Cascade

Learning the Grammar of Time

Version 1.0.0 · July 18, 2026

1 The grammar of time

Everything that matters unfolds in time. A heart beating, a grid balancing, a market turning, a disease spreading, a machine wearing towards the moment it fails. Complexity, wherever we find it, is not a static thing; it is a process. To understand a complex system at all is to understand how it moves.

We have never had a general instrument for that. We have models of particular systems, built one at a time, each fluent in a single dialect and mute everywhere else. What we have never had is something that speaks the underlying language: the deep regularities that a heartbeat and a power grid and a river share, not because they resemble one another but because each is a system changing under constraint, and such systems rhyme.

Our premise is that this language can be learned. That there is a grammar of time, and that a machine can be taught it. A machine that has it does not need to have met your system before. It can forecast what comes, anticipate what breaks, and simulate what never was.

Cascade is how we intend to find the curriculum and train the state-of-the-art model that learns it. The small models it races today are the search tool; the end goal is a large, open time-series foundation model trained through Cascade. This document explains the prize, the obstacle, the idea that gets around the obstacle, and why an open competition on Bittensor is the right machine for the job.

2 The prize: good forecasting is a luxury good

Almost everything in the world gets forecast. Almost nothing gets forecast well.

The gap between those two sentences is the opportunity. Every hospital already guesses at next week's admissions. Every small manufacturer already guesses at next quarter's demand. They do it with a spreadsheet trend, last year's number plus ten percent, or the instinct of whoever has been there longest. What they do not have is a model, because a model has always meant a specialist team, months of work, and a budget.

So a threshold forms. Above it, where the stakes justify a project, forecasting is excellent: banks have risk models, grid operators have demand curves, large retailers have planning teams. Below it, forecasting is a guess wearing a spreadsheet. The regional water utility, the mid-size hospital, the thousands of machines on a factory floor, the clinic, the farm, the warehouse: none of them are unimportant, and all of them are running on instinct, because building a bespoke model for each has never been worth it. Good forecasting is a luxury good, rationed by cost, and the ration excludes nearly everyone.

A machine with the grammar changes that arithmetic, because it does not need to be rebuilt for each new system. Trained once, it can forecast across many kinds of series, including series it has not seen before. It collapses the cost of a competent forecast from a project to a query. It will not beat the bank's flagship model on the bank's flagship problem, and it does not need

to. What it does is bring a serious forecast to the millions of processes that have only ever had a guess.

2.1 What that unlocks

The consequences are unglamorous, and they are everywhere.

Energy. Grids must match supply to demand every second, and the margin for error is now thinner than it has ever been, because wind and solar arrive when they choose. Better forecasts of demand and generation mean less spinning reserve burnt as insurance, fewer emergency purchases at panic prices, and more renewable energy actually used instead of curtailed.

Healthcare. A hospital that can see next week's admissions rosters nurses to match, instead of running short on Tuesday and paying agency rates on Thursday. Beyond the ward, the same capability reads a patient's vitals as what they are, a system moving through time, and the useful question is not what the heart rate is now but where it is heading.

Industry and supply chains. A factory that can forecast each machine's vibration and temperature services the one about to fail, rather than servicing everything on a calendar and still being surprised. A manufacturer that can predict its own demand holds less dead stock and misses fewer orders. Multiply that across every warehouse, every route, every SKU.

Water, agriculture, and climate. Rivers, reservoirs, soil moisture, crop yields, and local weather are all processes evolving under constraint. Most of them are monitored far more than they are modelled, and the sensors are already in the ground.

Finance and the public sector. Cash flows, claims, tax receipts, transport loads, and staffing needs are forecast today at whatever quality the institution can afford, which for most institutions is not much.

None of these are exotic problems. They are ordinary, they are everywhere, and they are almost all currently handled by guesswork, because the cost of doing better has always exceeded the value of doing better for any single one of them. Drop that cost far enough and the calculation inverts everywhere at once. And because a single model serves all of them, what is being built is one piece of infrastructure rather than a thousand bespoke projects.

It is tempting to size this against the market for forecasting software, and that would be a mistake, because it measures the wrong thing. That market is the thin band above the threshold: the organisations already rich enough to buy the luxury good. It is not the prize. It is the rounding error.

Consider instead what has happened to the world in the last two decades. We instrumented it. Meters, sensors, wearables, telemetry, logs, tick data, satellites: almost everything of consequence now emits a timestamped stream, and the streams are recorded by default. What has not kept pace is the modelling. We have built a planetary instrument for observing time and left most of the readings unread.

That is the surface a general forecaster addresses: not a segment of the software industry, but a fraction of the value of every decision made under uncertainty about what happens next: every roster, every reorder, every maintenance call, every hedge. The opportunity extends to almost any consequential process that changes over time.

This class of model is real. It is called a time-series foundation model, the race to build the best one is on, and there is not yet an uncontested winner.

3 The obstacle: there is no internet of time

Language models had the internet. Trillions of words, already written, free for the taking. Scale the model, scale the data, and the thing gets better.

Time has no equivalent. Historical data is scarce, fragmented, and mostly private, locked inside the companies that generated it. Worse, the domains that matter most are often the

thinnest: a rare medical event, a long climate cycle, a macroeconomic series produces a handful of points a year, not billions. And the tricks that expand data elsewhere do not work here, because chopping and flipping a time series destroys the very thing that makes it meaningful, which is the order things happened in.

So the scaling recipe that built modern AI hits a wall. You cannot collect your way out. Everyone racing to build a general forecaster runs into the same shortage, and it is the reason no one has yet built one that is unambiguously good.

4 The way around it: teach the grammar, not the data

Here is what the field has discovered, and it is the most interesting fact in this document.

You may not need real data for most of pretraining. You need data that teaches the grammar. Because the world's processes rhyme, the shared structures can be written down and generated on demand: the ways systems cycle, drift, shock, saturate, and recover. Train a model on enough of that invented behaviour and it learns the language itself, so that when it finally meets a real series it has never seen, it recognises the shape of what is happening and can carry on the sentence.

There is real evidence behind this. Some of the strongest recent forecasting models are trained largely or entirely on data that never happened, and models trained purely on synthetic systems have matched or beaten far larger rivals on real traffic and weather data they were never shown. The companion technical paper reviews that work and sets out Cascade's own experimental basis.

Which reframes the race. Architecture still matters, but the decisive under-explored variable is the curriculum: which dynamics to show the model, in what proportions, at what difficulty, in what order. Nobody has a settled answer. Today the largest labs guess: a team picks some synthetic patterns, tries a few combinations, ships the best one they found. That is a small number of guesses from a small number of people, in a space that is effectively infinite.

5 Cascade: run the search as an open competition

A guessing game with an unbounded search space and a cheap, objective test is exactly the sort of problem you should not solve with one team. You should run it as a competition.

Cascade fixes the model and publishes the exact training procedure, then invites anyone to submit a data recipe: a small program that generates training series. Every recipe is trained into the identical model under identical conditions, and whichever produces the best forecaster on real-world data takes the crown. It runs continuously, day after day, and the reigning champion is always the best curriculum anyone has yet found.

The economics of entry are the point. A competitor writes a program, not a model. No expensive hardware, no proprietary dataset, no permission. In a field where serious participation normally costs millions, the barrier is a laptop and an idea. So the search is run by everyone, it never stops, and it gets better as more people join, which is the opposite of how the labs are doing it.

Three design choices stop it being gamed.

1. **Everyone sits the same exam, and nobody has seen the questions.** Entries are graded on recent real-world data that is kept private, rotates constantly, and did not exist when the entry was submitted. You cannot memorise your way to a good score on data from the future, and a curriculum that wins once must keep performing as the data changes.

2. **Only the data differs.** Every entry is trained into the same model with the same settings and the same compute. That makes the submitted curriculum the main experimental variable: when an entry wins, the advantage came from its data.

3. **Anyone can audit the result.** Every round is recorded, signed, and published with the artefacts needed for independent audit under Cascade's verification protocol. The technical paper specifies that protocol in full: which artefacts are public, how the hidden evaluation data is committed in advance, how validators reproduce the computation, and how disagreements are resolved. Most claims about model quality ask you to trust the organisation making them. Ours do not.

6 Why it belongs on Bittensor

Emissions to a subnet are well spent when the subnet produces something the outside world wants, and when the work genuinely suits a decentralised network. Cascade fits on both counts.

The work is a search over an enormous space of ideas. It does not benefit from one team thinking harder; it benefits from many independent people trying different things and being measured honestly, which is precisely what the network provides. Miners propose curricula; validators measure the forecasting performance those curricula produce; emissions reward results that improve on the current best. Contributors need no training hardware and no private data, so the pool of possible participants is large. And because winning generators are public, other miners can inspect them, fork them, and try to improve them, so every round's result is a public artifact rather than a private one.

The output is legible: an open forecasting model that improves continuously, a verifiable record of what taught it, and a benchmark that says how good it is. Those are things a market can price.

7 What it produces, and how value accrues

The model Cascade produces is open and free to use. That is how it gets built, adopted, and trusted, and it lets outside users test the claims for themselves. But it raises the obvious question: if the model is free, where is the business?

The weights are unlikely to be the durable source of value. The open model is the distribution layer. Value may accrue around reliable hosted inference, private deployment and adaptation, and the benchmark and verification infrastructure that Cascade maintains. Domain-specific products and partnerships may also emerge as the model is adopted.

It is too early to know which of these routes will matter most. The durable position is the ability to keep improving the model and to show, with a credible record, where those improvements came from. Three assets support that ability, and they compound rather than depreciate.

The library of the grammar. Every round records which construction of dynamics trained the best forecaster, along with its lineage and measured performance. This is the accumulated output of a continuous global search, and it is the input every future forecasting model needs. A competitor can copy any single winning generator. They cannot fork the contributor base, the continuing stream of experiments, or the years of results showing which ideas held up.

The benchmark. Cascade is judged on a live stream of real-world data held under a strict cutoff and rotated constantly. Forecasting claims are easy to make when the test set is convenient or well known; they mean something when the model is tested on data that arrived after it was submitted. A trustworthy live benchmark is scarce in this field and independently valuable.

The verification layer. Every result is published with the artefacts needed for independent audit under Cascade's verification protocol. In a market where model claims are taken on faith,

checkable numbers matter most to the regulated buyers who most need forecasting and can least accept an unauditable black box.

8 How it grows

Cascade starts with data because it is the highest-leverage variable and can be tested cheaply on small models. The small models are part of the search process, not the endpoint. The path from here is deliberate.

1. **Compete on data.** Miners propose synthetic curricula; Cascade trains byte-identical models on each and rewards the curricula that produce the strongest real-world forecasting performance.
2. **Show it holds at scale.** Train winning curricula across larger model sizes to confirm that the ordering found cheaply at small scale still holds, so cheap experiments can predict expensive results.
3. **Make progress compound.** Periodically freeze the champion and let every competitor build on top of it, so the contest rewards continuous improvement rather than one lucky discovery.
4. **Open more of the process.** Once data quality can be measured reliably, widen the competition so participants also improve the training process and the model itself.
5. **Train the large model.** Use the strongest accumulated curriculum and training process to build a large, open, state-of-the-art time-series foundation model through Cascade.

Current status: Cascade is live on Bittensor mainnet. Miners are proposing curricula and validators are measuring the forecasting performance they produce under a fixed protocol. The immediate work is to establish a reliable baseline, broaden participation, and demonstrate that open competition produces sustained gains.

Further out, the ambition runs past forecasting, back to where this document began. A machine fluent in how systems move can be asked harder questions than what comes next: what is moving towards failure, what intervention might change the outcome, and what would follow in a world that never occurred. Getting there means a model that reads and writes across time series, language, and images, not numbers alone.

9 The bottom line

Good forecasting has always been a luxury good, rationed by the cost of building a model for each new problem. A machine that forecasts anything, instantly, removes the rationing, and the value lands across the vast neglected majority rather than at the top.

Such a machine is built not from data collected but from a grammar taught, and nobody has systematically searched for the best way to teach it. Cascade runs that search in the open, continuously, with every result verifiable by anyone, and keeps what it finds as a public asset.

The forecasting model is the near-term product. The grammar is the prize: a system that has genuinely learned how things move can forecast what comes, anticipate what breaks, and simulate what never was. Time is the one dimension every complex system shares. We think it can be learned, and that it should be learned in the open, by anyone who can contribute, with every result on the record.

This document is a plain-language overview intended for general audiences. It describes a research and engineering programme and the reasoning behind it. It is not investment advice, not an offer or solicitation of any kind, and not a prediction of financial returns. A companion technical white paper sets out the mechanism, the evidence, the limitations, the security assumptions, and the supporting literature in full detail.