Progress Report

Project ID: TS006

Enhancing Long-Sequence Time Series Forecasting by Integrating Efficient Attention Mechanisms

Submitted by:

210173T – GALLAGE D.

Department of Computer Science & Engineering
University of Moratuwa
Sri Lanka

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

1.1 Background

A fundamental problem in machine learning, time series forecasting has significant ramifications for fields like supply chain logistics, finance, energy management, and meteorology. Predicting future values based on historical observations is the goal, and the longer the prediction horizon, the more challenging this task becomes. While traditional statistical methods like ARIMA have long been staples, the advent of deep learning has introduced powerful new paradigms.

The self-attention mechanism of the Transformer architecture, which was initially created for natural language processing, has made it a particularly promising tool for sequence modeling tasks because it is very good at capturing long-range dependencies [1]. However, the direct application of Transformers to time series forecasting has been fraught with challenges. A primary obstacle is the self-attention mechanism's quadratic computational and memory complexity $\mathcal{O}(L^2)$ with respect to sequence length L), which renders it prohibitively expensive for the very long input sequences often required to model complex temporal patterns. This limitation has spurred a wave of research aimed at creating more efficient Transformer variants for time series analysis.

1.2 Motivation

The recently proposed Patch Time Series Transformer (PatchTST) marked a significant breakthrough by rethinking how time series data is presented to a Transformer [2]. By segmenting time series into "patches" and processing each channel (variable) independently, PatchTST achieved state-of-the-art results, even outperforming a simple linear model that had previously challenged the utility of complex Transformers [3].

PatchTST still uses a conventional "vanilla" Transformer encoder in spite of its architectural innovations. The model is clearly limited in its ability to handle extraordinarily lengthy historical contexts by its dependence on the original self-attention mechanism. Concurrently, models like Informer and FEDformer have specifically focused on optimizing the attention mechanism itself, achieving log-linear and linear complexity, respectively [4, 5].

This research is motivated by a compelling and logical next step to synthesize these parallel streams of innovation. By integrating the highly efficient attention mechanisms of Informer and FEDformer into the robust and effective architectural framework of PatchTST, we hypothesize that we can create a new class of hybrid models. These models would not only inherit the superior representational power of PatchTST but also overcome its primary computational limitations, resulting in a faster, more scalable, and potentially more accurate solution for long-term time series forecasting.

1.3 Research Objectives

The overarching goal of this project is to develop and rigorously evaluate a new generation of Transformer-based models for time series forecasting. The specific objectives are:

1. To develop two novel hybrid models:
   - PatchTST with Sparse: Integrating Informer's ProbSparse attention into the PatchTST framework.
   - PatchTST with Fourier: Integrating FEDformer's frequency enhanced structure into the PatchTST framework.
2. To conduct a comprehensive performance benchmark: The new models will be evaluated against the original PatchTST, Informer, and FEDformer on a suite of widely used public time series datasets.
3. To perform a detailed efficiency analysis: The computational time and peak memory usage of all models will be profiled to quantify the benefits of the integrated mechanisms.

2 LITERATURE REVIEW

The application of Transformers to time series forecasting is a rapidly evolving field. This review synthesizes the key developments that form the foundation of this research proposal, focusing on the progression from standard Transformers to the specialized architectures of PatchTST, Informer, and FEDformer.

2.1 The Transformer in Time Series

The Transformer's success is based on its self-attention mechanism. When creating a representation for a particular element, it enables the model to consider the relative importance of various elements in a sequence. For a sequence of input embeddings X, it computes the Query (Q), Key (K), and Value (V) matrices and calculates the output as:

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

This formulation allows for rich, context-aware representations. However, the matrix multiplication QK^T results in an $L \times L$ attention map, leading to the problematic $\mathcal{O}(L^2)$ complexity.

Early attempts to apply Transformers to time series often struggled with this bottleneck, limiting the look-back window and thus the model's ability to capture long-term patterns. Furthermore, these models often treated each time step as an independent token, failing to capture the local, continuous nature of time series data.

2.2 PatchTST: A New Paradigm for Tokenization

The PatchTST paper fundamentally challenged the standard tokenization approach by introducing a simple yet powerful architectural redesign [2]. As shown in Fig. 1, its core contributions were twofold, addressing both how the model sees the data (patching) and how it processes multiple variables (channel-independence).

2.2.1 Patching

Instead of treating each time step as an individual token, a method that fails to capture local context, PatchTST segments the input time series into overlapping subseries-level *patches*. A time series of length L is converted into a sequence of approximately $N \approx L/S$ patches (where P is the patch length and S is the stride). Each patch, containing P consecutive time steps, is then linearly projected into an embedding vector, which serves as a single input token to the Transformer.

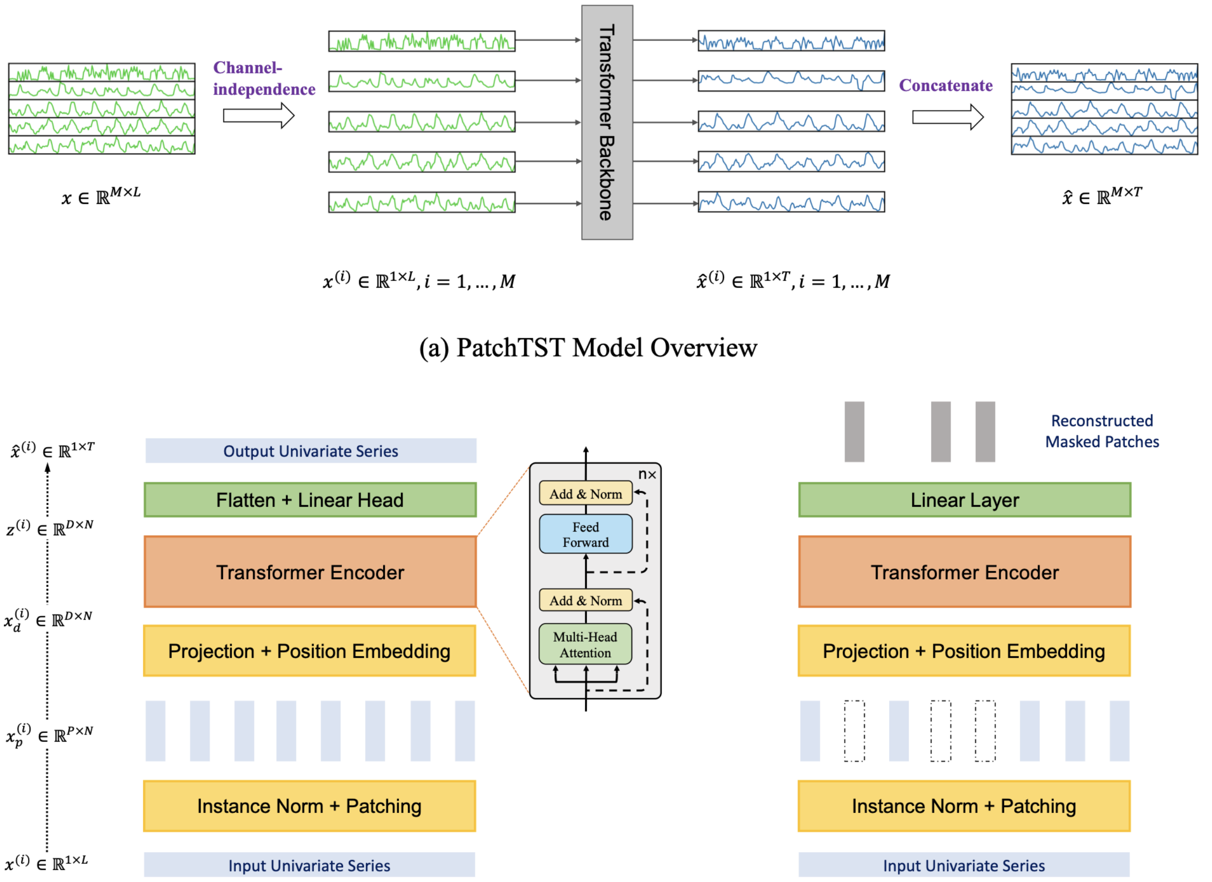
This approach has several profound benefits:

- **Drastic Complexity Reduction**: By reducing the input sequence length from L to N, the attention mechanism's complexity is reduced from $\mathcal{O}(L^2)$ to $\mathcal{O}(N^2)$, directly mitigating the primary computational bottleneck.
- **Retention of Local Semantics**: A single time point has little semantic meaning on its own. A patch, however, can encapsulate meaningful local patterns like trends, seasonality, or volatility, providing the Transformer with richer, more informative tokens.
- **Alignment with Vision Transformers**: This strategy mirrors the successful patching approach used in Vision Transformers [6], creating a more unified understanding of how Transformers can be applied to different data modalities.

2.2.2 Channel-Independence

For multivariate time series, most prior models employed a “channel-mixing” approach, where the values from all M variables at a single time step are combined into one large vector token. PatchTST takes the opposite approach. It processes each of the M channels independently, feeding each univariate time series through an identical, shared-weight Transformer backbone. This seemingly counter-intuitive design proved highly effective, as it reduces overfitting by simplifying the learning task for each backbone and allows the model to learn distinct temporal patterns for each variable while still sharing knowledge through the common weights.

While PatchTST set a new state-of-the-art, its innovation was purely architectural. The core engine, the self-attention block, remained the standard, computationally expensive version.


Channel-independence
Transformer Backbone
Concatenate
$x \in \mathbb{R}^{M \times L}$
$x^{(i)} \in \mathbb{R}^{1 \times L}, i = 1, \dots, M$
$\hat{x}^{(i)} \in \mathbb{R}^{1 \times T}, i = 1, \dots, M$
$\hat{x} \in \mathbb{R}^{M \times T}$
$\hat{x}^{(i)} \in \mathbb{R}^{1 \times T}$
$z^{(i)} \in \mathbb{R}^{D \times N}$
$x_d^{(i)} \in \mathbb{R}^{D \times N}$
$x_p^{(i)} \in \mathbb{R}^{P \times N}$
$x^{(i)} \in \mathbb{R}^{1 \times L}$
Output Univariate Series
Flatten + Linear Head
Transformer Encoder
Projection + Position Embedding
Instance Norm + Patching
Input Univariate Series
n×
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Reconstructed Masked Patches
Linear Layer
Transformer Encoder
Projection + Position Embedding
Instance Norm + Patching
Input Univariate Series

(a) PatchTST Model Overview

(b) Transformer Backbone (Supervised)

(c) Transformer Backbone (Self-supervised)

Fig. 1: PatchTST architecture.

The empirical success of this architecture was remarkable. As shown in the Table 1, which summarizes key results from the original paper, PatchTST significantly outperformed previous state-of-the-art models on a wide range of benchmarks.

Model	Weather (MSE)	Traffic (MSE)	Electricity (MSE)	Overall MSE Reduction
FEDformer	0.389	0.621	0.344	–
Autoformer	0.415	0.639	0.342	–
PatchTST/64	**0.314**	**0.432**	**0.290**	**$\sim$ 21.0%**

Table 1: A summary of comparative results (MSE for $T = 720$) showing PatchTST's significant performance improvement over prior SOTA Transformer-based models. The overall reduction is relative to the best-performing baseline.

2.3 Informer: Efficiency-Driven Transformer Design

The Informer model [4] was designed from the ground up to tackle the efficiency challenges of applying Transformers to long sequence forecasting. Its contributions are both algorithmic and architectural.

ProbSparse Attention is the main algorithmic innovation. This mechanism is based on the empirical finding that the majority of time series' attention scores follow a "long-tail" distribution, meaning that only a small percentage of query-key pairs significantly influence the outcome. To exploit this, ProbSparse attention does not compute the entire $L \times L$ attention matrix. It approximates the importance of each query using its Kullback-Leibler divergence from a uniform distribution. It then reduces the complexity from $\mathcal{O}(L^2)$ to a much more manageable $\mathcal{O}(L \log L)$ by choosing only the top-u most important keys for each query (where $u = c \cdot \ln L$ for a small constant c).

Beyond the attention mechanism, the Informer encoder introduces a *self-attention distilling* operation to further manage sequence length. After each attention layer, a 1D convolutional layer and a max-pooling layer are applied to the feature map, effectively halving the sequence length. This "distilling" process progressively shortens the sequence as it moves through the encoder stack, reducing memory usage and focusing subsequent layers on the most salient temporal features.

This combination of an efficient attention algorithm and a length-reducing encoder architecture allows Informer to handle extremely long input sequences while maintaining high performance.

2.4 FEDformer: A Frequency-Domain Approach

FEDformer [5] proposed an even more radical departure from standard attention. It leverages principles from signal processing, positing that the most critical long-term dependencies in a time series can be more compactly represented by its dominant frequency components. Its methodology involves:

1. Fourier Transform: The model applies a Fast Fourier Transform (FFT) to the sequence, converting it from the time domain to the frequency domain.

2. Frequency Mode Selection: It then performs a random selection of a fixed number of frequency modes or learns to select the most important ones. This operation is independent of the sequence length L.

3. Inverse Fourier Transform: The processed frequency components are transformed back to the time domain using an Inverse FFT (IFFT).

By operating on a fixed number of frequencies, FEDformer achieves a remarkable linear complexity $\mathcal{O}(L)$. This makes it exceptionally efficient for very long sequences and particularly adept at modeling data with strong periodicities.

2.5 Research Gap

The literature reveals a clear and compelling research gap. On one hand, PatchTST provides a superior architectural framework for representing time series data, but it is constrained by an inefficient, standard attention mechanism. On the other hand, Informer and FEDformer introduce highly efficient, algorithmically advanced attention mechanisms but may lack the representational benefits of PatchTST's patching and channel-independent design.

The opportunity lies in synthesizing these two streams of innovation. No existing work has combined the architectural advantages of PatchTST with the algorithmic efficiency of specialized attention mechanisms. This project aims to bridge that gap. The Table 2 summarizes the complexity of the core models, highlighting the potential for creating a hybrid model that is both structurally sound and computationally efficient.

Model	**Training Time**	**Training Memory**
Transformer	$\mathcal{O}(L^2)$	$\mathcal{O}(L^2)$
Informer	$\mathcal{O}(L \log L)$	$\mathcal{O}(L \log L)$
FEDformer	$\mathcal{O}(L)$	$\mathcal{O}(L)$

Table 2: Complexity analysis of different forecasting models.

3 METHODOLOGY

This research will employ a quantitative, experimental methodology to develop and validate the proposed hybrid models. The process is designed to be rigorous, reproducible, and directly comparable to existing state-of-the-art benchmarks.

3.1 Model Development

The core technical contribution will be the implementation of two novel hybrid models within a PyTorch framework. This will involve a modular replacement of the attention block in the publicly available PatchTST codebase.

1. PatchTST with Sparse: This model will be created by replacing the torch.nn.MultiHeadAttention module within the PatchTST EncoderLayer with the ProbSparseAttention module from the official Informer implementation. Key hyperparameters to be tuned will include the sampling factor c.

2. PatchTST with Fourier: This model will be developed by substituting the entire self-attention and feed-forward block in the PatchTST EncoderLayer with the frequency-domain FourierBlock from the FEDformer architecture. Key hyperparameters will include the number of selected frequency modes.

3.2 Experimental Setup

The experimental setup will be carefully aligned with the standards set in the original PatchTST paper to guarantee a thorough and equitable comparison. This entails applying the most recent baseline models, evaluation metrics, and publicly available benchmark datasets. The results produced by this study will be directly comparable to published literature by upholding this consistency, offering a transparent and unambiguous evaluation of the performance of the suggested hybrid models.

- **Datasets**: To ensure a fair and comprehensive evaluation, the models will be benchmarked on the same suite of public datasets used in the PatchTST paper: Weather[1], Traffic[2], Electricity[3], ILI[4] and ETT[5]. These datasets vary widely in their number of variables, length, and temporal characteristics, providing a robust testbed.
- **Evaluation Metrics**: Forecasting accuracy will be measured using Mean Squared Error (MSE) and Mean Absolute Error (MAE). These are standard metrics in the forecasting literature and will allow for direct comparison with published results.
- **Baselines**: The performance of PatchTST with Sparse and PatchTST with Fourier will be benchmarked against the three original models: PatchTST, Informer, and FEDformer. All baseline models will be run using their official codebases and recommended hyperparameter settings to establish a strong and fair point of comparison.

3.3 Evaluation Protocol

The evaluation of the newly developed models will be conducted through a comprehensive protocol, designed to assess their performance from multiple critical perspectives. This protocol will measure their raw forecasting accuracy. This multi-faceted approach will provide a holistic understanding of the models' strengths and weaknesses.

- **Forecasting Accuracy Benchmark**: All five models (two new, three baselines) will be trained and evaluated on all eight datasets for the standard long-term forecasting horizons ($T \in \{96, 192, 336, 720\}$). The resulting MSE and MAE scores will be tabulated to determine the relative performance.
- **Computational Efficiency Analysis**: During the training process of the main benchmark, the wall-clock training time per epoch and the peak GPU memory utilization will be systematically logged for each model on the largest datasets (e.g., *Traffic*, *Electricity*). This will provide quantitative evidence of the efficiency gains.

[1] `https://www.bgc-jena.mpg.de/wetter/`

[2] `https://pems.dot.ca.gov/`

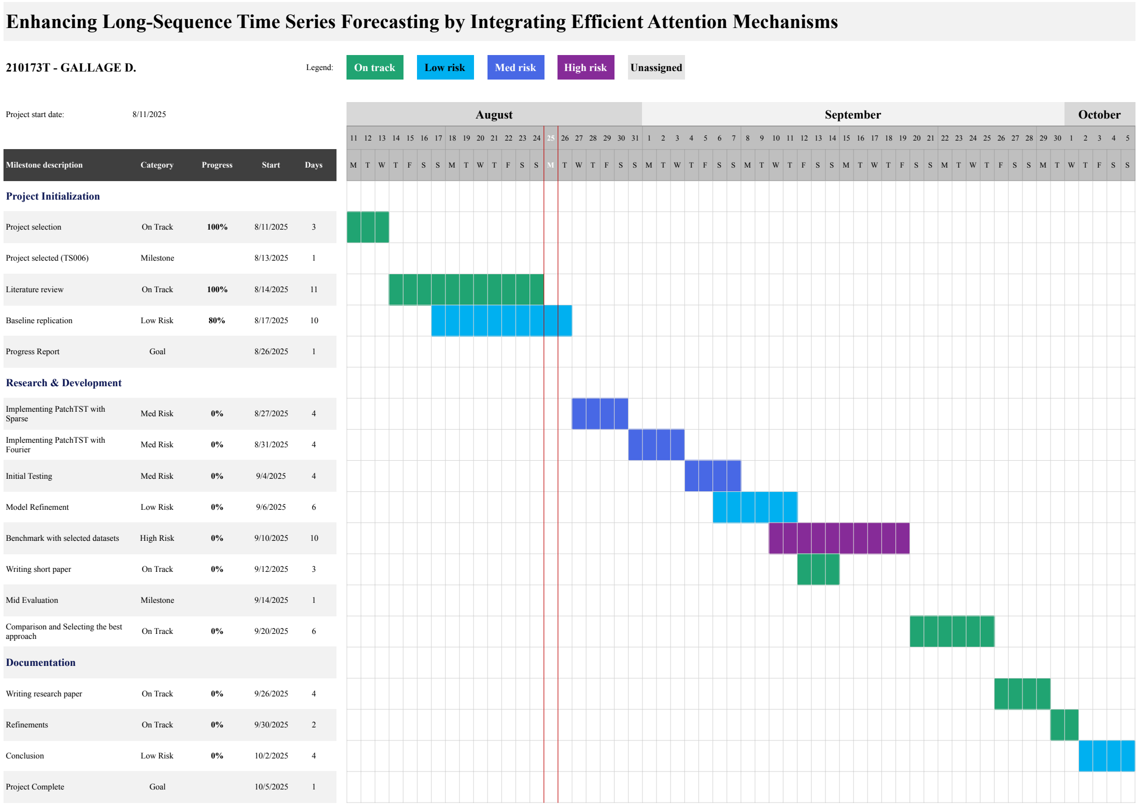
[3] `https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014`

[4] `https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html`

[5] `https://github.com/zhouhaoyi/ETDataset`

4 RESEARCH TIMELINE

The estimated timeline and significant milestones for this research project are shown in the Gantt chart in Fig. 2. It guarantees a methodical and timely progression through all stages of the study by offering a thorough breakdown of the tasks and their anticipated durations. Please be aware that this schedule is subject to change as the needs and nature of the research process change over time.



Enhancing Long-Sequence Time Series Forecasting by Integrating Efficient Attention Mechanisms

210173T - GALLAGE D.

Legend: 🟩 On track 🟦 Low risk 🟦 Med risk 🟪 High risk ⬜ Unassigned

Project start date: 8/11/2025

Milestone description	Category	Progress	Start	Days
Project Initialization				
Project selection	On Track	**100%**	8/11/2025	3
Project selected (TS006)	Milestone		8/13/2025	1
Literature review	On Track	**100%**	8/14/2025	11
Baseline replication	Low Risk	**80%**	8/17/2025	10
Progress Report	Goal		8/26/2025	1
Research & Development				
Implementing PatchTST with Sparse	Med Risk	**0%**	8/27/2025	4
Implementing PatchTST with Fourier	Med Risk	**0%**	8/31/2025	4
Initial Testing	Med Risk	**0%**	9/4/2025	4
Model Refinement	Low Risk	**0%**	9/6/2025	6
Benchmark with selected datasets	High Risk	**0%**	9/10/2025	10
Writing short paper	On Track	**0%**	9/12/2025	3
Mid Evaluation	Milestone		9/14/2025	1
Comparison and Selecting the best approach	On Track	**0%**	9/20/2025	6
Documentation				
Writing research paper	On Track	**0%**	9/26/2025	4
Refinements	On Track	**0%**	9/30/2025	2
Conclusion	Low Risk	**0%**	10/2/2025	4
Project Complete	Goal		10/5/2025	1

Fig. 2: Proposed Research Schedule

5 CONCLUSION

By standing on the shoulders of giants-integrating the proven efficiency of Informer and FEDformer into the state-of-the-art PatchTST framework-this project is strategically positioned to make a tangible impact. The proposed methodology is systematic and rigorous, and the timeline is realistic. This research promises not only to produce superior forecasting models but also to contribute valuable knowledge to the ongoing effort to build more powerful and scalable tools for understanding and predicting complex temporal data.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Kalagnanam, "A time series is worth 64 words: Long-term forecasting with transformers," in *International Conference on Learning Representations*, 2023.

[3] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" *arXiv preprint arXiv:2205.13504*, 2022, https://doi.org/10.48550/arXiv.2205.13504.

[4] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence timeseries forecasting," in *The ThirtyFifth AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, no. 12. AAAI Press, 2021, pp. 11 106–11 115.

[5] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proceedings of the 39th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds., vol. 162. Baltimore, Maryland: PMLR, July 2022, pp. 27 268–27 286.

[6] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3–7, 2021.* OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=YicbFdNTTy