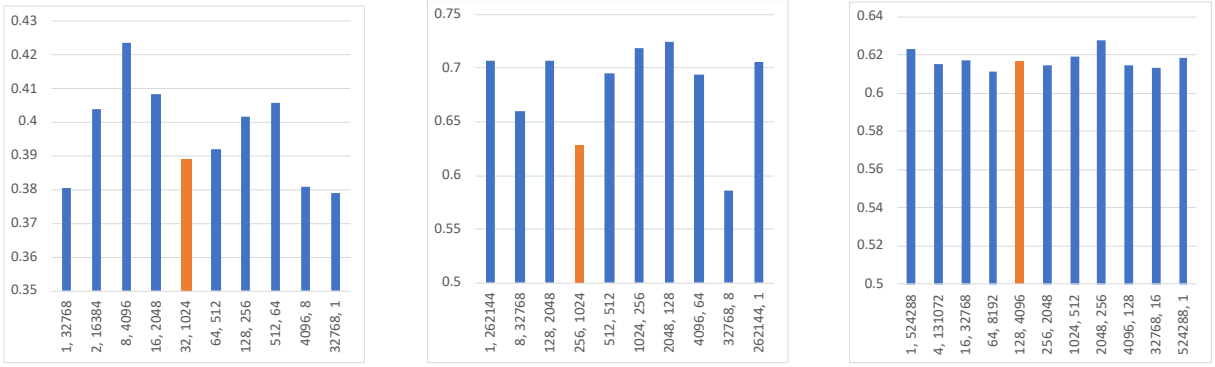
(c) Matching projection experiments

Table 1: Projection of highest performing reshaping from the reshaping survey on LRA. Base dimensions are [D-model, Sequence length].

4 Results

4.1 LRA

Figure 1 outlines the results of our reshaping survey across the LRA tasks IMDB, CIFAR, and Match-



(a) CIFAR (b) IMDB (c) Matching

Figure 1: Results of reshaping survey on LRA tasks. Orange bars denote the unchanged dimensions. The first dimension is the reshaped hidden dimension, second dimension is the sequence length. Flattened reshapings should be computationally equivalent and are included for completeness.

ing. See Appendix B.1 for results on all tasks and reshapings. We observe optimal 'aspect-ratios' which are distinct from the unaltered ratios. For CIFAR, reshaping to have a larger sequence-length improves performance by a number of percentage points. IMDB and Matching also show local optima at either end of the spectrum, which provides the largest resolution to a single dimension.

Table 1 outlines the results of Projection on the LRA tasks. For CIFAR, IMDB, and Matching, we set the optimal reshaped dimension to $[8, 4096]$, $[2048, 128]$, and $[2048, 256]$, respectively. We then project either the hidden or sequence dimension by a fixed scale, denoted by the "Projection dimension scale". For example, a scale of $\frac{1}{2}$ on CIFAR produces a DFT input shape of $[4, 4096]$ when projecting D-Model, while the same scale has a DFT input shape of $[8, 2048]$ when projecting Seq-Length.

We see that projecting before taking the DFT reduces model performance. Further, projecting the sequence length almost always results in lower performance than projecting the hidden dimension.

Experiments on the padding dimension are in Table 2. As with the projection experiments, we see that padding along both the sequence and the hidden dimension typically lowers model performance. However, increasing the resolution with padding can improve performance on the IMDB task for certain padding amounts.

Padded dim. scale	**D-Model**	**Seq-Length**
8, 4096 (no pad)	0.425	0.425
Scale 2	0.408	0.426
Scale 4	0.386	0.403

(a) CIFAR padding experiments

Padded dim. scale	**D-Model**	**Seq-Length**
2048, 128 (no pad)	0.726	0.726
Scale 2	0.738	0.737
Scale 4	0.730	0.708

(b) IMDB padding experiments

Padded dim. scale	**D-Model**	**Seq-Length**
2048, 256 (no pad)	0.638	0.638
Scale 2	0.621	0.621
Scale 4	0.626	0.624

(c) Matching padding experiments

Table 2: Padded of strongest dimensions in LRA.

4.2 Translation

The results of the reshaping survey on the translation task are given in Figure 2. In contrast to the LRA experiments, we observe that the performance of the base DFT input shape is greater than every tested reshaping. Notably, FNet is nearly comparable to Transformer after fixing input length.

Table 3 shows the results of the projection experiment. We do not project the sequence length due to its low performance on LRA. We again select the optimal reshaping, which for translation is $[512, 64]$. Here we see the original shape outperforms all projections on the hidden dimension.

Table 4a shows the results of padding along the sequence length dimension. Here, we observe modest increases when padding along the sequence length, although there is a point at which increasing the length hinders performance. Table 4b shows the results of padding along the hidden dimension. We do not observe any hidden-dimension padding that surpasses the baseline model performance.

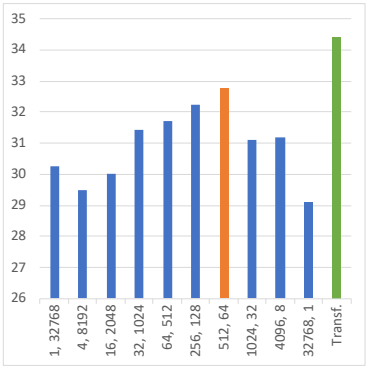


Figure 2: Results of reshaping survey on the Translation task. Orange bars denote the unchanged dimensions. The first dimension is the reshaped hidden dimension, second dimension is the sequence length. Flattened reshapings should be computationally equivalent and are included for completeness.

Projected dim. size	BLEU
128, 64	27.84
256, 64	28.85
512, 64 (base shape)	30.11
1024, 64	30.03
2048, 64	25.92

Table 3: Projection of strongest dimensions in translation.

4.3 Short IMDB

Figure 6 displays the results of the reshaping survey on the Short-IMDB task. Like the Translation results, the model using unaltered DFT input has higher accuracy than the other tested models. Interestingly, reshapings that had high accuracy in the long-IMDB task do not appear to transfer to the short-IMDB task.

5 Conclusion

It is clear that tuning the FNet's DFT input dimension can affect model performance. In LRA, between optimal and base input dimensions, we see that CIFAR, IMDB, and Matching all increase performance by 9%, 15%, and 2%, respectively. However, altering does not appear to help for all tasks. In Translation and short IMDB, altering the input dimension to the DFT layer lowers overall performance. There could be several reasons for this performance degradation.

First, performance variance could be due to input length. The translation task uses a max sequence length of 64, which is significantly shorter than the LRA tasks, which had a minimum of 1024. If true,

Padded dim. size - seq len	BLEU
64 (unpadded)	32.77
128	33.23
256	33.28
512	29.43

(a) Sequence length

Padded dim. size - hidden dim.	BLEU
256 (unpadded)	32.77
512	32.02
700	32.64
1024	32.04

(b) Hidden dimension

Table 4: Padded of strongest dimensions in LRA.

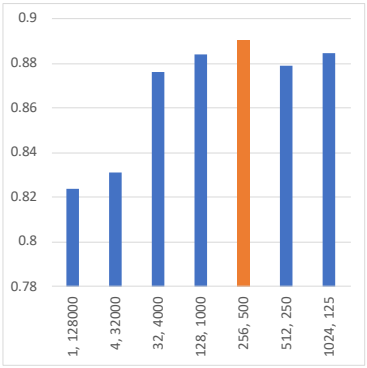


Figure 3: Results of reshaping survey on the 'short IMDB' task. Orange bars denote the unchanged dimensions. The first dimension is the reshaped hidden dimension, second dimension is the sequence length.

it may be harder to tune DFT input shapes with shorter sequence-lengths.

Second, certain tokenization methods may be more amenable to tuning the DFT input dimension. We observe a performance boost through the reshape survey on IMDB and Matching, both of which use byte-level tokenization. Therefore, reshaping may be more potent on tasks that use a byte-level tokenization.

We have not yet characterized the mechanism for why FNet performance can be affected by altering the input dimension to the DFT. We believe that future work on tokenization techniques, base-sequence-length, and testing on additional tasks could be ideal routes to further explore why adjusting the input shape to the DFT can alter the model's overall performance.

References

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th IWSLT evaluation campaign. In *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pages 2–17, Lake Tahoe, California.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.

A Example of projection methodology

Suppose we wish to project an input of reshaped size [32, 1024] to [32, 2048]. Let's suppose the original input had [64, 512]. Before taking the DFT, we take the following steps:

- Project the original input's hidden dimension by 2, yielding [64, 1024]
- Reshape to [32, 2048]
- Take DFT
- Un-reshape back to [64, 1024]
- Project back down to [64, 512]

In the case that we would like to reshape to [64, 1024], we now project the sequence length. Before and after each projection, we transpose the input to $[H, S]$.

In the case of padding the hidden dimension, replace the Projection layers with padding/cropping operations. We pad initially, then crop back down at the end.

Model	Time complexity
BERT	$2n^2d_h + 4nd_h^2$
FNet (matrix)	$n^2d_h + nd_h^2$
FNet (FFT)	$nd_h[log(n) + log(d_h)]$

Table 5: Number of mixing layer operations (forward pass). n is the sequence length and d_h is the model hidden dimension.

B Additional results

B.1 LRA Survey

See Figure 4 for the full results on the Survey task across all LRA tasks. Again, Pathfinder and PathfinderX models tended to have relatively unstable models (some models not finding strong performance, at 50% accuracy), so we excluded them from further testing. Additionally, ListOps does not appear to improve performance based on altering the hidden-seq ratio, so it was also excluded from further testing.

B.2 Translation survey

Figure 5 shows the full results of altering the aspect ratio into the DFT for the Translation task.

B.3 Short IMDB Survey

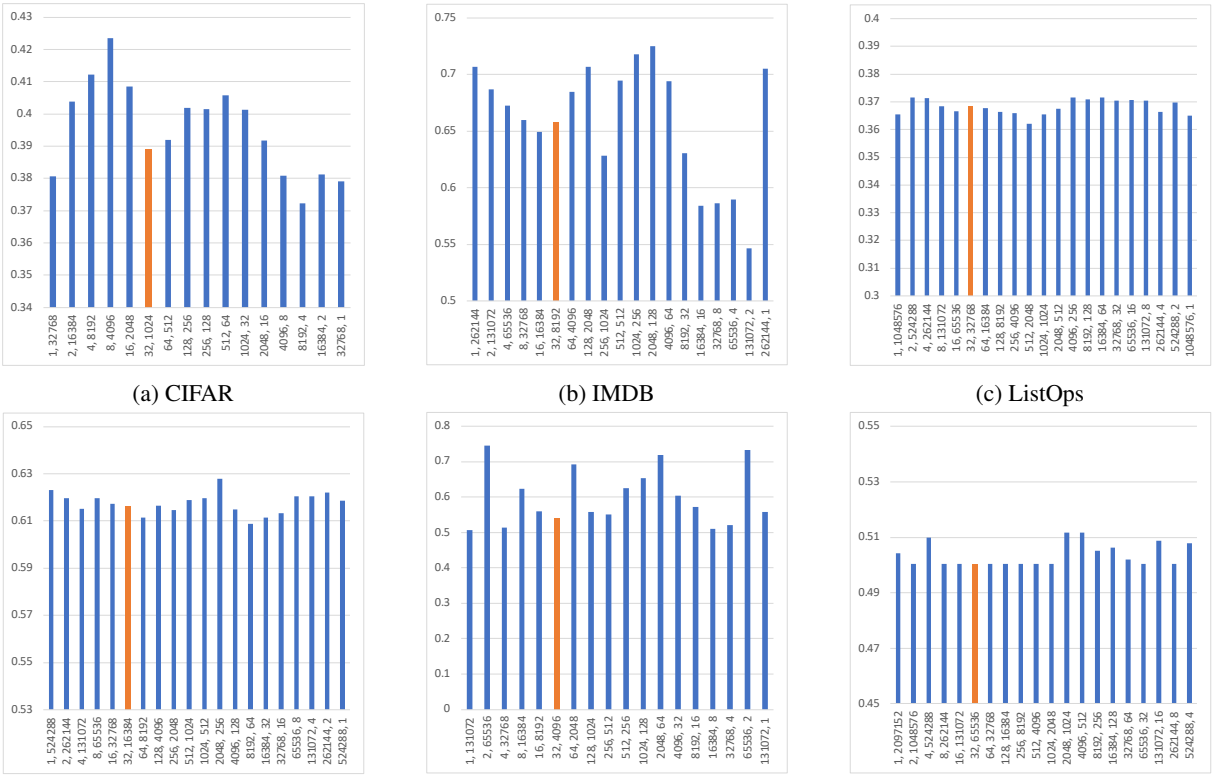
Figure 6 shows the full results of altering the aspect ratio into the DFT for the short-IMDB task.

C Time complexity of reshaping and padding

Lee-Thorp et al. (2022) assembled the rough time complexity of FNet compared to standard the standard transformer, which is listed in 5.

Suppose that for some reshaping we have sequence-length l and hidden dimension d_h. The time-complexity in the FFT case would be $d_h l(\log(l) + \log(d_h)) = d_h l \log(d_h l)$. We can see here that the no reshaping would be more time-intensive than another if the product of l and d_h is fixed.

However, we can see that increasing the sequence length (or hidden dimension) by padding has time complexity of $n \log n$ for padded length n. Padding will also increase the time complexity by n^2 in decoders, where the attention-mechanism is still present.



(a) CIFAR (b) IMDB (c) ListOps

(d) Matching (e) Pathfinder (f) PathfinderX

Figure 4: Results of varying hidden dimension and sequence length dimensions into the Fourier Transform. Orange lines denote the unchanged dimensions. First dimension is hidden dimension, second dimension is the sequence length. Each run is the average of the maximum validation BLUE score, taken as an average over 3 random seeds.

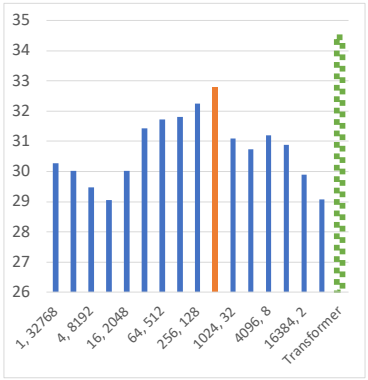


Figure 5: Results of varying hidden dimension and sequence length dimensions into the Fourier Transform. Orange bars indicate the unchanged dimension scales. The first dimension on the x axis is the hidden dimension, second dimension is the sequence length.

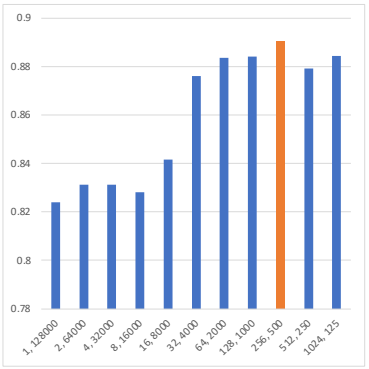


Figure 6: Results of reshaping survey on short IMDB