CS4782 Final Report

Angela Cui, Vipin Gunda, James Kim, Derek Liu, Oliver Lopez

1. Introduction

The paper we chose to reproduce was “A Time Series is Worth 64 Words: Long-term Forecasting with Transformers” by Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam.

Accurate long-term time series forecasting is essential across domains such as energy, healthcare, and finance. However, standard state-of-art deep learning models such as Transformers face limitations when applied to time series, particularly with long input sequences. Their self-attention mechanism scales quadratically with sequence length, restricting the model’s ability to learn long-range dependencies efficiently. To address this, the authors propose PatchTST, a Transformer-based model tailored for multivariate time series and self-supervised representation learning based on patching and channel independence, which involves dividing time series into subseries-level patches to capture local semantic information while reducing sequence length and processing each univariate time series separately with shared weights, respectively.

2. Chosen Result

Our project focused on reproducing and extending two specific components of the original paper:

1. Randomized Mask Ratios in Self-Supervised Learning

The self-supervised PatchTST model masks a fixed percentage of patches during pretraining. We explored whether randomizing the mask ratio would encourage robustness and prevent overfitting to a fixed masking pattern. Our hypothesis was that this would yield a stronger self-supervised representation, more adaptable for downstream tasks like forecasting.

2. Improved Patch Embeddings via 1D Convolutions

In the original architecture, each patch is embedded using a linear projection. We hypothesized that replacing this linear layer with convolutional layers could improve performance by better capturing local temporal dependencies within each patch. Working off the fact that CNNs work well for images, we reasoned that by convolving before flattening, the model may gain richer semantic information, potentially improving both representation quality and downstream forecasting accuracy.

Upon investigating existing works, we found [2] and decided to use it as a baseline for our embedding experiments.

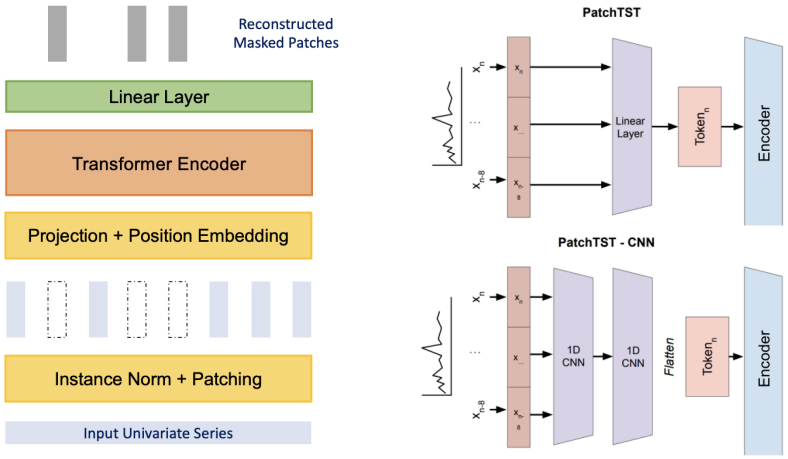

Figure 1: (Left) Self-supervised transformer backbone (Right) Top is original patch embedding, Bottom uses 1D CNN layers to extract features from a given patch

3. Methodology

To investigate the impact of variable masking rates in self-supervised learning, we modified the PatchTST pretraining pipeline to apply a randomized masking ratio between 0.2 and 0.6 during each forward pass. For every batch, a new mask ratio was uniformly sampled from this range and used to randomly select a proportion of patch embeddings to be masked. These masked patches were then replaced with learnable mask tokens, and the model was trained to reconstruct the original values. Aside from this randomized masking strategy, all other aspects of the pretraining and evaluation process—such as loss functions, optimizer

settings, and downstream fine-tuning—remained consistent with the original PatchTST setup. Since the task was to reconstruct the missing patches, no attention rescaling or architectural modifications were required.

Due to computational resource constraints, we used a smaller context window and a reduced-size Transformer architecture. While this scaled-down model is supported by the original paper for ablation studies and efficiency comparisons, the full model is used in the paper to achieve state-of-the-art results. Similarly, we trained for a limited number of epochs—20 for pretraining, 10 for linear probing, and 20 for full fine-tuning—whereas the original paper trains on the order of 100 epochs. Our goal was to evaluate the effects of randomized masking rather than to reproduce the highest possible benchmark performance.

Similarly, our investigation of the embedding strategy largely followed the existing structure from [2]. We used the same Transformer architecture across experiments (after reinitializing the weights) in order to compare the effectiveness against a consistent baseline. The model was trained from scratch on each dataset using a linear layer, as in the original paper, and then trained from scratch with two convolutional layers. We used a fixed kernel_size of 3 with a hidden layer of 8 channels, and an output layer of 32 channels. ReLu was applied to the activations of each layer. While this is only a small subset of reasonable configurations, we were not able to explore other options since full experiments could take several hours.

4. Results & Analysis

To assess the effectiveness of randomized mask ratios in self-supervised learning, we conducted experiments on the ETTH1 and ETTH2 datasets using a three-stage pipeline: (1) pretraining, where we compared fixed and randomized mask ratios, (2) linear probing, where only the encoding and decoding layers were updated while the Transformer backbone was frozen, and (3) full fine-tuning, where the entire model was unfrozen and trained end-to-end. During pretraining, we observed that the randomized masking strategy resulted in slightly higher (but not significantly worse) training and validation loss compared to fixed masking across both datasets. This is likely because the average random mask ratio hovered around 0.4—matching the fixed setting—but the variability introduced harder examples in some iterations, making the training landscape more challenging. In both the linear probing and fine-tuning stages, there were individual epochs where one method outperformed the other, but overall, both approaches exhibited highly similar convergence patterns. The trends suggest that given sufficient training time, both masking strategies would converge to comparable final performance, reinforcing the idea that randomized masking does not offer significant gains over the fixed approach in this setting.

Our experimentations with convolutional embeddings had mixed results. They never yielded more than, at best, marginal improvements in model performance. However, in some experiments, they greatly improved the speed of convergence. However, on other datasets, they caused almost immediate overfitting. Although we did not explicitly predict this behavior, it makes sense upon further analysis. The feature length of the datasets varies widely, thus patch and convolutional window lengths that work well on one dataset may overfit on another due to different relative sizes. While we were not able to perform a large experiment on different parameters due to time and resource constraints, doing so could be quite useful, especially if it revealed a reliable formula to predict performance settings. This could help greatly reduce convergence time when training a new model on unseen datasets.

5. Reflections

This project reinforced the value of simplicity and thoughtful architectural design in deep learning for time series forecasting. Despite our intuition that 1D convolutional layers might yield richer patch embeddings, our results showed only marginal improvements, and in some cases, the added complexity led to overfitting. Similarly, while we hypothesized that introducing randomized mask ratios in the self-supervised setting would improve robustness and generalization, we found that the original fixed-ratio approach was already quite effective, and randomization occasionally degraded performance. These outcomes taught us that additional modifications must be carefully validated, but we also learned that these seemingly small design choices such as patch segmentation strategies or masking schemes have the potential to improve model performance.

Looking forward, we see several promising directions for future work. One possibility is to explore more diverse patch embedding strategies, such as max pooling, EMA smoothing, or dropout-enhanced features, which was an area we saw explored in a different paper [2]. Another direction is to investigate adaptive masking schedules in self-supervised learning to help the model progressively learn harder reconstructions.

Also, we encountered challenges related to hyperparameter sensitivity and computational resource constraints, particularly when experimenting with more complex architectures. Thus, with additional time and computational resources, perhaps these directions could further enhance the model's ability to capture long-range temporal dependencies and generalize across diverse time series tasks.

6. References

[1] G. L. Asher, “Exploring Tokenization Techniques to Optimize Patch-Based Time-Series Transformers,” Computer Science Senior Theses, no. 47, Dartmouth College, 2024. [Online]. Available: https://digitalcommons.dartmouth.edu/cs_senior_theses/47

[2] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, “A Time Series is Worth 64 Words: Long-term Forecasting with Transformers,” in Proc. Int. Conf. Learn. Representations (ICLR), 2023.