Imagine an AI agent that acts over time.

At every step, it reads the series so far and decides what to do next.

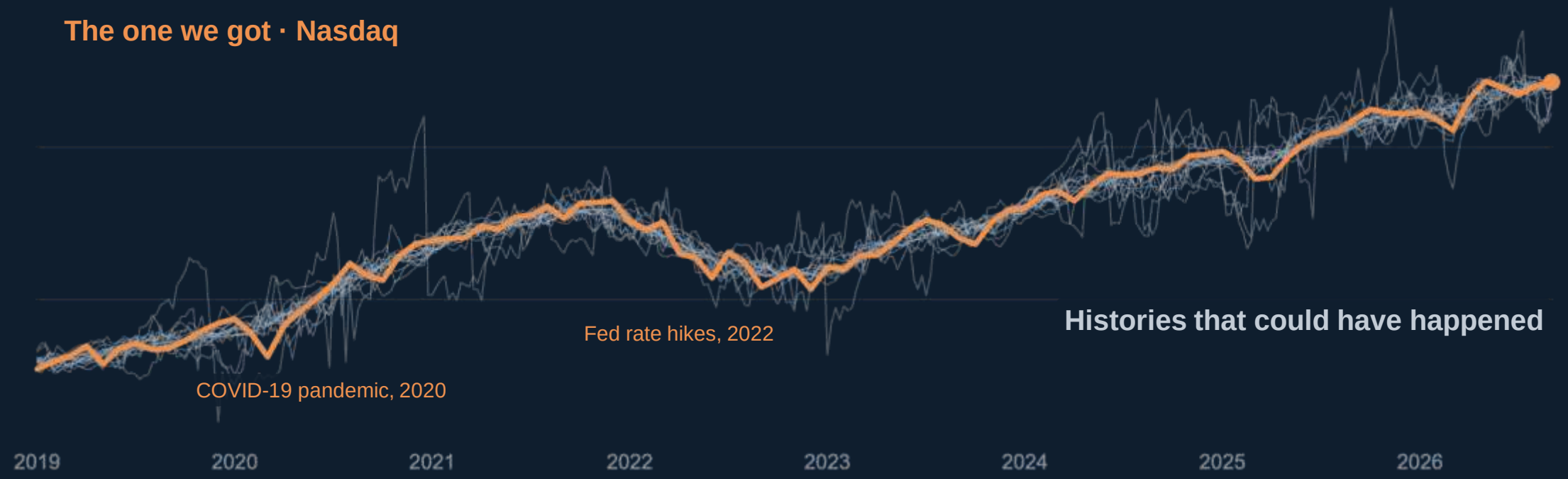


MANY TASKS LOOK LIKE THIS

- Trading and hedging
- Charging a grid battery
- Planning inventory
- Controlling a factory process

We have one history.

One path is not enough to train an agent. But the market could have written many others.



Orange: Nasdaq Composite, monthly close, Jan 2019 – Sep 2026 (Yahoo Finance), log scale. Grey: illustration, the same smoothed course with different weekly details.

NEURIPS 2026

TimeTok

Granularity-Controllable Time-Series Generation via Hierarchical Tokenization

Seokhyun Lee*, Jaeho Kim*, Changjun Oh, Mihaela van der Schaar, Changhee Lee

Korea University · University of Cambridge · *equal contribution

Overview: Six short chapters

01

Why generation matters

Agents rehearse in generated worlds, and need futures that follow a view

02

What’s missing

The same gap in finance, healthcare and industry

03

Our goal

One framework for generation at any level of detail

04

Modeling

Why raw signals fail, and why ordered tokens work

05

Results and use cases

Benchmarks, sketches, and what-if analysis

06

Our vision

Where this goes, and three takeaways

Why Generation: Train agents in a learned market

REAL HISTORY



~5,000 days

Twenty years of daily data: only twenty one-year paths.

→

MARKET GENERATOR



Learns how markets move

A VAE on path signatures, fitted to that history.

→

SIMULATED MARKETS

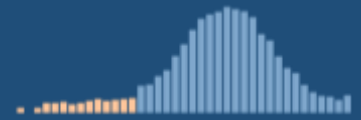


As many paths as needed

Statistically close to the real paths.

→

HEDGING AGENT



Practises in them

Learns to shrink the orange tail: its worst losses.

IN RL TERMS generator = world model | agent = policy trained inside it

→ Time-series generation builds a world in which decisions can be rehearsed.

Buehler, Horvath, Lyons, Perez Arribas, Wood, “A data-driven market simulator for small data environments”, 2020 (S&P 500, ~5,000 days). Use of the generator to train J.P. Morgan’s deep-hedging engine: Risk.net, 2022. Thumbnails are illustrative.

Why Generation: The future is decided off the chart

PEOPLE AND LLMS KNOW · THE BIG PICTURE

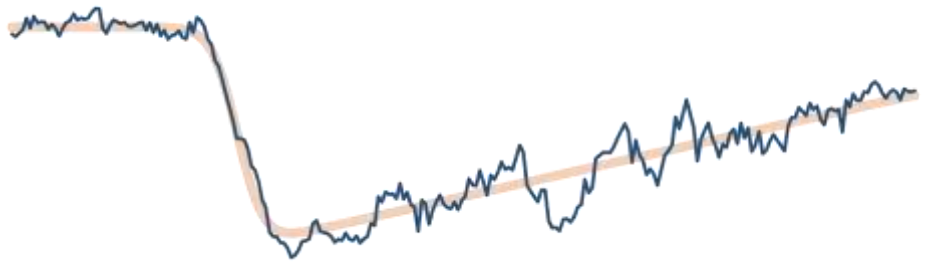
Earnings | Product launch | Rate decision | Election



“A sharp drop after the announcement, then a slow recovery over the quarter.”

ONLY DATA KNOWS · THE DAYS INSIDE IT

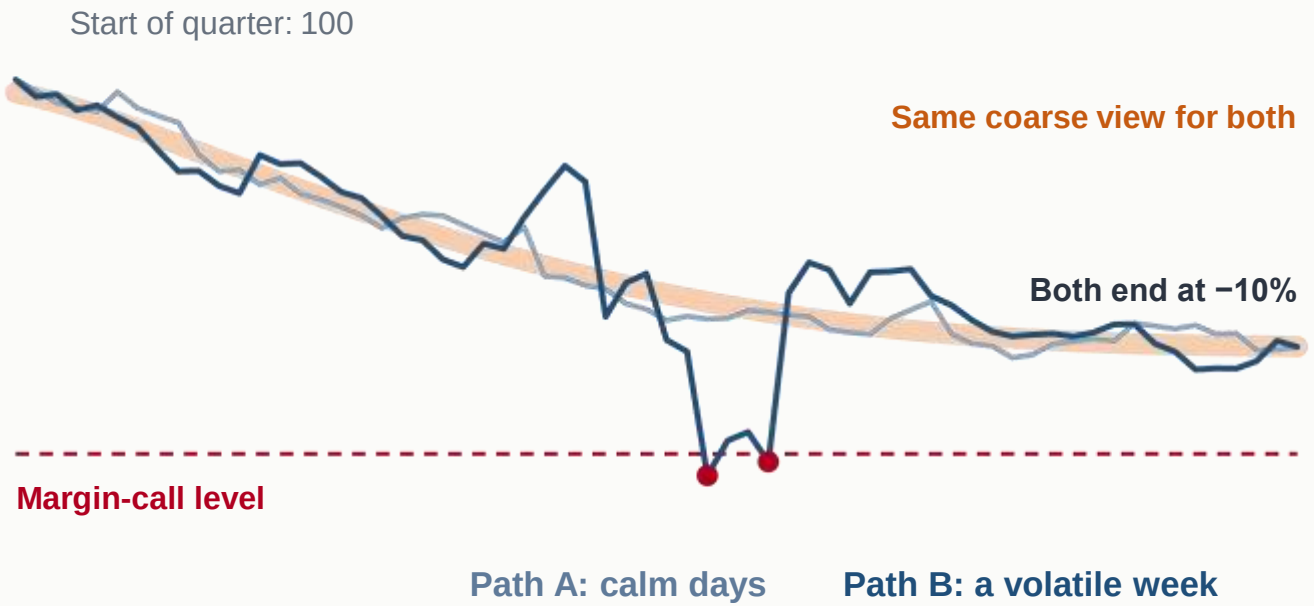
Nobody has a view on next Tuesday’s noise.



Fat tails, jumps and calm-then-wild stretches: learned from price history, not from the news.

→ ***The agent we need is one that does well in the futures that follow our view.***

Why Generation: Same big picture, very different days



Both paths follow the same coarse view: **−10%** over the quarter.

Path B has one volatile week. Its daily swings touch the margin-call level, and a leveraged position is closed out.

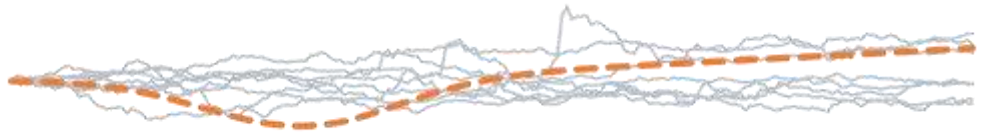
Margin calls, stop-losses and barrier options depend on the daily details, not the big picture.

→ ***What we need: many realistic daily paths that all follow one coarse view.***

What’s Missing: Today’s generators replay history

1 · TODAY’S GENERATORS

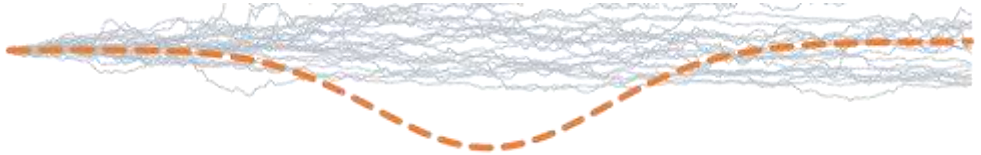
Can’t be told which future to follow



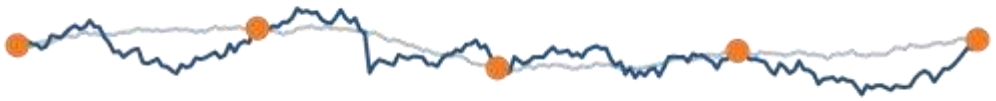
Samples (grey) wander anywhere, not along your view (orange).

2 · GENERATE MANY, KEEP THE MATCHES

Fails exactly for new scenarios



Samples stay near the past. For an unseen crash, none match.



3 · FILL IN THE GAPS WITH A BROWNIAN BRIDGE

The same random noise every day

Bridge (grey) between coarse points (orange). A real series (navy) mixes calm stretches and sudden jumps.

Missing piece · learned generation that takes a coarse series as its condition.

What's Missing: The same gap in healthcare

IN THE HOSPITAL

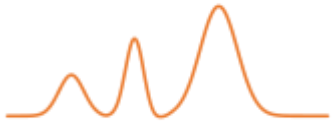


Hospital-grade ECG

→ **Certified classifier**
trained and clinically validated on this

→ **✓ Reliable**
the detail it learned from is there

IN DAILY LIFE



Smartwatch or low-cost device

→ **Certified classifier**
the same model, unchanged

→ **! Unreliable**
the sharp detail it relies on is missing

→ ***The model needs detail that low-cost devices cannot record.***

One smartwatch signal. Many hospital-grade possibilities.



Normal shape

Notched peak

Extra narrow spike

Fast ripple

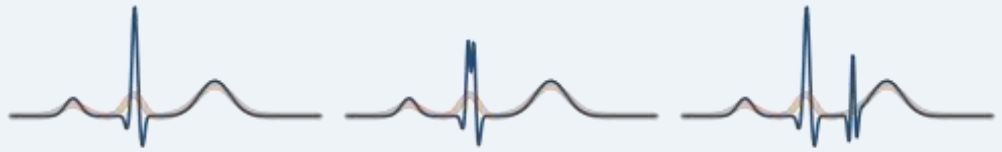
Illustration with synthetic signals. Orange: one low-detail smartwatch series. White: hospital-grade series that all smooth to (almost) exactly that orange curve.

What's Missing: Refine first, then use the certified model



Smartwatch signal

→



TimeTok: many plausible hospital-grade candidates

→

Certified classifier

unchanged, run on each candidate

The candidates agree

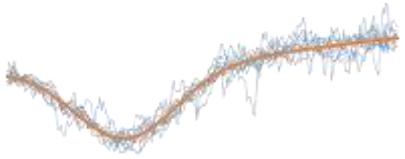
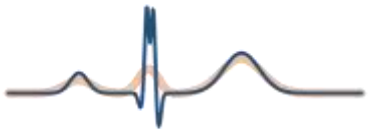
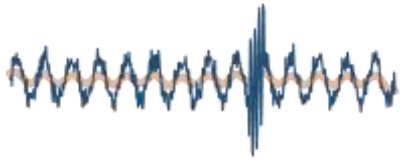
The smartwatch signal was enough. Act on the result.

The candidates disagree

The missing detail matters. Ask for a hospital-grade ECG.

→ ***The certified model stays untouched, and you learn when to measure again.***

What’s Missing: One pattern across industries

	Finance	Healthcare	Vehicles and machines
			
Have (coarse)	A view of the quarter	A smartwatch signal	Field sensors in real driving
Need (fine)	Daily price paths	Hospital-grade ECG	Lab-grade sensor detail
Can’t change	Risk engines, trained agents	Certified classifiers, clinicians	Diagnostics built on lab data
Why many samples	Losses hide in rare paths	Diagnoses hide in brief details	Faults are brief transients

→ Needed everywhere: coarse-to-fine generation that returns a distribution, not a single guess.

Our Goal: One framework, any level of detail

01

Generate from scratch



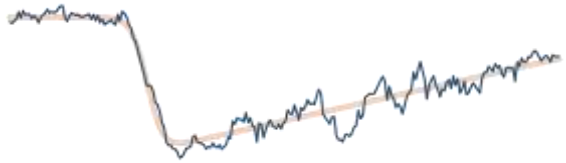
No input. Realistic series at whatever level of detail you ask for.

02

Refine a coarse input into many detailed series

REHEARSE THE FUTURE · E.G. FINANCE

From a view to daily paths



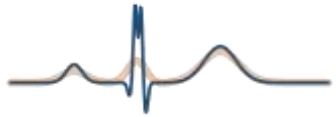
In: a view of the quarter (analyst or LLM)

Out: realistic daily paths that follow it

Same need in energy, retail demand, supply chains

READ THE PRESENT · E.G. HEALTHCARE

From a smartwatch to the hospital



In: a low-detail smartwatch ECG

Out: plausible hospital-grade signals

Same need in vehicles, factories, IoT sensors

Both from one model. Standard generation is just the case with no input.

Our Goal: One model for every granularity pair

DEFINITION · GC-TSG

Learn one generative model for every pair of levels, coarse in and fine out:

$$p\left(\mathbf{x}^{(j)} \mid \mathbf{x}^{(i)}\right), \quad j > i$$

Input level “none” means generation from scratch.

Gen · standard generation: one cell

GC-TSG · every shaded cell, one model

One model per cell

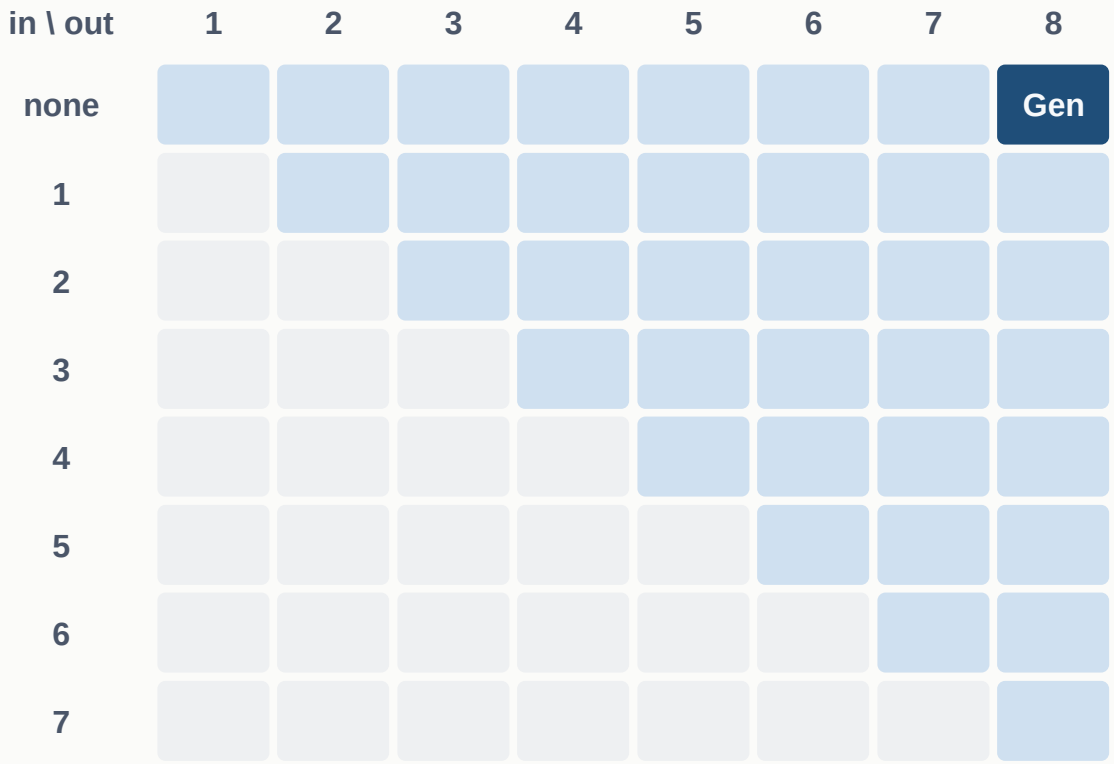
36 models

8 from scratch + 28 refinements

TimeTok

1 model

all levels consistent with each other



Related Work: Where existing work stands

		Generates series	Any output level	Refines coarse input	Ordered coarse-to-fine tokens
Time-series generators	TimeGAN, Diffusion-TS, ImagenTime, TimeVQVAE	✓	–	–	–
Multi-scale time-series generation	TimeMAR (WWW ’26)	✓	–	–	in latent space only
Time-series tokenizers	TOTEM, Chronos	–	–	–	–
Multi-scale image generation	VAR, FlowAR	images	–	–	in latent space only
Ordered image tokenizer	FlexTok	images	–	–	ordered, no explicit levels
TimeTok	This work	✓	✓	✓	✓

→ ***Each line covers one piece. TimeTok is the first to cover all four.***

Modeling: Generation is refinement, coarse to fine

1 · WHAT WE OBSERVE

Only the finest level



Training data is just x, at level 8.

→

2 · WHAT WE COMPUTE

Every coarser level

$$\mathbf{x}^{(\ell)} = R_\ell(\mathbf{x})$$

Real-world operators, e.g. smoothing or downsampling. Deterministic: no new information.

→

3 · WHAT FOLLOWS

Generation = refinement

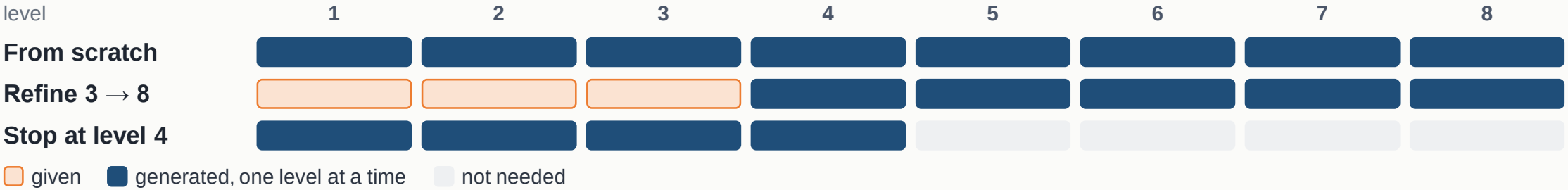


Learning p(x) = learning every refinement step.

$$p(\mathbf{x}) \underset{\text{(a)}}{=} p\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(8)}\right) \underset{\text{(b)}}{=} \prod_{\ell=1}^{8} p\left(\mathbf{x}^{(\ell)} \mid \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\ell-1)}\right)$$

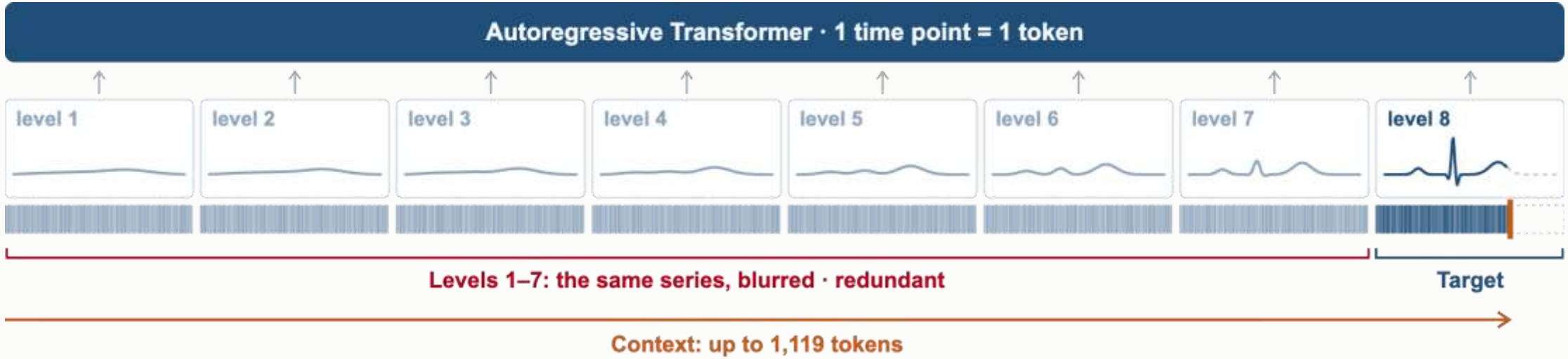
(a) coarse levels are computed from x

(b) chain rule, coarse to fine



→ ***Train on fine data only, and get every refinement step: start anywhere, stop anywhere.***

Modeling: In raw signal space, the ladder breaks



SEQUENCE LENGTH

1,120 values

TimeTok: 128 tokens

GENERATION STEPS

1,120

TimeTok: 8 blocks

CRPS, ECG5000

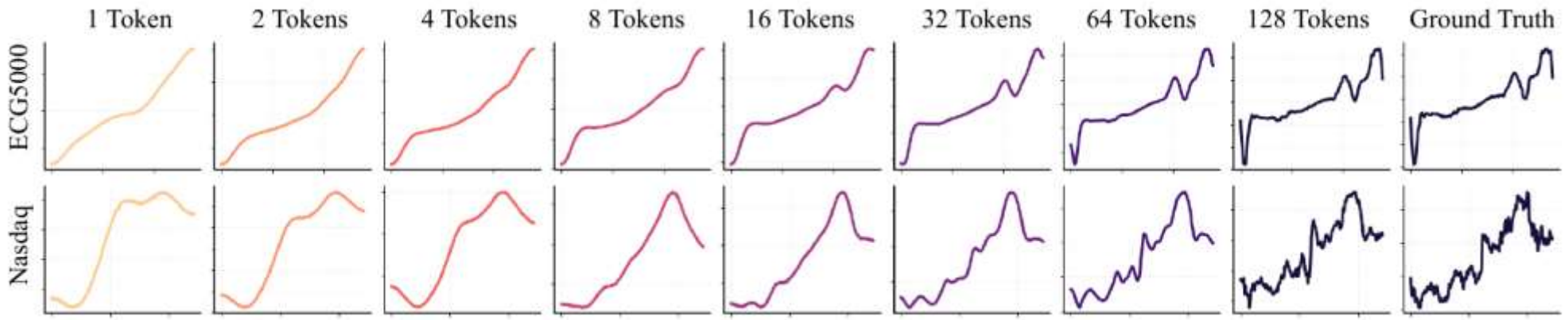
5.804

TimeTok: 1.587

The idea was right. The space was wrong.

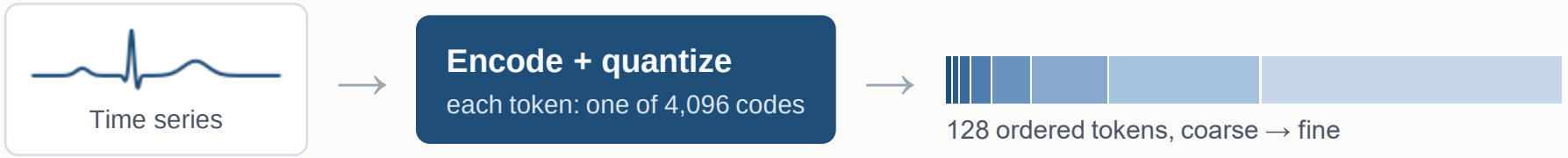
MODELING · THE KEY IDEA

What if reading more tokens meant seeing more detail?



TimeTok decoded from the first 1, 2, 4 … 128 tokens. Top row: ECG5000. Bottom row: Nasdaq.

Modeling: Token count sets the level of detail

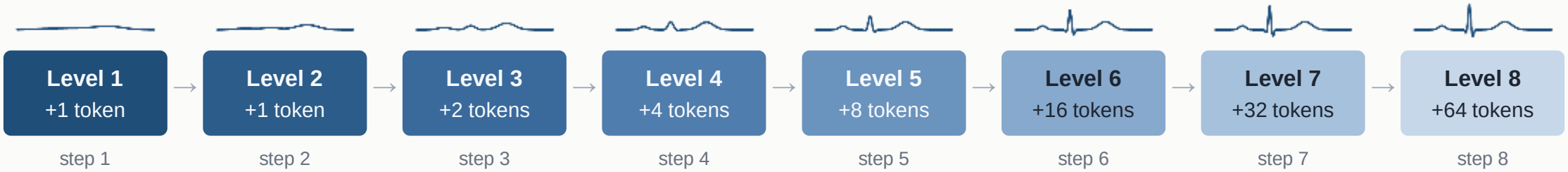




→ ***Early tokens carry the big picture; later tokens add finer detail.***

Modeling: Generate one level at a time

The same ladder as before, now over **blocks of tokens** instead of raw signals.



One step = one whole block

A VAR-style Transformer predicts each block given all coarser blocks, like next-scale prediction in images.

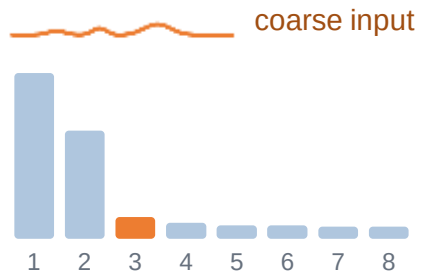
8 steps, not 1,120

The condition is a short, discrete prefix, not every coarser raw signal in full.

→ Same idea, different space: the ladder now works.

Modeling: Refining an input in three steps

1 · Read the input

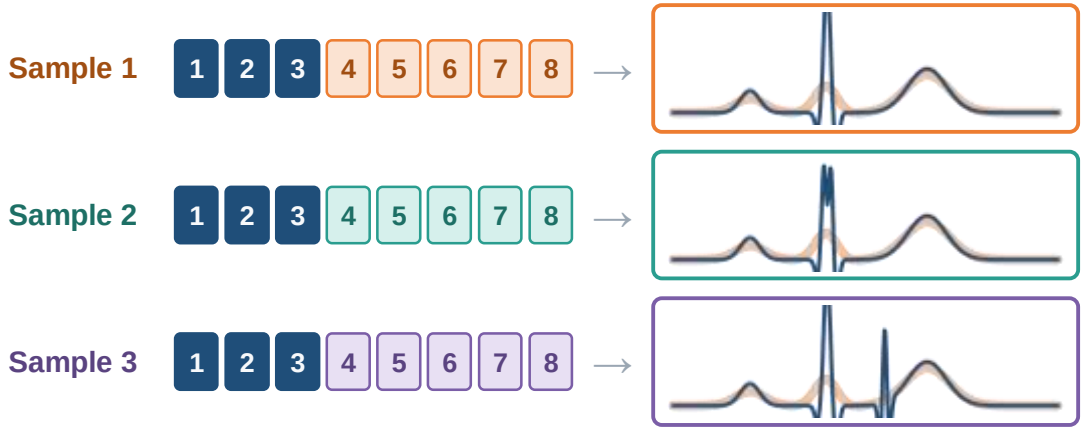


error vs. the input, by level · **detected: level 3**

Encode it, decode at every level, and pick the level where the error drops the most.

2 · Sample new blocks

3 · Decode



■ kept prefix: levels 1–3, same for all ☐ levels 4–8: sampled anew each time

Each sample of new tokens decodes to a different plausible series: many versions of one input.

No input? Start from an empty prefix: that is generation from scratch.

Modeling: Detail becomes a budget you choose

Level	Tokens	Size	What you get
1	1	12 bits	 Overall shape
3	4	48 bits	 Main swings
5	16	192 bits	 Most peaks
8	**128**	**1,536 bits**	 **Full detail**

Each token is one of 4,096 codes (12 bits).

Only the big picture?

Generate a few tokens. Fast and cheap.

Need every detail?

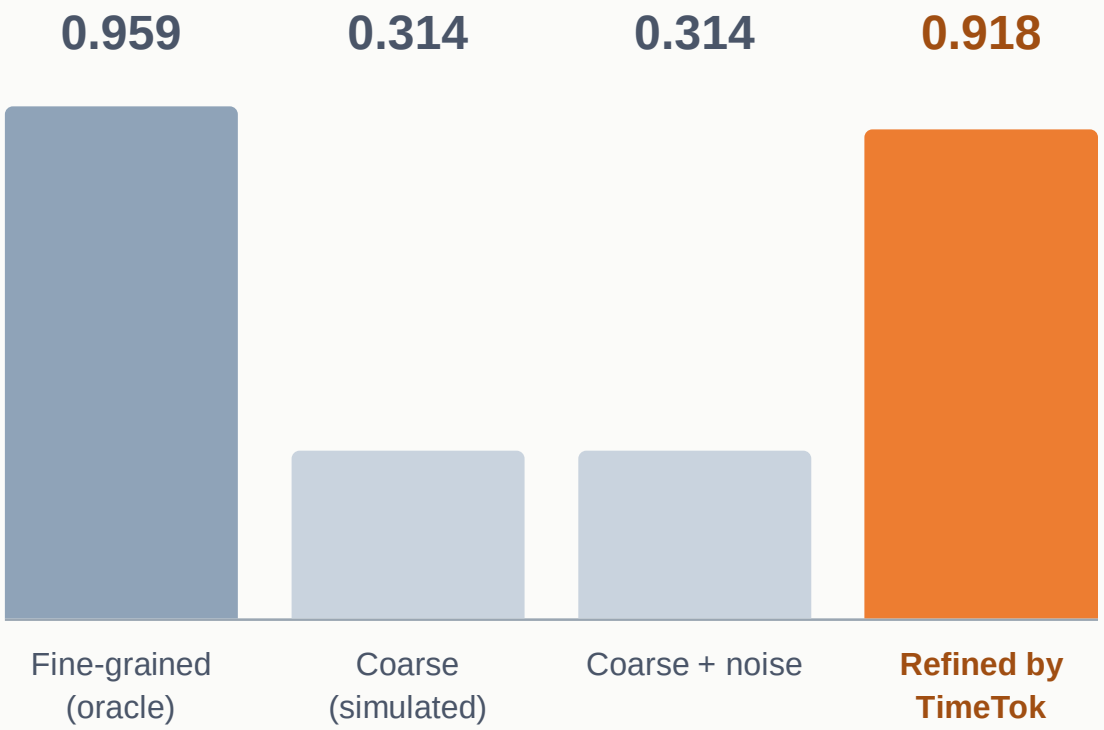
Generate all 128 tokens. More compute, more detail.

Existing models

Always the same number of tokens, whatever detail you need.

→ ***Pay for the detail you need, not a fixed number of tokens.***

Results: Back to the certified classifier



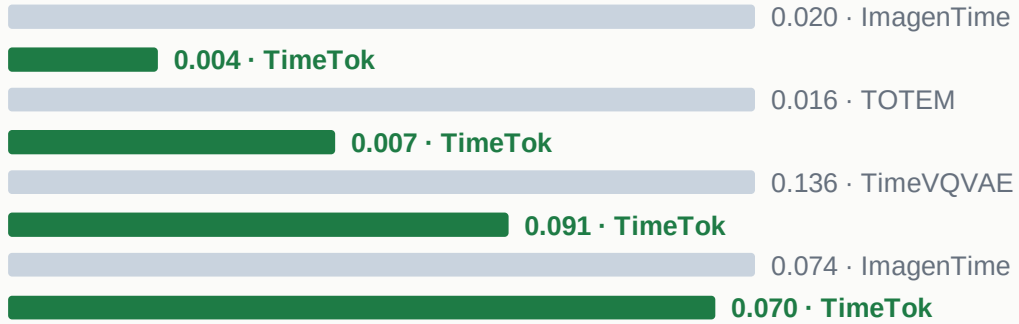
0.314 → 0.918

Refined signals keep what the unchanged classifier needs: accuracy recovers to 96% of the oracle.

ECG5000, classifier trained on fine-grained signals. Coarse inputs simulated by downsampling and linear interpolation (Table 5).

Results: Best quality, from scratch, at every level

FINEST LEVEL · STANDARD GENERATION



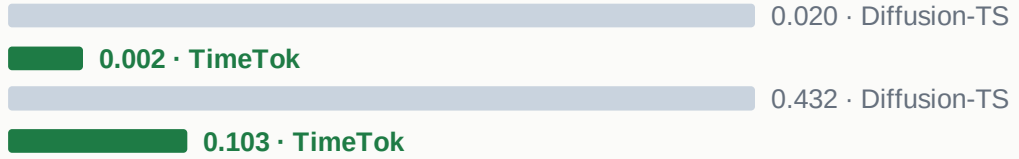
5.0× lower

2.3× lower

1.5× lower

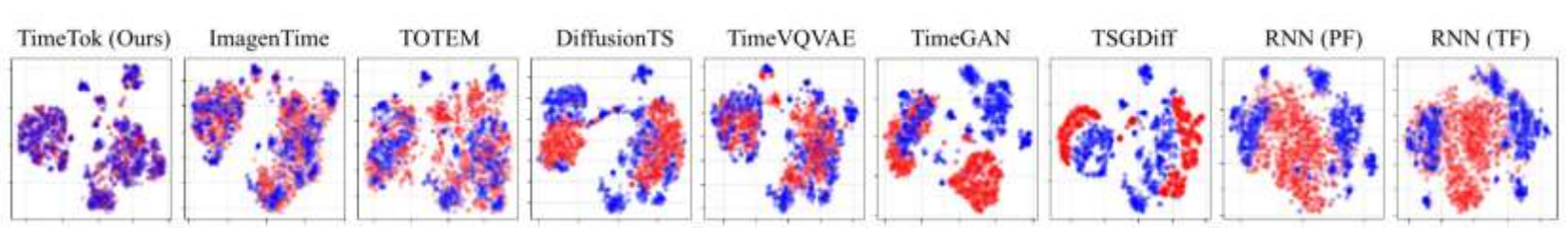
1.1× lower

EVERY LEVEL · AVERAGE OF 1–8



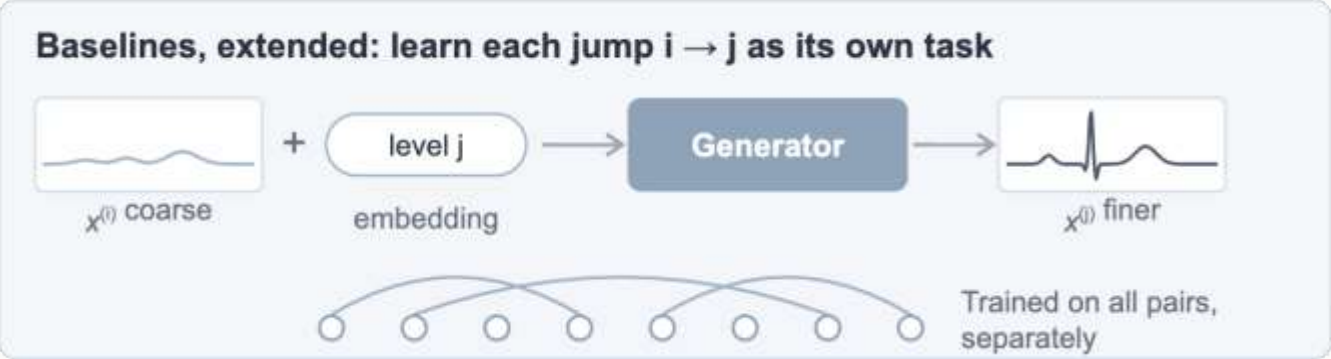
10× lower

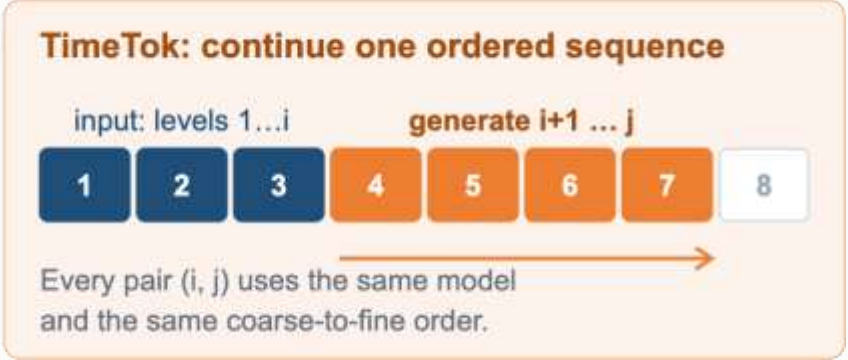
4.2× lower



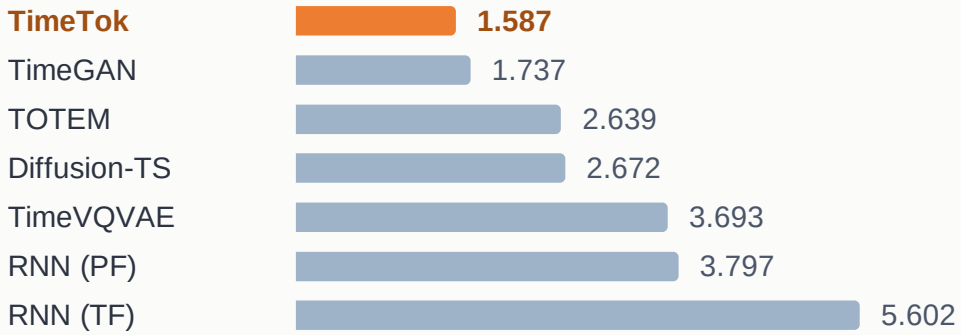
t-SNE, ECG5000: generated (red) vs. real (blue). TimeTok first.

Results: Refining coarse inputs

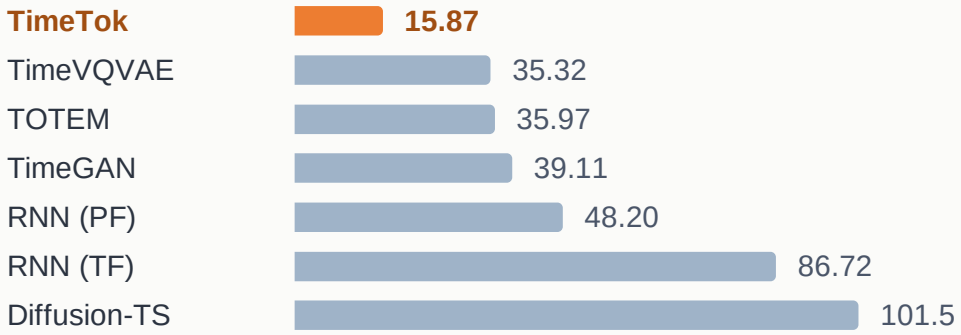




ECG5000 · CRPS-sum, lower is better



Nasdaq · CRPS-sum, lower is better



Stays faithful to the input: re-smoothed outputs match the input best (RMSE 0.010 vs. 0.011 on ECG5000, 0.096 vs. 0.196 on Nasdaq).

Averaged over all level pairs i → j, 5 samples per input (Table 3). Baseline extensions: Appendix A.7.

Results: One exact objective vs. separate jumps

TimeTok: one chain, from coarse to fine

$$\log p(\mathbf{x}) = \sum_{\ell=1}^{8} \log p\left(\mathbf{x}^{(\ell)} \mid \mathbf{x}^{(<\ell)}\right)$$

From scratch and refining are the same chain, started at a different level.

Baselines: many separate tasks in one network

$p_\theta\left(\mathbf{x}^{(j)} \mid \mathbf{x}^{(i)}, j\right)$ refine: one task per pair (i, j)

$p_\theta\left(\mathbf{x}^{(j)} \mid \varnothing, j\right)$ from scratch: another task

Nothing ties the tasks together: the network leans towards some of them.

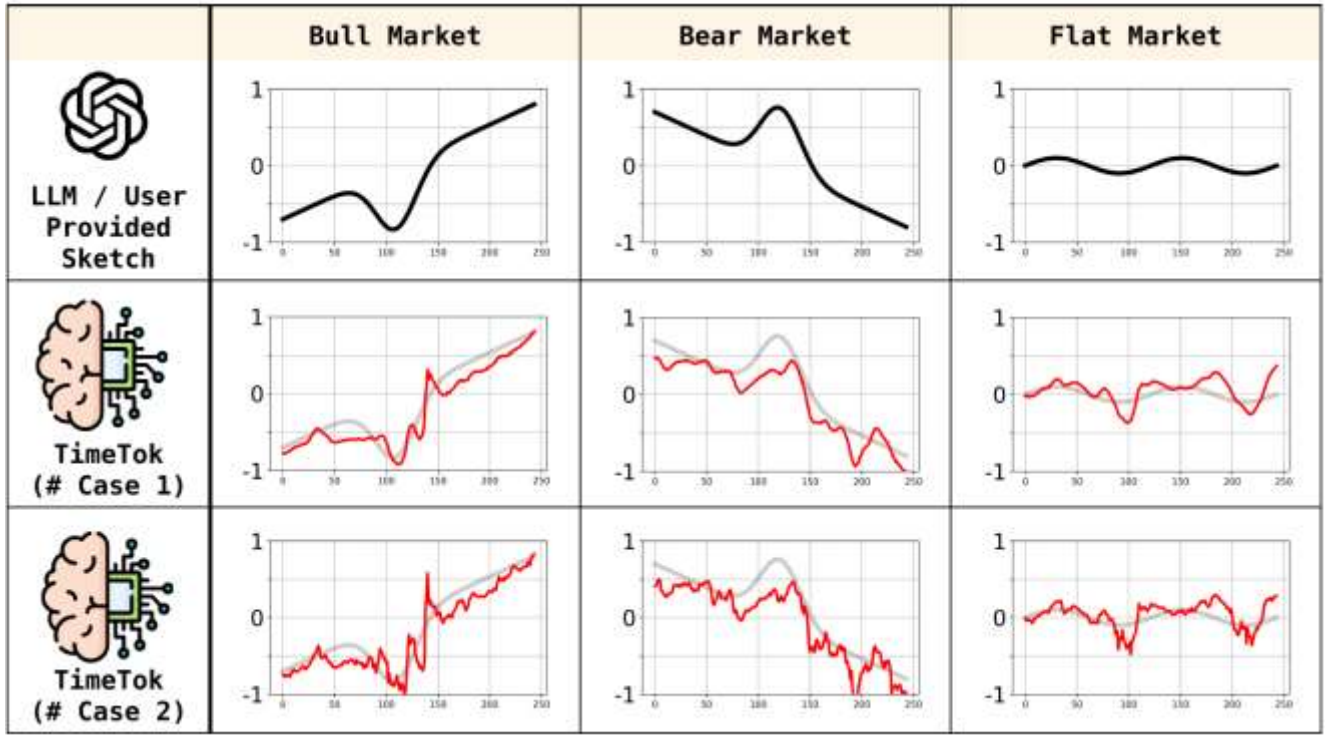


Each point is one model: refinement error averaged over all level pairs (Table 3) vs. from-scratch error averaged over levels 1–8 (Table 4). Both axes log scale.

Use Case · Finance: A view, drawn as a sketch

1. Describe a scenario in words: “a bull market with a sudden drop”.
2. An LLM, or a person, draws a rough curve.
3. TimeTok, trained on Nasdaq data, turns it into realistic price paths. Two samples shown.

The view-to-paths step from our motivation: people give the view, the data gives the days.



Vision: A detail layer for time-series AI

FROM PEOPLE

A view in words

An analyst or an LLM: “sharp drop, then slow recovery”

FROM DEVICES

Low-cost device data

Smartwatches, low-cost devices, IoT

FROM NOTHING

No input at all

Generate from scratch

→

TimeTok

One model. Any level of detail, chosen like a dial.



→

FINANCE

Agents and risk engines

Rehearse thousands of futures that follow the view

HEALTHCARE · INDUSTRY

Certified models, unchanged

Read refined signals, and flag when to measure again

PLANNING

What-if tools

Keep the trend, explore the details

→ People supply the big picture. TimeTok supplies the detail, at the level, and the cost, you choose.

Thank you.

Seokhyun Lee*, Jaeho Kim*, Changjun Oh, Mihaela van der Schaar, Changhee Lee

Korea University · University of Cambridge · *equal contribution

TimeTok · NeurIPS 2026

Paper

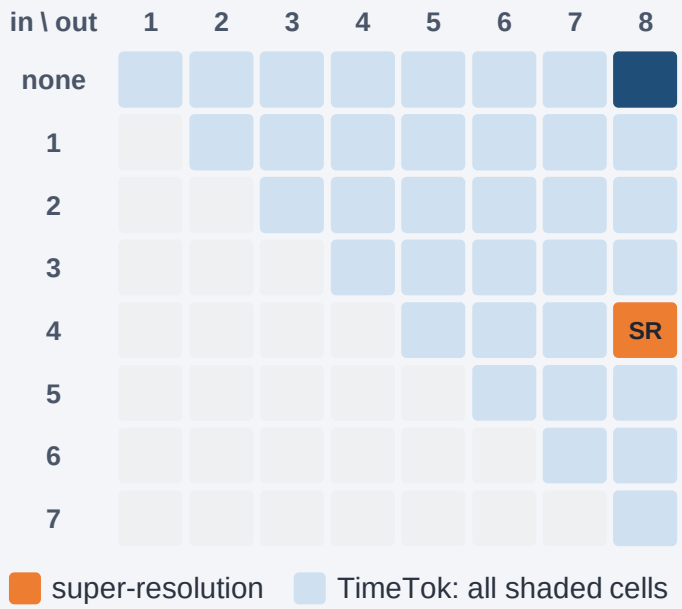
arXiv:2605.01418

LinkedIn

Seokhyun Lee

Backup: Isn’t this just super-resolution?

	Super-resolution (e.g. SRT)	TimeTok
What is learned	p(fine | coarse) for one pair	One joint distribution over all levels
Generate from scratch	No	Yes, at any level
Input level	Fixed, must be known	Any, detected automatically
Output level	Fixed	Any finer level, including in between
Different pairs agree	Not guaranteed (separate models)	By construction (one model)



Super-resolution is one cell of the GC-TSG table. TimeTok learns the whole table.

Not yet done: a head-to-head with a dedicated time-series super-resolution model on its own fixed pair.

Backup: Why not just retrain on coarse data?

RETRAIN WHEN YOU CAN

If you own the model, have labels at the coarse level and can afford re-validation, a classifier trained on coarse data is simpler and may be better. A coarse signal cannot carry more label information than it has.

REFINE WHEN YOU CAN'T

The model is certified or a vendor's. Labels exist only for fine data. Many models or human readers consume the signal. Or you need to know whether the missing detail matters.

Refinement does not create information. It lets fixed consumers use coarse data, and says how much is missing.

Backup: Why generate if you have real data?

REPLACING REAL DATA

Usually worse. Open Banking test: −32% accuracy with synthetic data only (FCA, 2024).

ADDING A LITTLE

Often helps. Amazon’s Chronos: ~10% synthetic series improved zero-shot forecasting. ImageNet: ResNet-50 76.4 → 78.2%.

DOING WHAT DATA CAN’T

Rehearse many futures from one history, follow a view history never saw, share without exposing records.

TimeTok’s case rests on the third column: conditions and futures that no dataset contains.

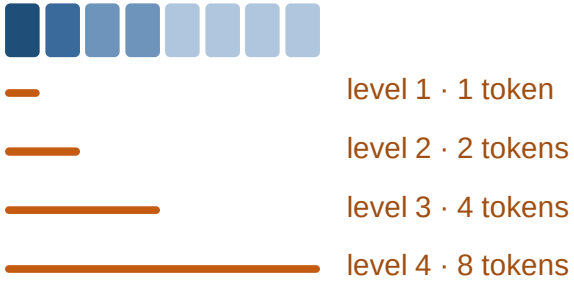
FCA Synthetic Data Expert Group (2024); Ansari et al., Chronos (TMLR 2024); Azizi et al., Synthetic data from diffusion models improves ImageNet classification (TMLR 2023). Gains shrink, then reverse, as the synthetic share grows.

Backup: Three design choices, three reasons

How do tokens get an order?

Training rebuilds a random level from only its token prefix. So early tokens must hold the big picture.

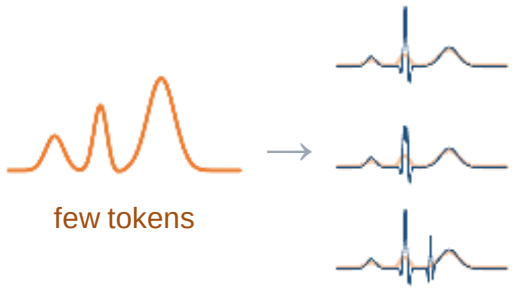
$$\hat{\mathbf{x}}^{(\ell)} = D\left(z_1, \ldots, z_{2^{\ell-1}}\right)$$



Coarse-to-fine flow-matching loss (Eq. 2)

Why a generative decoder?

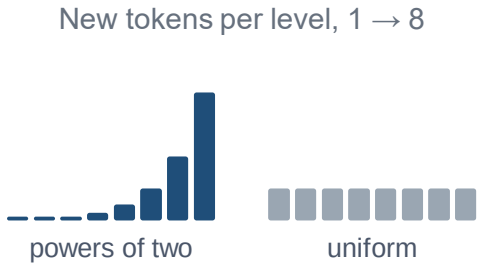
Few tokens fit many fine signals. The decoder samples one instead of averaging them into a blur.



Conditional flow matching

Why powers of two?

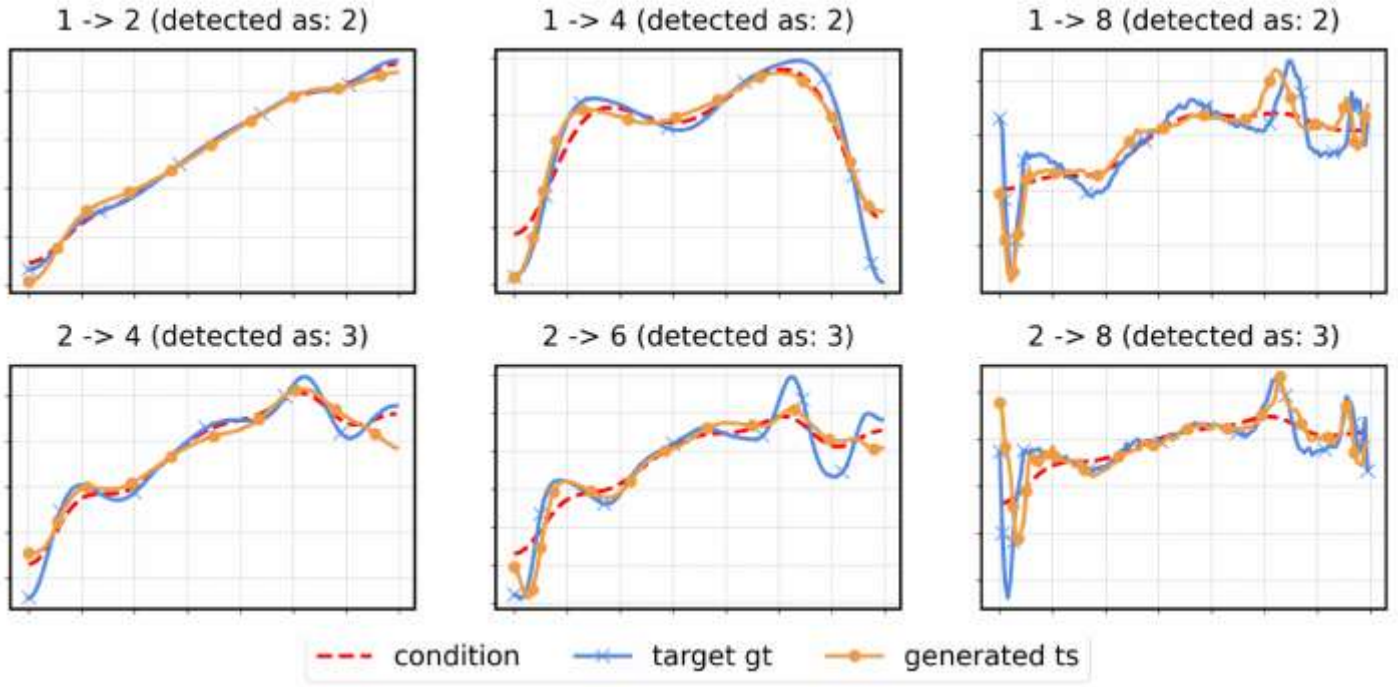
Coarse levels hold less information. Nearly the same loss as a uniform split, and coarse levels cost only a few tokens.



Ablation, Appendix D.3

→ ***The order is learned, the decoder samples, and coarse levels stay cheap.***

Backup: What refinement looks like

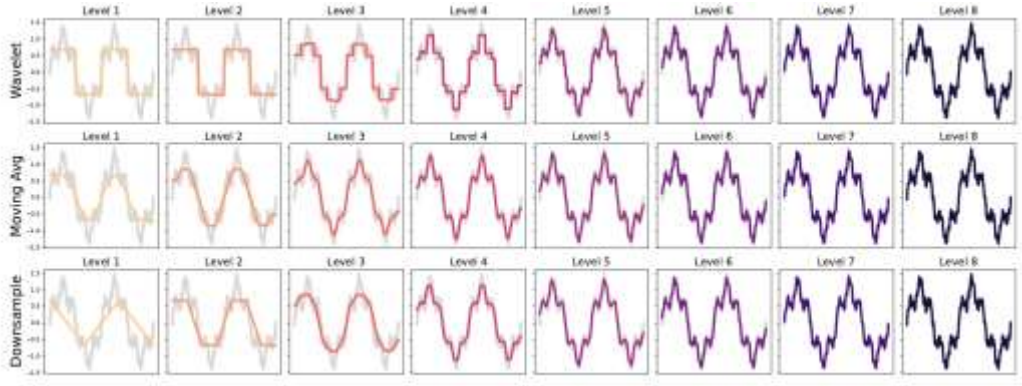


Each row: one coarse input, refined to three different target levels.

Red dashed: the input. **Orange:** TimeTok. **Blue:** the true signal at that level.

The title of each panel shows the level TimeTok detected on its own.

Backup: Robust to how inputs were coarsened



Test-time coarsening	ECG · TimeTok	ECG · best other	Nasdaq · TimeTok	Nasdaq · best other
Gaussian (as in training)	1.587	1.737	15.87	35.32
Wavelet	2.354	3.084	17.29	35.68
Moving average	2.080	2.702	17.13	35.75
Downsampling	2.126	2.686	16.50	34.63

CRPS, lower is better. TimeTok was trained with Gaussian smoothing only. Baselines: best of TimeGAN and TimeVQVAE (Table 17).

Backup: A real input comes alone

RAW-SPACE CHAIN

To refine a level-3 input, the chain needs levels 1 and 2 as well. A real input gives only level 3.



The missing coarser levels must be rebuilt first, or the chain cannot start.

TIMETOK

Encode the single input. Its token prefix already holds levels 1 to 3.



Keep that prefix and continue. No coarser history is needed.

In token space, one coarse input is enough to start the chain.

Backup: Why powers of two, not a uniform split?

1 · TOKENS NEEDED TO DESCRIBE A LEVEL

Level 1

1 token · powers of two

16 tokens · uniform

Level 3

4 tokens · powers of two

48 tokens · uniform

Level 5

16 tokens · powers of two

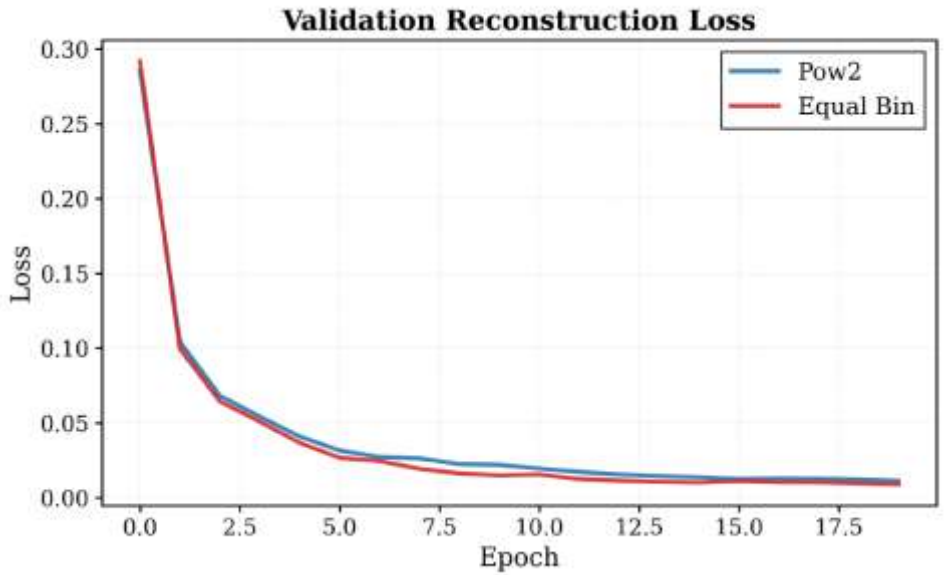
80 tokens · uniform

Level 8

128 tokens · powers of two

128 tokens · uniform

2 · RECONSTRUCTION QUALITY



Pow2 (blue) and uniform (red) reach almost the same loss.

Same quality, but coarse levels cost far fewer tokens: cheaper to generate, store and condition on.

Backup: One tokenizer across many datasets

TRAINING DATA

UTSD: 2.95 million series from many domains, each at its own granularity.

LEVEL MEASURED FROM THE SIGNAL (DFA EXPONENT)



← rough: fine level

smooth: coarse level →

Tokenizer trained on UTSD	ECG FID ↓	ECG ACC ↑	ECG F1 ↑	Nasdaq FID ↓	Nasdaq MSE ↓
TimeVQVAE	0.367	0.924	0.450	0.082	0.047
TOTEM	0.035	0.929	0.394	0.068	0.045
TimeTok, no DFA	0.042	0.944	0.583	0.111	0.044
TimeTok with DFA	0.010	0.951	0.668	0.046	0.042

Standard generation on ECG5000 and Nasdaq with a tokenizer trained on UTSD (Table 6). A first step: foundation-scale evaluation is still preliminary.

Why many markets? Agents learn by practice.

An AI agent that trades or hedges repeats one loop, on many markets.

WHAT THE AGENT NEEDS



Repeat tens of thousands of times

Calm markets, crashes, rebounds, slow declines

WHAT WE HAVE

The Nasdaq Composite, 2019 – 2026



One history

It happened only once

Right: Nasdaq Composite, monthly close, Jan 2019 – Sep 2026 (Yahoo Finance), log scale. Left: illustration.