Large Language Models for Time Series

Paradigms, Methods, Controversies, and the Rise of Foundation Models

From LLMTime to Chronos, TimesFM, MOIRAI, and Beyond

yuyaow@bu.edu

Department of Mathematics and Statistics

Outline

The time series problem

A time series is a sequence $\{y_t\}_{t=1}^{T}$, possibly multivariate $y_t \in \mathbb{R}^d$, often with covariates x_t.

Five canonical tasks:

- **Forecasting:** predict $y_{T+1:T+H}$ given history $y_{1:T}$.
- **Classification:** assign a label to a whole series (ECG arrhythmia, activity).
- **Anomaly detection:** flag unusual patterns (fraud, equipment failure).
- **Imputation:** fill in missing observations.
- **Representation learning:** learn features transferable across tasks.

Classical tools:

- Statistical: ARIMA, exponential smoothing, state-space models, GARCH.
- Deep learning: DeepAR, N-BEATS, Informer, Autoformer, PatchTST, TimesNet.

The issue. These models are typically *trained from scratch on a single dataset* — small models, narrow scope, no transfer. The "ImageNet moment" has not yet arrived for time series.

The "foundation model" gap in time series

NLP and vision have foundation models: GPT-4, LLaMA, CLIP, DINOv2, SAM. Pre-train once on huge data, transfer to anything.

Time series has had nothing comparable until 2023.

Why was TS late to the foundation-model revolution?

- **Data heterogeneity.** Different sampling frequencies, scales, units, domains. Unlike text (all made of tokens) or images (all pixels), TS has no universal "vocabulary."
- **No natural tokenization.** How do you tokenize a continuous real-valued signal?
- **Distribution shift.** Non-stationarity is the rule, not the exception.
- **Fragmented benchmarks.** M-competitions, Monash, ETT — small, single-domain.

Two emerging routes around these problems:

1. **Reuse LLMs:** can a pretrained language model, with clever adaptation, forecast time series?
2. **Build TS-native foundation models:** pretrain from scratch on massive TS corpora (Chronos, TimesFM, MOIRAI, ...).

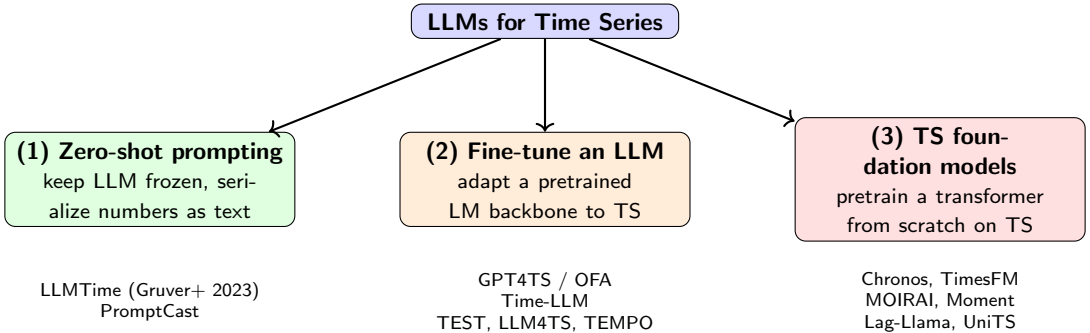
This lecture surveys both.

Three hypotheses driving the field

1. **The universality hypothesis.** Patterns that make LLMs good (compositionality, long-range dependence, hierarchical structure) also describe time series. A sequence model that handles language should handle numbers.
2. **The knowledge-transfer hypothesis.** LLMs encode prior knowledge about the *world* (calendars, weather, economic cycles) that could help forecasting.
3. **The scale hypothesis.** With enough TS data and compute, large transformers will beat bespoke models — just as in NLP.

All three are live research questions. As we will see, empirical evidence for (1) and (3) is *strong*; evidence for (2) is *mixed and contested*.

A taxonomy of approaches



Each paradigm answers a different question:

- Paradigm 1: can an *out-of-the-box* LLM forecast? (zero-shot capability)
- Paradigm 2: can we *cheaply adapt* a pretrained LLM to TS?
- Paradigm 3: should we build *TS-native* foundation models instead?

LLMTime: LLMs are zero-shot forecasters

Gruver, Finzi, Qiu, Wilson (NeurIPS 2023). A startling claim: *frozen* GPT-3 / LLaMA-2, with *no* fine-tuning, often beats specialized forecasters.

The trick: careful numerical tokenization.

- Scale the series so values lie in a fixed range.
- Encode each number digit-by-digit with explicit separators:
  `1.234, -0.56` $\rightarrow$ `"1,234|-0,56"`
- Feed this string to the LLM, let it autoregressively generate the continuation.
- Decode back to numbers.

Why does it work? The authors show the LLM induces a *prior over continuations* that is:

- roughly scale- and shift-equivariant under their tokenization,
- concentrates on *smooth, simple* continuations (matches Occam-style priors),
- can be elicited as a *full probabilistic forecast* via token-level log-likelihoods.

Empirical finding: on Monash and Darts benchmarks, GPT-3/LLaMA-2 matches or beats N-BEATS, NHiTS, ARIMA — *with zero training*.

Probabilistic forecasts from a language model

A key statistical observation. Language models are *density models*:

$$p_\theta(\text{tokens}) \;=\; \prod_t p_\theta(w_t \mid w_{<t}).$$

After serializing the series $y_{1:T}$ as a string $s(y_{1:T})$:

$$p_\theta\big(s(y_{T+1:T+H}) \mid s(y_{1:T})\big) \;\overset{\text{decode}}{\longrightarrow}\; p\big(y_{T+1:T+H} \mid y_{1:T}\big).$$

Consequences:

- We get *predictive distributions*, not just point forecasts.
- Quantile/interval forecasts come for free via sampling.
- CRPS, log-score, interval coverage — all computable.

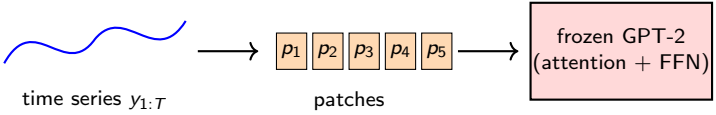
Caveats.

- Tokenization artifacts: two nearby real numbers can have very different token sequences.
- Context length: long series need truncation or sliding windows.
- Cost: calling GPT-4 at inference is *far* more expensive than running N-BEATS.

GPT4TS / "One Fits All" (Zhou et al. 2023)

A simple, strong recipe.

1. Take a pretrained GPT-2.
2. *Freeze* all self-attention and feed-forward weights.
3. Train only: input embedding layer, output projection, positional embeddings, and LayerNorm parameters.
4. Use *patching* (as in PatchTST): split the series into overlapping patches, embed each patch as a "token".



Remarkable finding. This freezing recipe matches or beats SOTA on *all major TS benchmarks* (forecasting, classification, imputation, anomaly detection) across 10+ datasets.

Lesson. A large pretrained *sequence* model provides a useful inductive bias — even if the pretraining was on text.

Time-LLM: reprogramming TS into language (Jin et al., ICLR 2024)

Goal. Use a frozen LLM (LLaMA-7B) as a forecaster by *reprogramming* patches into the language embedding space.

Architecture (three components):

1. **Patch reprogramming.** Each patch p_i is projected into the LLM's embedding space by a learned cross-attention with a small set of *text prototypes* (linear combinations of LLM word embeddings).

$$e_i = \mathrm{Attn}(p_i;\, E_{\mathrm{proto}}), \qquad E_{\mathrm{proto}} \subset \text{LLM word embeddings.}$$

2. **Prompt-as-prefix.** Prepend a natural-language prompt containing summary statistics ("series has trend: upward; min: 0.3; max: 1.8; dominant lag: 7").
3. **Output projection.** A small head maps the LLM's output hidden states back to numerical forecasts.

Empirical result. Time-LLM achieves SOTA on long-horizon forecasting (ETT, Weather, Traffic, Electricity), especially under *few-shot* regimes.

Key idea in one line: *align time-series tokens to language tokens, then reuse everything the LLM already knows about sequences.*

Other fine-tuning approaches — a quick tour

Method	Key idea
LLM4TS (Chang+ 2023)	Two-stage fine-tuning: first on autoregressive TS pretraining, then supervised forecasting.
TEST (Sun+ 2024)	Text prototypes as anchors; contrastive alignment of TS embeddings to word tokens.
TEMPO (Cao+ 2024)	Decompose into trend/seasonality/residual; encode each with soft prompts into a GPT.
PromptCast (Xue & Salim 2023)	Turn forecasting into a pure text-to-text task; prompt templates for sensor data.
S^2IP-LLM (Pan+ 2024)	"Semantic-space-informed prompt" — find prompts anchored in LLM's semantic manifold.
UniTime (Liu+ 2024)	Language-instruction-based unified forecaster across datasets of different frequencies.

Common threads:

- Freeze most of the LLM; train lightweight adapters (LoRA, prompt tokens, input/output heads).
- Exploit *patching* — borrowed from PatchTST — as the TS-to-token bridge.
- Add *text side information* (statistics, domain metadata) as a prompt prefix.

The push for TS-native foundation models

Motivation. If LLM priors are useful but language is the "wrong" substrate, why not pretrain a transformer *directly* on enormous amounts of time-series data?

Ingredients of a TS foundation model:

- **A TS tokenizer** — quantization, patching, or continuous embedding.
- **A transformer backbone** — encoder, decoder, or encoder-decoder.
- **A massive pretraining corpus** — millions to billions of series, many domains and frequencies.
- **A pretraining objective** — next-token prediction, masked reconstruction, or mixture.

Four representative models (2023–2024) we'll discuss:

- **Chronos** (Amazon) — quantize-then-T5.
- **TimesFM** (Google) — decoder-only, patched.
- **MOIRAI** (Salesforce) — any-variate, masked encoder.
- **Lag-Llama**, **Moment**, **UniTS** — other flavors.

Chronos (Ansari et al., Amazon 2024)

Key idea: treat forecasting as language modeling over quantized numbers.

1. **Scale**: mean-scale the series so values lie in a bounded range.
2. **Quantize**: map each value to one of V bins $\Rightarrow$ sequence of integers in $\{1, \ldots, V\}$.
3. **Train**: a standard T5 encoder-decoder with cross-entropy over the token vocabulary.
4. **Forecast**: autoregressively sample next-token distributions; dequantize; rescale.

Pretraining corpus. $\sim$ 28B observations across hundreds of public datasets, plus synthetically generated "TSMix" and Gaussian-process-simulated series.

Results. Chronos matches or exceeds fully-supervised baselines on a broad zero-shot benchmark — no per-dataset training needed.

Statistical interpretation. This is literally *nonparametric density estimation over integer sequences*, with the categorical distribution over tokens playing the role of the predictive.

TimesFM (Das et al., Google 2024)

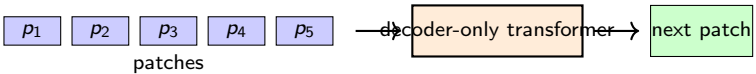
Architecture. Decoder-only transformer, analogous to GPT.

Input pipeline.

- Split series into non-overlapping patches of length P.
- Each patch is a token; residual MLP embedding projects to model dim.
- Output patches of length $P_{\text{out}} > P$ for long-horizon forecasts.

Pretraining corpus. $\sim$ 100B timepoints — Google Trends, Wikipedia pageviews, synthetic data, public TS repositories.

Loss. Mean squared error (point) + quantile losses (probabilistic).



Zero-shot results. Matches or beats supervised methods on Monash and Darts; strong long-horizon performance.

MOIRAI (Woo et al., Salesforce 2024)

Focus. Multivariate forecasting across arbitrary number of variates and frequencies.

Key innovations:

- **Any-variate attention.** Flatten (n variates) $\times$ (T timepoints) into a single token sequence; let attention learn cross-variate structure.
- **Multi-patch-size projection.** Different patch sizes for different frequencies (yearly, monthly, hourly); model auto-selects.
- **Mixture output head.** Predict a *mixture* of Student-t, negative binomial, log-normal, etc., covering heterogeneous data types.
- **Masked encoder objective.** BERT-style masking of patches; predict the masked ones.

Dataset: LOTSA. $\sim$27 billion observations over 9 domains, largest open TS pretraining corpus.

Result. SOTA on zero-shot multivariate forecasting, especially for datasets with many variates and mixed frequencies.

Other foundation models worth knowing

Model	Architecture	Distinguishing feature
Lag-Llama (Rasul+ 2024)	Decoder-only	Lag features as inputs; probabilistic (Student-t) head; univariate.
Moment (Goswami+ 2024)	Encoder-only	Masked reconstruction pretraining; family of sizes 40M/125M/385M. General-purpose embedding model.
UniTS (Gao+ 2024)	Unified seq2seq	Prompt tokens indicate task (forecast/classify/impute) — one model, many tasks.
TimeGPT-1 (Nixtla 2023)	Proprietary	Commercial TS foundation model via API; popularized the concept.
Timer (Liu+ 2024)	Decoder-only	Autoregressive pretraining on 1B-point corpus; strong few-shot.

Common design patterns:

- **Patching** is now universal — borrowed from ViT/PatchTST.
- **Scale normalization** (instance norm, RevIN) is essential for cross-domain transfer.
- **Probabilistic heads** are the norm — point forecasts alone are insufficient.
- Many models published their pretraining data (LOTSA, Monash Archive) to spur open research.

Why might LLM-style models work for time series?

Three intellectual threads converge:

1. **Sequence-model universality.** Transformers are universal sequence-to-sequence learners; time series are just sequences. Architectural mismatch is small.
2. **Compositional structure.** Many time series decompose into

$$y_t \;=\; \mathrm{trend}_t + \mathrm{season}_t + \mathrm{noise}_t,$$

   echoing hierarchical compositions in language (morpheme $\to$ word $\to$ sentence).
3. **Patches are tokens.** The PatchTST insight: contiguous windows of a TS behave like subword tokens — locally meaningful, globally combinable.

A mental model. Pretraining learns a rich *prior* $p_\theta(y_{1:T})$ over TS, which at inference is conditioned on history:

$$p_\theta(y_{T+1:T+H} \mid y_{1:T}) \;\propto\; p_\theta(y_{1:T+H}) \big/ p_\theta(y_{1:T}).$$

A good prior $\Rightarrow$ good zero-shot forecasts. Fine-tuning refines the prior for a target domain.

Scaling laws — do they hold for time series?

In NLP (Kaplan; Hoffmann) loss follows a power law:

$$L(N, D) \approx \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}} + L_{\infty},$$

with N parameters, D tokens.

Recent work on TS scaling laws:

- **Edwards et al. (2024)**, **Shi et al. (2024)**, **Yao et al. (2024)** empirically fit similar laws for Chronos-style models.
- Observed exponents are *similar in magnitude* to NLP but with different constants.
- Data bottleneck: TS pretraining corpora are ∼10B–100B tokens — small compared to the trillions used in LLM pretraining.

Consequence for practice. Model capacity alone isn't the limit; *diversity and quality of TS data* is. This motivates synthetic augmentation (TSMix, KernelSynth) and cross-domain corpus construction.

Open question. Is there an irreducible loss floor L_{∞} that is *much higher* for TS than for language? If so, how low can it go?

A cautionary result — Tan et al. (ICML 2024)

Paper: "Are Language Models Actually Useful for Time Series Forecasting?"

Experiment. For three popular methods (GPT4TS, Time-LLM, LLaTA), replace the entire LLM backbone with:

1. an identity function,
2. a randomly initialized transformer of matching size, or
3. a simple attention module.

And compare forecasting accuracy.

Finding. On most benchmarks, *removing the LLM does not hurt — and sometimes helps.*

Interpretation.

- The *input pipeline* (patching, normalization) and *output head* may do most of the work.
- Pretrained LLM *knowledge* about calendars, economics, weather, ... often doesn't transfer to raw numeric forecasting.
- Compute and latency cost of a 7B-parameter LLM is rarely justified over a 1M-parameter TS-specific model.

Caveat. The paper focuses on *numeric-only* benchmarks. For tasks with rich *text metadata*, LLMs still add clear value.

The verdict as of 2025–2026

Consensus emerging from follow-up work:

- For *pure numerical forecasting*, **TS-native foundation models** (Chronos, TimesFM, MOIRAI) tend to outperform LLM adaptations at comparable cost.
- For *text-augmented forecasting* (news + prices, clinical notes + vitals), **LLM adaptations** retain a genuine edge — they can reason over both modalities.
- Paradigm 1 (zero-shot prompted LLMs) is the *right baseline* when no TS data is available, but rarely the deployed solution at scale.

A cleaner statement of what we've learned:

- *Scale and pretraining* on sequence data are useful for TS; *scale and pretraining on language* are mostly not (for pure numeric tasks).
- The *transformer* inductive bias transfers well; *LLM knowledge* transfers narrowly.
- The relevant foundation-model pretraining corpus is *TS itself*, not the Internet.

When text truly helps: multimodal time series

Real problems often involve both numbers and text:

- **Finance:** prices + news headlines + earnings transcripts.
- **Healthcare:** vital signs + clinician notes + lab reports.
- **Retail:** sales + product descriptions + promotional calendars.
- **Energy:** demand + weather forecasts (natural language) + event schedules.

Recent lines of work.

- **Time-MMD / TimeMMD (Liu+ 2024)**: benchmark of multimodal datasets; show text modestly helps forecasting.
- **GPT4MTS, MM-TSFM**: cross-attention between numeric and text encoders.
- **FinGPT, BioGPT-TS**: domain-specific LLMs with native TS tokens.

Key advantage of LLM backbones here. They handle the text modality natively, and can integrate TS via adapters — a much harder architectural lift for TS-native models.

$\Rightarrow$ *multimodal* is where the LLM vs. TS-foundation-model debate is most alive.

LLMs as time series *reasoners*

Beyond forecasting, LLMs can perform *reasoning* about time series:

- **Anomaly explanation.** "Why is today's sensor reading abnormal?" $\rightarrow$ LLM narrates the context.
- **Causal questions.** Suggest hypotheses linking events to observed dynamics.
- **Natural-language querying.** "Plot the weekly average of series X over the last year, excluding holidays."
- **Report generation.** Summarize trends, anomalies, and forecasts in text.

Agentic systems.

- **TS-Agent, ChatTime, TSGBench**: LLMs as orchestrators of specialized TS tools (forecasters, anomaly detectors, visualizers).
- **Code-interpreter workflows**: LLMs write and run TS analysis code, then interpret results.

Benchmarking reasoning. Early benchmarks (Merrill+ 2024; TimeBench) reveal LLMs *struggle* on multi-step numerical reasoning over raw series — pointing to a major research opportunity.

How should we evaluate?

Standard point-forecast metrics: MAE, MAPE, sMAPE, MSE, NRMSE.

Probabilistic metrics:

- **CRPS (Continuous Ranked Probability Score):** $\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(z) - \mathbf{1}\{y \le z\})^2 \, \mathrm{d}z$.
- **Quantile loss / pinball loss** at multiple levels.
- **Coverage and interval width** (reliability diagrams).
- **Log-likelihood / NLL** where the model provides a density.

Benchmark landscape (2024–2026):

- **Monash Forecasting Repository** — classical, single-frequency aggregated.
- **GIFT-Eval** (Salesforce) — designed specifically for foundation models; 23 datasets.
- **GluonTS benchmarks** — probabilistic focus.
- **LOTSA** — MOIRAI's training corpus; also used for holdout eval.
- **ETT, Weather, Traffic, Electricity** — long-horizon standards.

Open challenge: *zero-shot* evaluation protocols with strong domain leakage controls are still immature.

Domain applications driving the field

Finance. High-frequency price forecasting, portfolio construction, volatility modeling, text-driven return prediction. Constraints: non-stationarity, low SNR, adversarial regime changes.

Healthcare. ICU vital sign prediction, early warning scores, sepsis onset, ECG/EEG foundation models (BioSignal GPT). Constraints: irregular sampling, missingness, privacy.

Climate & weather. Pangu-Weather, GraphCast — strictly speaking spatio-temporal; LLM-style pretraining is spreading to climate reanalysis data.

Energy. Load forecasting, renewable generation, grid anomalies. Heavy reliance on probabilistic forecasts.

Industrial IoT. Predictive maintenance, RUL estimation, anomaly detection at scale across millions of sensors — the *natural habitat* for foundation models.

Retail & demand. Heterogeneous items, cold-start new SKUs — zero-shot forecasting is particularly valuable.

Frontier challenges (selected)

1. **Theory of transferability.** When does a TS foundation model transfer zero-shot to a new domain? Analog of the graphon transferability theorem for time series.
2. **Inductive biases vs. scale.** When is it worth hard-coding structure (seasonality decomposition, state-space priors) vs. scaling parameters?
3. **Uncertainty and calibration.** Conformal prediction for sequence models (online CP, distribution-free bounds), calibrated predictive distributions under non-stationarity.
4. **Long-context & long-horizon.** Efficient attention variants (Mamba, SSM hybrids); hierarchical decoders; memory mechanisms.
5. **Causality and interventions.** Forecasting under counterfactual policies; LLMs as world models for TS.
6. **Privacy and federated pretraining.** Sensitive domains (health, finance) cannot pool data — need private TS foundation models.
7. **Benchmark hygiene.** Contamination of "zero-shot" benchmarks by pretraining data is endemic; better evaluation protocols are a priority.

A statistical research agenda

Classic questions, freshly important:

- **Minimax rates** for sequence prediction under non-parametric function classes, matched to foundation-model architectures.
- **Nonparametric identification** of what a TS foundation model has learned — what are its implicit priors over trend, seasonality, shocks?
- **Mis-specification and robustness.** How does Chronos behave under heavy tails, regime shifts, structural breaks? Domain adaptation theory for sequence models.
- **Two-sample and anomaly testing** leveraging foundation-model embeddings as test statistics.
- **Uncertainty**: distribution-free prediction intervals (online conformal), full posterior summaries, Bayesian model averaging over pretrained backbones.
- **Causal inference** on time series with interventions (interrupted time series, synthetic controls) using foundation-model representations as nuisance components.

Why statisticians belong here. The field is dominated by benchmark-chasing; careful statistical thinking (identifiability, efficiency, calibration, inference) is in short supply — and in high demand.

Takeaways

1. **Three paradigms**, each answering a different question: prompt a frozen LLM, adapt a pretrained LLM, pretrain a TS-native foundation model.
2. **Zero-shot forecasting** is real — LLMTime, Chronos, TimesFM, MOIRAI all work without per-dataset training.
3. **Where LLM knowledge helps** is *text-rich multimodal tasks*, not pure numerical forecasting.
4. **TS foundation models** — pretrained on billions of timepoints — are the emerging default for large-scale forecasting.
5. **Scaling laws** appear to hold for TS; the bottleneck is corpus diversity, not compute.
6. **Theory, calibration, and evaluation** lag architectural progress — and offer strong research opportunities for statisticians.

Big-picture reframing. What matters is not that the models came from NLP, but that *sequence modeling at scale* finally arrived in time series. The "ImageNet moment" is here — for better and occasionally for worse.

Key references

Surveys:

- Jiang et al. (2024) *Empowering Time Series Analysis with LLMs: A Survey*.
- Jin et al. (2024) *Position Paper: What Can LLMs Do for Time Series?*
- Zhang et al. (2024) *Large Models for Time Series and Spatio-Temporal Data: A Survey*.

Landmark methods:

- Gruver, Finzi, Qiu, Wilson (NeurIPS 2023) — LLMTime.
- Zhou et al. (NeurIPS 2023) — GPT4TS / One Fits All.
- Jin et al. (ICLR 2024) — Time-LLM.
- Ansari et al. (2024) — Chronos.
- Das et al. (ICML 2024) — TimesFM.
- Woo et al. (ICML 2024) — MOIRAI and the LOTSA corpus.
- Rasul et al. (2024) — Lag-Llama.
- Goswami et al. (2024) — Moment.

Critical views:

- Tan et al. (ICML 2024) — *Are Language Models Actually Useful for Time Series?*
- Merrill et al. (2024) — benchmarking LLM numerical reasoning on TS.