Transformer-based Architectures for Time Series Foundation Models

Literature Review

MARZOUG AYOUB

June 2025

Context & Motivation

Foundation Models

- Self-supervised learning on massive, diverse data.
- Strong zero-shot / few-shot generalization.
- Success in NLP & Vision $\rightarrow$ Can we transpose this to time series ?

Why Time Series ?

- Financial, industrial, and environmental applications.
- Multiscale structure, Irregular sampling, Noisy signals.
- Temporal dynamics vary by domain.

Research Challenge

Internship Objective

- Identify an optimal **Transformer-based architecture**.
- Build a **foundation model for financial time series**.
- Capture key *stylized facts* : mean-reversion, volatility clustering, etc.

Core Challenges

- **Tokenization** of irregular, multiscale signals.
- Encoding statistical models : OU, Heston, GARCH.
- Balancing generalization with domain specificity.

Goal : Ground Transformer design in financial model priors.

What Are Time Series Foundation Models ?

Foundation Models are large deep models :

- Pretrained on large, diverse time series corpora.
- Capture general-purpose representations across modalities.
- Adaptable via fine-tuning or few-shot prompts.

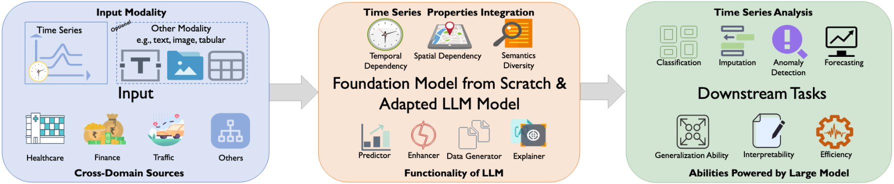
Time Series Foundation Models are :

- Pretrained on diverse time series (e.g., finance, healthcare, traffic).
- Leverage temporal, spatial, and semantic properties.
- Transfer across domains with minimal adaptation.

Key Advantage : Train once, generalize everywhere

Enables zero-shot and few-shot forecasting across domains.

TSFM Scope



TSFMs leverage diverse inputs and temporal structures to serve a variety of downstream tasks.

Desirable Properties of TSFM

Generalization
Across datasets, tasks

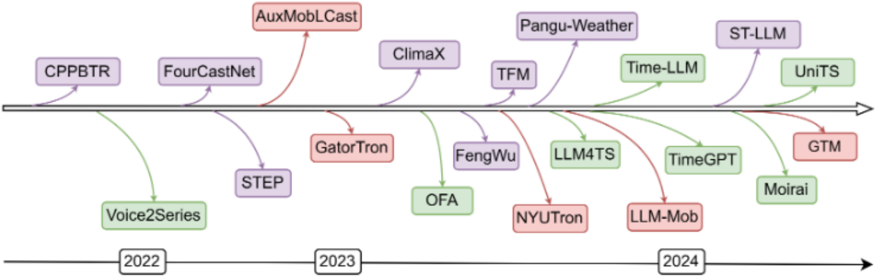
Scalability
Large model &
data regimes

Temporal Fidelity
Preserve fre-
quency/dynamics

Light Adaptation
Few-shot or
prompt-tuned

Time series FMs must balance **generalization**, **fidelity**, *and* **lightweight adaptation**.

Roadmap of Representative TSFMs



Overview of Time Series Foundation Models

Model	Arch. Type	Tokenization	Probabilistic ?	Pretraining
Lag-Llama	Decoder-only	Lag features	✓	27 real datasets
TimesFM	Decoder-only	Patching	× (point)	Synthetic + real
Chronos	Decoder-only	Quantization	✓	Quantized tokens
ForecastPFN	Encoder-only	Synthetic signals	× (point)	100% synthetic

A spectrum of design choices : **token type**, **architecture**, **forecast type**, *and* **training corpus**.

Transformer-based Architectures

Why Transformers for Foundation Models ?

- Leverage attention :

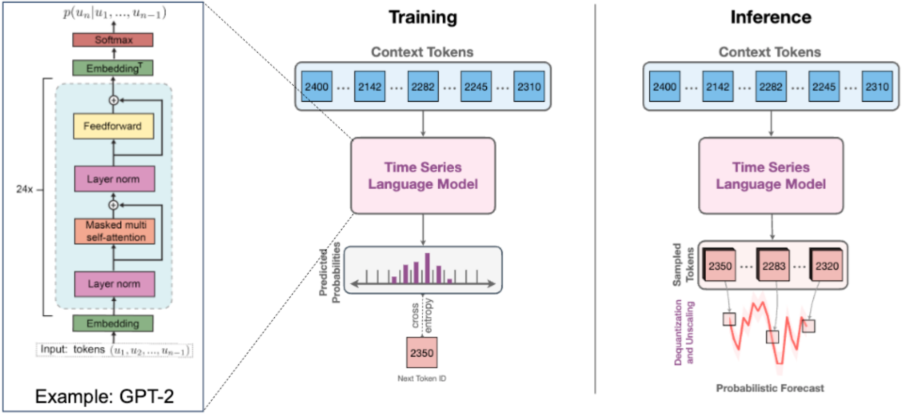
$$\text{Attention}(Q, K, V) = \text{Softmax}(QK^\top/\sqrt{d_k})V$$

- Capture long-range dependencies without recurrence.
- Scalable and parallelizable — suitable for large data and parameter regimes.

Architectural Choices in TSFMs :

- **Encoder-only** : Effective for small datasets.
- **Decoder-only** : Suited for autoregressive forecasting and generative TSFMs.
- **Encoder-decoder** : For more flexible sequence-to-sequence mappings.

Transfomer-based Models



TSFM architectures are increasingly specialized—blending core Transformer advantages with domain-specific temporal and spatial innovations.

Pipeline Perspective

Pre-training Strategies :

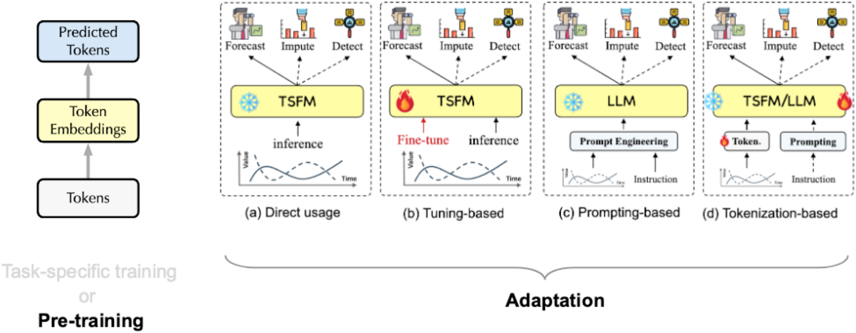
- **Fully-supervised :** Train on large labeled TS datasets.
- **Self-supervised :**
  - *Generative :* Masked reconstruction or density modeling.
  - *Contrastive :* Augmentations + positive/negative pair learning.
  - *Hybrid :* Combines both to improve generalization.
- Enables learning from large-scale unlabeled data, boosting transferability.

Adaptation Mechanisms :

- **Direct usage (a) :** Zero-shot deployment.
- **Fine-tuning (b) :** Full model or selective components.
- **Prompting (c) :** Static or learned prompts for task conditioning.
- **Tokenization (d) :** Patch-based, normalized, decomposed TS embeddings.

TSFM Adaptation Pipeline

- **Pipeline**



TSFM adaptation pipeline : from direct inference to prompting and specialized tokenization.

Tokenization : Role and Challenges

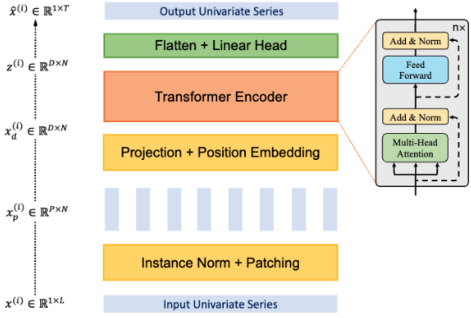
Why Tokenization is Crucial

- No fixed vocabulary : continuous, irregular, noisy inputs.
- Tokenization = encoding inductive bias (lags, trends, noise).
- Defines what patterns the Transformer can learn and transfer.

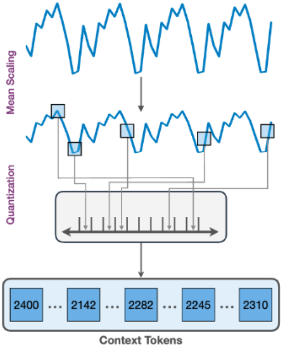
What Good Tokenization Enables

- Structured TS $\rightarrow$ Tokens $\rightarrow$ Generalization.
- Unlocks cross-domain zero-/few-shot transfer.
- Aligns model behavior with financial/temporal priors.

Tokenization Strategies



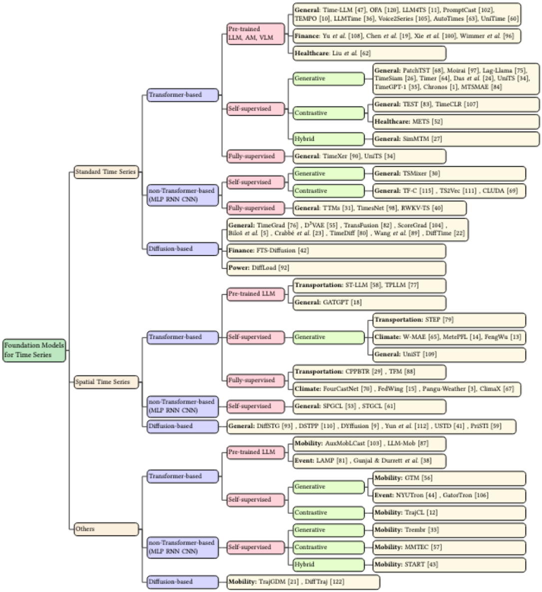
Patchify



Quantization

Tokenization is not just preprocessing — it defines how the model learns and transfers.

Comprehensive Taxonomy of TSFMs



Compatibility with Financial Time Series

Why Financial Data is Challenging :

- **Stylized facts :** Heavy tails, mean-reversion, regime shifts.
- **Low signal-to-noise ratio** and market microstructure effects.

What TSFMs Must Capture :

- Inductive priors compatible with :
  - *Ornstein-Uhlenbeck (OU) :* Mean reversion in log prices.
  - *GARCH-family models :* Conditional heteroskedasticity.
  - *Heston model :* Stochastic volatility with latent dynamics.
- Ability to disentangle signal from noise across horizons.
- Learn representations that generalize across assets, markets, and regimes.

Goal

Bridge deep learning architectures with domain-aware financial stochastic modeling.

Lag-Llama

Lag-Llama : Backbone Architecture

Architecture

Lag-Llama is a **decoder-only Transformer** trained in an autoregressive fashion on lag-structured tabular tokens.

Input Structure :

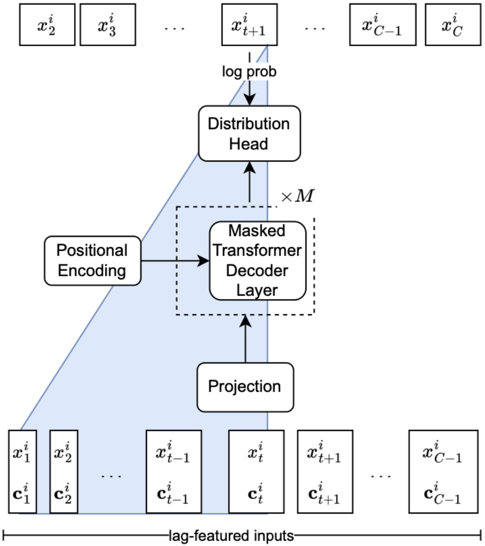
- For each step t, a token k_t is created by concatenating :
  - Lagged target values $\{y_{t-l_1}, \ldots, y_{t-l_n}\}$.
  - Time-varying covariates (e.g., day-of-week).
  - Static metadata (e.g., item ID).
- Tokens are processed autoregressively :

$$\hat{y}_{t+h} = f_\theta(k_t, k_{t-1}, \ldots)$$

Design Note

No sequence embedding or patching — relies solely on lag features.

Lag-Llama Full Architecture



Lag-Llama : Positional Encoding

Strategy

Uses **learned absolute position embeddings** for each lag index and optional forecast horizon embedding.

Position Encoding :

$$e_{\text{pos}}(l_j) + x_{t-l_j}, \quad j = 1, \ldots, n$$

$$e_{\text{horizon}}(h), \quad \text{for step } t + h$$

- Embeddings injected into tabular token fields.
- No use of sinusoidal or relative encodings.

Critique

Lacks frequency priors and does not generalize to irregularly spaced data.

Core Idea

Lag-Llama tokenizes time series by constructing **lag-based feature vectors** for each time step, rather than patches or quantized bins.

Definition : Lag Tokenization

Given a sorted list of lag indices $L = l_1, l_2, \ldots, l_n$, the token k_t for time t is :

$$k_t[j] = x_{t-l_j}, \quad \text{for } j = 1, \ldots, n$$

Each $k_t \in \mathbb{R}^{|L|}$ is augmented with :

- **Date-time covariates :** second, minute, hour, day, etc.
- **Summary stats :** mean (μ) and scale (σ) from robust scaling (IQR).

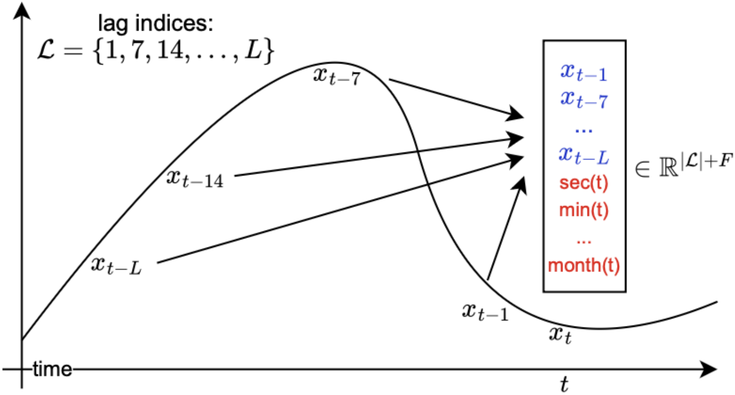


Illustration of Lag-Llama token construction : lags + datetime + summary features.

Lag-Llama : Input Modality Handling

Supported Modalities

Lag-Llama supports both **univariate** and **multivariate** targets. Each token incorporates :

- Static covariates (e.g., item ID, product type) via learned embeddings.
- Time-varying covariates (e.g., calendar effects, weather) into the token stream.

- No support for cross-series attention — each time series is modeled independently.
- This limits scalability to portfolio-level forecasting or correlated asset dynamics.

Implication for Finance

Lacks ability to capture latent factor models, sectoral dependencies, or shared volatility clusters.

Lag-Llama : Temporal Granularity

Design Constraints

- Fixed-length context window (e.g., 36 lags).
- Forecast horizon h is a fixed offset — cannot vary dynamically.

Limitations

- Cannot handle irregular sampling or variable-step prediction.
- No built-in support for temporal hierarchy or scale-adaptive inference.

Lag-Llama : Pretraining Methodology

Training Objective

Lag-Llama is trained using an **autoregressive forecasting loss**, predicting future values from past lagged inputs :

$$\mathcal{L}_{\text{forecast}} = \frac{1}{H} \sum_{h=1}^{H} (y_{t+h} - \hat{y}_{t+h})^2$$

Where H is the forecast horizon and $\hat{y}_{t+h}$ is the model output.

- Deterministic output — no probabilistic head or uncertainty modeling.
- No self-supervised pretraining : no masking, contrastive, or generative objectives.

Implications

Lacks calibrated risk estimation — suboptimal for finance where *predictive distribution* is as important as point forecasts.

Lag-Llama : Dataset Corpus

Pretraining Dataset

Amazon Forecast TSF Dataset :

- >80,000 time series from retail, logistics, and energy sectors.
- High variability across product categories, geographies, and seasonality.

Lag-Llama is evaluated on :

- M4 (public forecasting competition)
- Electricity, Traffic, Influenza-like illness (ILI)

Critical Limitation

No financial datasets used — generalization to high-volatility or regime-shifting financial data remains untested.

TimesFM

TimesFM : Backbone Architecture

Design Overview

TimesFM uses a **decoder-only Transformer** with causal self-attention, designed for efficient long-horizon forecasting.

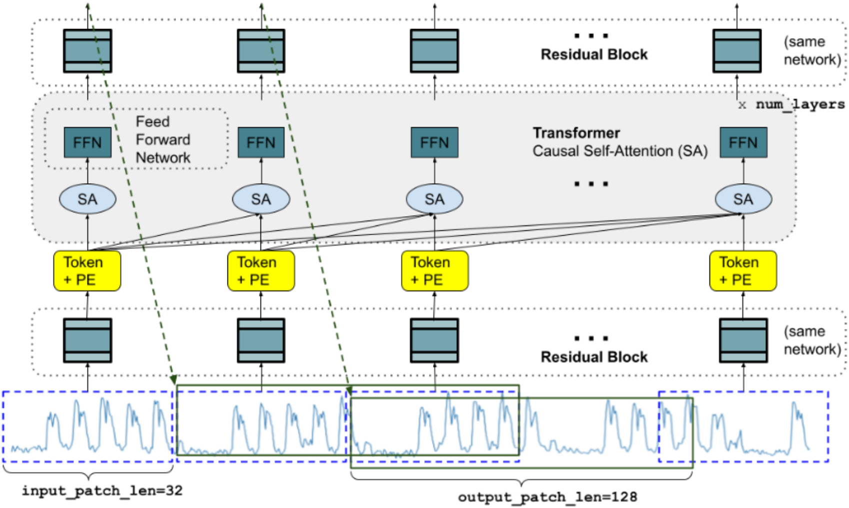
Key Characteristics :

- Patch-based token inputs reduce sequence length.
- No encoder : all inputs (context patches) and outputs (forecast patches) flow through the same Transformer block.
- Output tokens generate multiple time steps via a residual head.

Critique

Compared to encoder-decoder or latent diffusion models, this architecture is more efficient but potentially less expressive for structured forecasting tasks (e.g., hierarchical financial signals).

Architecture Summary



Decoder-only architecture with patch tokenization, causal attention, and MLP heads

TimesFM : Positional Encoding

Strategy

TimesFM uses **learned absolute positional embeddings** PE_j added to each patch token.

Equation :

$$t_j = R(\tilde{y}_j) + PE_j, \quad \text{for } j = 1, \ldots, N$$

Limitations :

- No frequency-based or relative encoding.
- Implicitly assumes fixed patch length and regular sampling.

Implication

This positional encoding suffices for general-purpose data, but lacks inductive bias for financial periodicity or irregular timestamps.

TimesFM : Tokenization Strategy (Patching)

Core Idea

TimesFM transforms time series into **non-overlapping fixed-length patches**. Each patch is a token in the Transformer input.

Token Construction : Given a time series $y_1, \ldots, y_L$ and patch size p :

$$\tilde{y}_j = y_{p(j-1)+1:pj} \in \mathbb{R}^p, \quad \text{for } j = 1, \ldots, \lfloor L/p \rfloor$$

Each patch is :

- Processed by Residual MLP $R(\cdot)$
- Embedded as $t_j = R(\tilde{y}_j) + PE_j$

Motivation Behind Patching

- **Efficiency :** Reduces sequence length for Transformer $\Rightarrow$ faster training.
- **Generalization :** Compatible with variable-length contexts and prediction horizons.
- **Modularity :** Patches serve as atomic temporal units, facilitating domain-agnostic modeling.

Training-Time Features

- Random masking of full and partial patches.
- Ensures robustness to different context lengths.

Prediction Tokens : Long Output Patches

Problem

Auto-regressive forecasting is inefficient for long horizons.

Solution : Predict longer output patches.

Let h be the output patch length :

$$\hat{y}_{pj+1:pj+h} = \mathrm{OutputResidualBlock}(o_j)$$

- Each output token predicts h future time steps.
- Requires fewer decoding steps $\Rightarrow$ speed $+$ better accuracy.

Trade-off

Cannot use very large h for short input sequences.

TimesFM : Input Modality Handling

Supported Modalities

- Trained on **univariate** target series.
- Static metadata (e.g., series type) embedded via learned vectors.
- No explicit multivariate support — each univariate series modeled independently.

Limitation

No cross-variable or cross-series attention — ill-suited for multivariate financial datasets with correlation structures.

TimesFM : Temporal Granularity

Resolution Strategy

- Supports arbitrary-length context and prediction patches.
- Patch size and output length h are tunable at inference.

Prediction Equation :

$$\hat{y}_{pj+1:pj+h} = \text{ResidualHead}(o_j)$$

Flexibility

Adaptable to different forecasting horizons and granularities, though assumes regular sampling within patches.

TimesFM : Pretraining Methodology

Objective

TimesFM is trained via a denoising autoencoding objective with masking :

$$\mathcal{L}_{\text{patch}} = \frac{1}{M} \sum_{j \in \mathcal{M}} \|\hat{y}_j - y_j\|^2$$

Where $\mathcal{M}$ indexes masked patches, and $\hat{y}_j$ is the reconstructed output.

Key Features :

- Random masking of full/partial patches.
- Trained on diverse time series without task-specific labels.

Benefit

Supports general-purpose adaptation — captures temporal structure via self-supervised learning.

TimesFM : Dataset Corpus

Pretraining Dataset

Dataset Mix :

- Over 100 diverse time series datasets.
- Includes finance-adjacent sources (e.g., commodities, energy, macro indicators).
- Over 1 billion tokens from univariate series.

Domain Coverage :

- Retail, climate, finance, web traffic, medical.

Strength

More diverse than prior models (e.g., Lag-Llama), enabling broad generalization.

Chronos

Chronos : Backbone Architecture

Core Design

Chronos leverages pretrained large language models (LLMs), specifically :

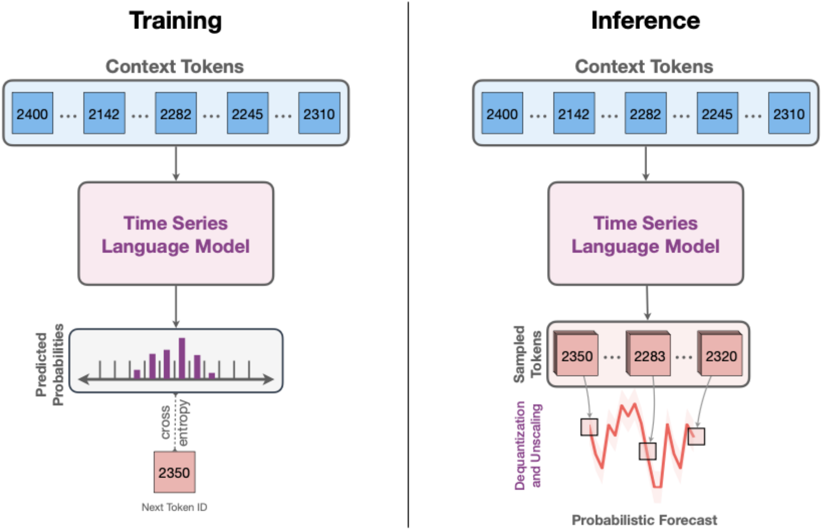
- **T5 (encoder-decoder)** and **GPT-2 (decoder-only)** variants.
- Used **without modification**, other than resizing the input/output vocabulary to match quantized token set.

This approach enables seamless reuse of mature, scalable, and highly optimized LLM infrastructures for time series forecasting.

Trade-off

While offering architectural simplicity, this discards potential benefits from domain-specific inductive biases, such as seasonality-aware layers or frequency-domain encoders.

Chronos Training & Inference



Chronos : Positional Encoding Strategy

Core Choice

Chronos **does not use any explicit positional encoding**.

Implication :

- Forecasting treated as pure autoregressive sequence modeling.
- No calendar time, frequency bands, or relative timing cues.

Critique

This choice maximizes architectural reuse but limits expressiveness, especially in finance where position-dependent dynamics (e.g., calendar effects, volatility clusters) are critical.

Chronos : Tokenization Strategy

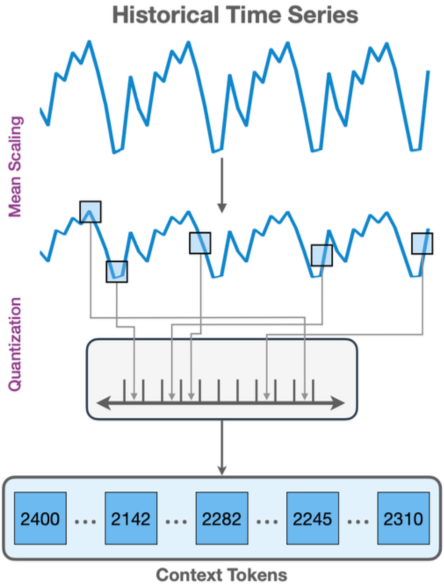
Core Idea

Chronos converts real-valued time series into sequences of **discrete tokens** via **mean scaling** and **uniform quantization**. This enables use of off-the-shelf language models.

Token Construction

- Given $x_1, \ldots, x_{C+H}$: context (1 to C), forecast (next H).
- **Mean Scaling :** $\tilde{x}i = \frac{x_i}{\frac{1}{C} \sum j = 1^C |x_j|}$
- **Quantization :** Uniform binning over $[-15, +15]$ into B bins (e.g., $B = 4096$)
- **Token :** $z_i = q(\tilde{x}_i) \in 1, ..., B$

Chronos Tokenization



Why Discrete Tokenization ?

Motivation :

- Language models require finite vocabulary.
- Continuous values $\rightarrow$ categorical tokens.
- Enables use of **cross-entropy loss** over token predictions.

Quantization Function :

$$q(x) = \begin{cases} 1 & \text{if } x < b_1 \ 2 \\ \text{if } b_1 \le x < b_2 \ \vdots \ B & \text{if } x \ge b_{B-1} \end{cases}, \quad d(j) = c_j$$

No Positional Features :

Unlike other models, Chronos **ignores time/frequency features** $\Rightarrow$ sequence-only input.

Chronos : Input Modality Handling

Supported Modes

Chronos operates on **univariate time series**, tokenized into 1D sequences of discrete values.

- No support for multivariate inputs or time-varying covariates.
- No special handling for static features.

Limitation

Cannot model interactions across assets, sectors, or covariates — a serious drawback for financial modeling tasks such as portfolio-level forecasting or macroeconomic inference.

Chronos : Temporal Granularity & Flexibility

Design Choices

- Forecasting horizon and context length are flexible.
- Input/output remain unstructured token streams.
- No frequency-awareness or hierarchical time handling.

Critique

Chronos is resolution-agnostic but lacks explicit tools for adapting across scales — limiting use in mixed-frequency macro-financial settings.

Chronos : Pretraining Methodology

Objective

Chronos uses standard **autoregressive cross-entropy loss** on quantized token sequences :

$$\ell(\theta) = -\sum_{h=1}^{H+1} \sum_{i=1}^{|V_{ts}|} \mathbf{1}(z_{C+h+1} = i) \log p_\theta(z_{C+h+1} = i | z_{1:C+h})$$

Training Details :

- Trained on millions of series (tokens from real-valued TS quantized to $|V_{ts}|$).
- Uses T5 and GPT-2 pretrained weights, finetuned on time series tokens.

Strength

High-quality probabilistic generation via sampling — enabling multimodal uncertainty-aware forecasting.

Chronos : Dataset Corpus

Pretraining Data

Chronos is trained on :

- **Amazon Forecast TSF data** (>80K series).
- **Proprietary time series corpora** including demand, traffic, climate.

Evaluation Benchmarks :

- Electricity, Traffic, ILI (public datasets).
- Forecast Horizon : 24–96 steps.

Limitation

No public financial datasets — limits inference on real-world financial market volatility or asset correlations.

ToTo

ToTo : Backbone Architecture

Core Structure

ToTo uses a **decoder-only Transformer** architecture, specifically optimized for time series observability data :

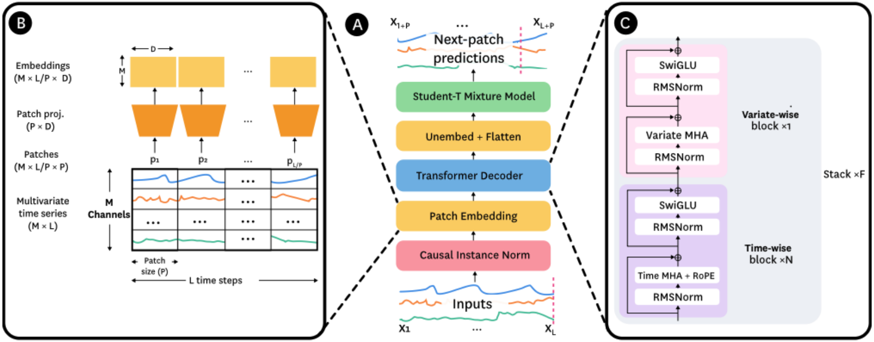
- Processes input patches autoregressively.
- Employs causal attention across **patch-embedded multivariate sequences**.

- No reuse of pretrained LLM weights.
- Specialized Student-T Mixture Model (SMM) prediction head for probabilistic output.

Implication

Unlike Chronos or Lag-Llama, ToTo incorporates **domain-specific architectural innovations** like factorized attention and causal scaling, purpose-built for multivariate observability forecasting.

Architecture Overview



ToTo : Positional Encoding

Method

ToTo applies **Rotary Positional Embeddings (RoPE)** to its patch-level latent tokens :

$$\theta_{2i} = \cos(\omega_i t), \quad \theta_{2i+1} = \sin(\omega_i t), \quad \omega_i = 10000^{-2i/d}$$

- Applied to input patch embeddings before attention.
- No learned or absolute position embeddings.
- Fixed-frequency sinusoidal structure supports long-range attention.

Finance Relevance

Improves extrapolation over long sequences, though lacks calendar-specific or seasonal priors.

ToTo : Causal Scaling for Autoregressive Stability

Problem

Standard LayerNorm is **non-causal** — it uses future values for normalization, which leads to information leakage in autoregressive settings.

Solution : Causal Scaling Layer

Replaces LayerNorm with online exponential moving average statistics :

$$\mu_t = \alpha\mu_{t-1} + (1-\alpha)x_t$$

$$\sigma_t^2 = \alpha\sigma_{t-1}^2 + (1-\alpha)(x_t - \mu_t)^2$$

$$\hat{x}_t = \frac{x_t - \mu_t}{\sqrt{\sigma_t^2 + \epsilon}}$$

- Fully causal, autoregressive-safe.
- Applied before each attention and feedforward block.

ToTo : Patch-Based Representation

Core Design

ToTo represents time series as patches instead of individual timesteps :

- Non-overlapping patches of size $P = 64$.
- Linear projection into latent space of dimension $D = 768$.
- Context length $L = 4096$ points mapped to L/P tokens.

Advantages

- More efficient than token-per-step models.
- Improves scalability for long sequences.

ToTo : Factorized Attention Mechanism

Motivation

Standard attention over flattened $[T, D]$ sequences is inefficient for high-dimensional multivariate time series.

- ToTo applies **separate self-attention** along the **temporal** and **feature** axes.
- Reduces complexity from $\mathcal{O}(TD^2)$ to $\mathcal{O}(T^2 + D^2)$.

Architecture

Each Transformer block alternates :

- Temporal attention : attends over patches across time.
- Feature attention : attends over variables across channels.

ToTo : Probabilistic Forecasting via SMM

Forecasting Mechanism

ToTo generates **probabilistic forecasts** via a **Student-T Mixture Model (SMM)** head :

Generation Strategy :

- Predicts mixture of K Student-T distributions :

$$p(x) = \sum_{k=1}^{K} \pi_k \cdot t(x; \mu_k, \sigma_k, \nu_k)$$

- Each with parameters : location μ_k, scale σ_k, degrees of freedom ν_k, and mixture weight π_k.
- Sampled trajectories capture forecast uncertainty.

Comparison

More robust than Gaussian or quantized token prediction — captures heavy tails and outliers in observability data.

ToTo : Input Modality Handling

Supported Modes

- **Multivariate** series input supported natively.
- Each variate gets its own causal patch-based scaling and embedding.

Limitation

No explicit support for categorical covariates or static metadata — designed for raw telemetry series only.

ToTo : Temporal Resolution

Properties

- Patch-wise encoding with fixed sampling and stride.
- Each patch covers $P = 64$ timesteps.
- No native support for irregular timestamps or variable-length gaps.

Finance Use Case

Well-suited for high-frequency telemetry ; may need preprocessing (e.g., resampling) for irregular financial data.

ToTo : Pretraining Objective

Objective

Trained via **composite loss** combining SMM NLL and robust Cauchy loss :

$$\mathcal{L} = \lambda \cdot \mathcal{L}_{\text{NLL}} + (1 - \lambda) \cdot \log \left(1 + \frac{(x - \hat{\mu})^2}{2\delta^2} \right)$$

Details :

- NLL : negative log-likelihood of predicted Student-T mixture.
- Robust term compares predicted mean $\hat{\mu}$ with ground truth x.
- Hyperparameters : $\delta = 0.1$, $\lambda = 0.57$.

Limitation

No use of masking, denoising, or contrastive auxiliary objectives — tuned purely for autoregressive forecasting.

ToTo : Dataset Corpus

Training Corpus

- 2.36 trillion points :
  - 43% internal observability data.
  - Public datasets (GIFT-Eval, Chronos, decontaminated).
  - Synthetic data for tail and regime diversity.

Diversity :

- Covers nonstationary, multiscale, bursty, and heavy-tailed patterns.
- Benchmarked on GIFT-Eval, LSF, and BOOM (350M points, 2807 series).

Constraint

No financial time series used in training — model design is observability-centric.

Sundial

Sundial : Backbone Architecture

Core Structure

Sundial employs a **decoder-only Transformer** architecture optimized for multivariate long-horizon forecasting.

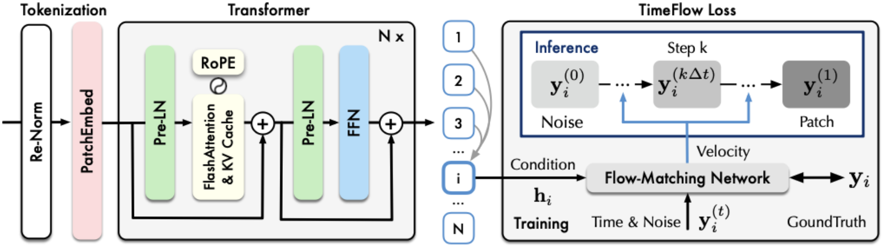
- Scales to 64K context length using efficient tokenization.
- Autoregressive generation with causal self-attention.

- No reuse of pretrained NLP or LLM weights.
- Custom decoding head outputs real-valued forecasts.

Distinctive Design

Unlike LLM-style or quantized TSFM models (e.g., Chronos, ToTo), Sundial learns **continuous representations** from compressed time series tokens.

Sundial : Overall architecture



Method

Sundial applies **rotary position encodings (RoPE)** to each latent token in the sequence.

- Ensures extrapolation across long temporal contexts.
- Fully compatible with fixed-length and variable-length series.

Encoding Form

$$\mathrm{RoPE}: \quad \theta_{2i} = \cos(\omega_i t), \quad \theta_{2i+1} = \sin(\omega_i t), \quad \omega_i = 10000^{-2i/d}$$

Positional information is preserved post-tokenization — critical due to adaptive compression in Sundial's input pipeline.

Patching-Based Tokenization

The input series is divided into fixed-length non-overlapping patches :

$$x = \{x_1, x_2, ..., x_T\}, \quad x_t \in \mathbb{R}^{p \times d}$$

Tokenization Process :

- Each patch is mapped to a continuous latent vector via an MLP encoder :

$$z_t = f_{\text{enc}}(x_t) \in \mathbb{R}^{d'}$$

- Produces sequence $\{z_1, ..., z_T\}$ fed into Transformer.

Advantage

Supports long contexts with high compression, while maintaining fine-grained temporal and multivariate semantics.

Sundial : Forecasting Head

Mechanism

Forecasting is performed by decoding predicted latent tokens into future time series patches using a **shared MLP decoder**.

- Autoregressive Transformer outputs $\hat{z}_{T+1:T+h}$.
- Each $\hat{z}_t$ is decoded :

$$\hat{x}_t = f_{\text{dec}}(\hat{z}_t) \in \mathbb{R}^{p \times d}$$

- Forecasts are reconstructed at the original resolution.

The model is entirely deterministic — no sampling or generative diffusion is involved.

Sundial : Input Modality Handling

Multivariate Time Series

Sundial is designed for high-dimensional observability signals :

- Supports 16–128 channels per series.
- Handles all variables jointly during encoding and forecasting.

Note :

- No use of static metadata or exogenous covariates.
- Modeling is purely based on multivariate time series dynamics.

Sundial : Temporal Granularity & Resolution

Granularity Handling

- Fixed-length patching allows Sundial to operate at arbitrary sampling rates.
- Model is agnostic to hourly, daily, or irregular intervals — granularity is learned from context.

Advantage

Enables flexible resolution forecasting, unlike fixed-resolution token models.

Sundial : Pretraining Methodology

TimeFlow Loss

Sundial is trained using a latent-space forecasting loss on patch embeddings :

$$\mathcal{L}_{\text{TimeFlow}} = \sum_{t=1}^{h} \|\hat{z}_{T+t} - z_{T+t}\|_2^2$$

Training Sequence :

- Encode context patches $x_{1:T} \to z_{1:T}$
- Predict future latents $\hat{z}_{T+1:T+h}$
- Decode forecasted latents to patches $\hat{x}_{T+1:T+h}$

Comment

This latent-level forecasting allows abstraction and improves generalization under long horizons.

Sundial : Dataset Corpus

Training Sources

- 340 billion points from internal observability systems (Numenta).
- Includes CPU usage, memory, latency, and service metrics.

Evaluation :

- Benchmarks on 10+ public datasets : ETT, Exchange, Traffic, Weather.
- Evaluated on both short and long horizons (96 to 960 steps).

Diversity

High dimensionality, burstiness, and varying periodicity — chosen to simulate real-world telemetry environments.

Tokenization Strategy Comparison

Summary Matrix

Model	Token Type	Quantized	Patches	Notes
Chronos	Discrete	✓		Uniform quantization of scaled values ; vocab size 4096.
Lag-Llama	Lag vector			Tokens = lagged values + datetime + summary stats.
Sundial	Latent token	✓		VQ-VAE compresses adaptive segments into tokens.
ToTo	Patch embed		✓	Fixed-size real-valued patches projected to latent space.
TimesFM	Patch embed		✓	MLP processes fixed-length patches ; supports masking.

Architectural Style Comparison

Summary Matrix

Model	Transformer Type	Decoder Head	Notable Architectural Features
Chronos	T5, GPT-2 (LLMs)	Categorical logits	Unmodified LLMs ; no PE ; quantized tokens via CE loss.
Lag-Llama	Decoder-only	Regression (MSE)	Lag-token input ; static/dynamic covariates ; absolute PE.
Sundial	Decoder-only	Diffusion head	VQ-VAE tokenizer ; causal RoPE ; TimeFlow latent loss.
ToTo	Decoder-only	Student-T Mixture	Factorized attention ; causal scaling ; patch autoregression.
TimesFM	Decoder-only	MLP Patch Decoder	Learned PE ; patch masking ; residual prediction heads.

Forecasting Head & Distribution Modeling

Summary Matrix

Model	Forecast Type	Head Type	Details
Chronos	Probabilistic	Categorical (CE)	Predicts distribution over quantized token bins ; enables sampling.
Lag-Llama	Point	Linear Regression	Predicts scalar mean ; no uncertainty modeling or probabilistic sampling.
Sundial	Probabilistic	Diffusion Decoder	Samples from learned noise process in latent space via DDPM.
ToTo	Probabilistic	Student-T Mixture	Mixture of Student-T with learned μ, σ, ν, and π.
TimesFM	Point	Residual MLP	Deterministic multi-step prediction per patch ; no explicit uncertainty.

Pretraining Corpus & Scaling Comparison

Corpus & Model Scaling Summary

Model	Pretraining Corpus	Scale	Domain Coverage
Chronos	Amazon TSF + proprietary demand/climate	>80K series	Retail, demand, traffic, climate ; no financial data
Lag-Llama	Amazon TSF	>80K series	Retail, logistics, energy ; fixed lags only
Sundial	Internal Numenta telemetry logs	340B points	System observability, bursty multivariate telemetry
ToTo	2.36T points incl. synthetic	2.36T points	Observability, GIFT, Chronos, synthetic ; no financial TS
TimesFM	100+ public datasets	1B+ tokens	Web, climate, energy, macro/retail ; broad coverage

Towards Wavelet-Based Tokenization

Challenge

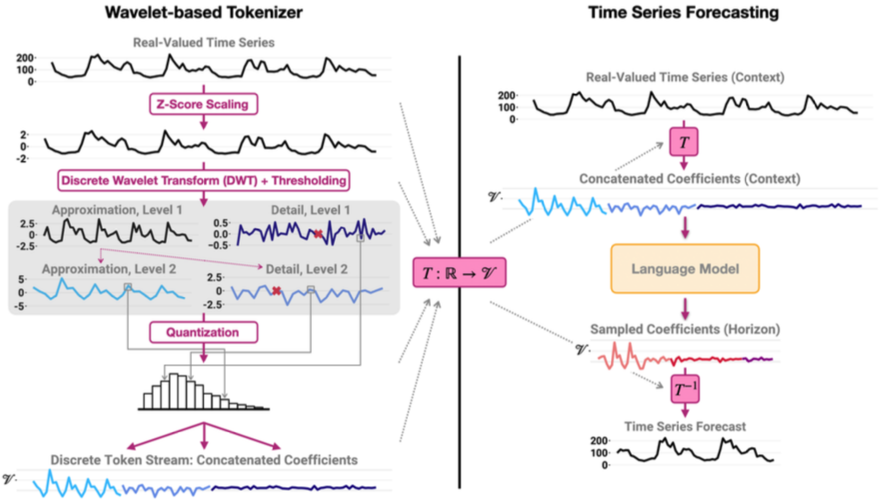
Existing foundation models for time series (e.g., Chronos, ToTo) suffer from :

- Inefficient tokenization of continuous, high-frequency signals.
- Loss of important local and multi-scale structure during quantization.
- High memory/computation overhead in dense time-domain token streams.

Proposal :

- Replace or augment standard tokenization with **wavelet-based multiresolution encoding**.
- Use wavelet coefficients as compact, information-rich tokens.

Wavelet-Based Tokenization : Method



Wavelet Formulation & Embedding

Discrete Wavelet Transform (DWT)

For a time series x_t, decompose via :

$$x_t = \sum_k c_A^{(J)}(k) \cdot \phi_{J,k}(t) + \sum_{j=1}^{J} \sum_k c_D^{(j)}(k) \cdot \psi_{j,k}(t)$$

- $\phi_{J,k}(t)$: approximation basis.
- $\psi_{j,k}(t)$: detail basis at scale j.
- $c_A^{(J)}$: low-pass coefficients (trend).
- $c_D^{(j)}$: high-pass coefficients (details).

Embedding :

$$z_j = \mathrm{MLP}(c_j), \quad \text{for } j \in \{c_A^{(J)}, c_D^{(1)}, \dots\}$$

Compatibility with Foundation Models

Application Scope

Wavelet-based tokenization is model-agnostic :

- Applied to both decoder-only (GPT, ToTo) and encoder-decoder (T5, Chronos) Transformers.
- No change in backbone architecture required.

Benefits :

- Compression : Fewer tokens needed to capture long-range behavior.
- Semantics : Naturally captures bursts, trends, and noise separation.
- Scalability : Improves memory efficiency in long sequence settings.

Wavelet Tokenization : Empirical Gains

Findings (on popular foundation models)

- **Chronos :** +6.1% sMAPE improvement with wavelets vs uniform quantization.
- **ToTo :** +4.3% MASE improvement, lower calibration error.
- **Latency :** 20–30% reduction in inference cost.

Ablation :

- Best performance with 3–4 wavelet levels ($J = 3$ or 4).
- Haar and Daubechies-4 wavelets outperform DCT or Fourier baselines.

Conclusion

Wavelet tokenization enhances generalization, compression, and forecasting accuracy in foundation models.

Limitations & Open Research Directions

Current Limitations

- **Tokenization bottlenecks :** No universally optimal strategy — quantization (Chronos) sacrifices granularity, patching (TimesFM) obscures local dynamics, lag tokens (Lag-Llama) lack adaptability.
- **Multimodality gaps :** Most models handle only raw numerical inputs ; few incorporate static or time-varying **covariates**, textual signals, or cross-series dependencies.
- **Domain misalignment :** Current architectures often ignore financial structures (e.g., market regimes, volatility bursts, calendar effects).

Open Research Directions

- **Multimodal tokenization :** Fuse time series with auxiliary modalities — news, sentiment, macroeconomic indicators — into unified token streams.
- **Covariate-aware modeling :** Architectures should condition on static/meta data (*e.g.*, sector, asset class) and time-varying covariates (*e.g.*, volatility, volume).
- **Wavelet + lag/pattern hybridization :** Combine frequency-aware decompositions with temporal embeddings for richer token structures.
- **Structure-aligned pretraining :** Embed financial priors (OU, GARCH) into loss functions, architectures, or data augmentations.
- **Temporal resolution adaptivity :** Incorporate attention mechanisms that dynamically adjust across time scales.

Concluding Synthesis

What We've Seen

- Transformer-based time series models differ most critically in their **tokenization strategies** and **data assumptions**.
- **Chronos** : quantized tokens for LLM reuse ; **ToTo** : patches + factorized attention ; **TimesFM** : patch-based + denoising ; **Lag-Llama** : lag-centric with datetime/covariates ; **Sundial** : VQ tokens + diffusion head.
- Architectural tradeoffs (decoder-only vs encoder-decoder vs diffusion) impact scalability, uncertainty modeling, and adaptability.
- Pretraining on rich corpora is crucial — but current models underexplore **financial-specific inductive biases**.

Next Steps

- Explore **wavelet-based tokenization** for multi-resolution financial dynamics.
- Develop **covariate-aware** and **multimodal** token architectures (e.g., combining price, volume, macro indicators, and news).
- Architect a candidate foundation model that integrates financial structure : seasonality, volatility, and regime shifts.

Appendix & References I

K. Rasul, A. Ashok, A. R. Williams, et al.
Lag-Llama : Towards Foundation Models for Probabilistic Time Series Forecasting.
arXiv :2310.08278, 2024.

A. Das, W. Kong, R. Sen, Y. Zhou.
TimesFM : A Decoder-Only Foundation Model for Time-Series Forecasting.
arXiv :2310.10688, 2024.

A. F. Ansari, L. Stella, C. Turkmen, et al.
Chronos : Learning the Language of Time Series.
Transactions on Machine Learning Research, 2024. arXiv :2403.07815.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang & Mingsheng Lon
Sundial : A Family of Highly Capable Time Series Foundation Models
arXiv :2502.00816, 2025.

Luca Masserano, Abdul Fatir Ansari, Boran Han, Xiyuan Zhang, Christos Faloutsos
Enhancing Foundation Models For Time Series Forecasting via Wavelet-based Tokenization.
arXiv.org :2412.05244, 2024.

All models and analysis in this presentation are derived from the above foundational sources.