SS2P2: A STABLE STATE-SPACE POINT PROCESS FOR LIMIT-ORDER-BOOK SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The limit order book simulation and prediction are important for financial market analysis and market making strategies. The existing State-Space Point Process (S2P2) model predicts the next LOB event accurately but suffers from instability when rolled out which leads to drift or explosion in the simulation. We decouple the output heads of the S2P2 model to create a Stable State-Space Point Process (SS2P2) model. The new model incorporates a softmin-bounded rate head and a soft-max mark head, ensuring stability and non-explosiveness in the point-process sense. Decoupling the heads gives us a hard closed-form intensity ceiling that bounds the event count and prevents explosion, while the soft-max mark head keeps the rate neutral at a fixed state so simulation stays stable. On Coinbase BTC, ETH, and SOL order flow against six baselines and three seeds each, we attain the best NLL and type accuracy on two of three assets (SOL NLL 0.27 vs. 0.38) and tie NHP on the third. We pass verified rate calibration on all nine checkpoints where the uncapped S2P2 fails on six, and lead five of twelve stylized facts in closed-loop roll-outs. The SS2P2 model provides a reliable and accurate framework for simulating limit order book events, making it a practical foundation for market-making world models and agent training.

1 INTRODUCTION

A limit order book (LOB) evolves through a continuous-time stream of order placements, cancellations, and trades. A generative model of this stream serves *prediction*, scoring the time and type of the next event given the history, and *simulation*, rolling forward on its own samples to reproduce the stylized facts of real order flow: calibrated event rates, burst–lull clustering, heavy tails (Cont, 2001). Scoring the observed stream tests fidelity to one realization; simulation tests whether the model has learned the process (Kumar et al., 2025), and is harder and more valuable for market-making world models, agent training, and counterfactual analysis (Li et al., 2025; Nagy et al., 2025).

Neural MTPPs (Du et al., 2016; Mei & Eisner, 2017; Shchur et al., 2021) excel at prediction; the continuous-time state-space model S2P2 (Chang et al., 2025) reports state-of-the-art held-out likelihood on standard event-sequence benchmarks, but only teacher-forced. Under our protocol, our factorized, uncapped S2P2 benchmark variant (Section 2) fails as a simulator. *Instability*: the self-excitation that wins one-step likelihood compounds in closed loop, where each sampled event raises the intensity and, with no rate bound, roll-outs drift hot. *Failed calibration*: the standard patch, rescaling the intensity to the real rate, fails; most S2P2 checkpoints, including every one on one asset, respond so sharply to the rescaling that our procedure finds no multiplier passing verification.

SS2P2 keeps our S2P2 backbone but changes its heads and training. Throughout, *stable* means the conditional intensity is globally bounded by an architectural constant, so finite-horizon event counts are stochastically dominated by homogeneous Poisson counts and roll-outs cannot explode; this implies neither stationarity, ergodicity, nor realism, which we assess empirically. Our contributions:

1. **Stable-by-construction heads**: a softmin-bounded rate head whose closed-form ceiling is an exact thinning bound, times a soft-max mark head that is rate-neutral at a fixed state.
2. **Carried-state training and simulation**: TBPTT on contiguous windows reduces the cold-start mismatch with roll-outs, which carry the state at $O(1)$ cost per arrival.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

3. **Verified post-hoc rate calibration**: a scalar root-find, for any decoder, that preserves mark ratios per state, keeps SS2P2's ceiling exact, and must pass a full-scale roll-out.

2 Background and Related Work

Marked temporal point processes. An MTPP models events $\mathcal{H}_T = \{(t_i, e_i)\}_{i=1}^{N_T}$ with times t_i and types $e_i \in \mathcal{E}$ via per-type conditional intensities $\lambda_e(t \mid \mathcal{H}_t)$. With total intensity $\lambda(t) = \sum_e \lambda_e(t)$, mark distribution $p(e \mid t) = \lambda_e(t)/\lambda(t)$, and compensator Λ, the log-likelihood is

$$\log \mathcal{L} = \sum_i \Big[\log \lambda(t_i) + \log p(e_i \mid t_i)\Big] - \Lambda(T), \qquad \Lambda(t) = \int_0^t \lambda(u \mid \mathcal{H}_u)\, du.$$

Ogata thinning (Lewis & Shedler, 1979; Ogata, 1981) requires a dominating rate $\bar{\lambda} \geq \lambda(t)$; the S2P2 authors avoid thinning because its dominating-rate and sampling-boundary hyperparameters "can exacerbate bias" (Chang et al., 2025), an objection SS2P2's exact ceiling removes.

The S2P2 backbone. S2P2 (Chang et al., 2025) stacks L latent-linear-Hawkes/SSM layers with diagonal state, ZOH-discretized between events, and post-residual LayerNorm (Ba et al., 2016),

$$x_{t'}^{(\ell)} = \bar{A}^{(\ell)} x_t^{(\ell)} + (\bar{A}^{(\ell)} - I)\tilde{B}^{(\ell)} h^{(\ell-1)} \big[+\tilde{E}^{(\ell)}\alpha \text{ at an event}\big],$$

with output $h_t := h^{(L)}(t) \in \mathbb{R}^H$. The published decoder reads $\lambda_e = \mathrm{softplus}(\beta_e a_e)/\beta_e$, $a = Wh_t + b$, with learnable sharpness β_e and unit asymptotic slope, so it has no explicit ceiling.

Our *factorized, uncapped* benchmark variant is likewise ceiling-free: a scalar total rate $\lambda = \mathrm{softplus}(w^\top h_t + b)$ on one half of h_t and a soft-max mark head on the other. Its backbone, our reimplementation, keeps the published block structure (Appendix A.7) but uses real diagonal states with log-spaced timescale initialization (vs. complex DPLR), a learned skip projection (vs. identity residual with diagonal feedthrough), dynamics gates conditioned at each interval's end (vs. start), and between-event queries holding inter-layer inputs at last-event values. SS2P2 keeps this backbone through h_t and replaces only the heads, but it alone trains with carried state (Section 4.1).

Related work. Neural MTPPs use recurrent (Du et al., 2016; Mei & Eisner, 2017), attention (Zhang et al., 2020; Zuo et al., 2020), jump-diffusion (Zhang et al., 2024), and state-space (Chang et al., 2025) encoders (survey: Shchur et al., 2021); Shi & Cartlidge (2022) evaluate LOB neural Hawkes models on prediction and simulation. Parametric order-flow Hawkes models (Hawkes, 1971; 2018; Jain et al., 2024a;b) and state-dependent extensions (Morariu-Patrichi & Pakkanen, 2022) have exact stationarity certificates but limited expressiveness; SS2P2 keeps neural expressiveness and restores non-explosiveness by a closed-form rate bound. Return-level work (Huang et al., 2024; Ang et al., 2026; Soun et al., 2026) concerns bounded, stationary returns without intensity feedback. Event-level generators span GANs (Li et al., 2020; Coletta et al., 2021; 2022), autoregressive SSMs (Nagy et al., 2023), diffusion-guided agents (Huang et al., 2026), foundation-scale transformers (Li et al., 2025; Kawawa-Beaudan et al., 2026), and agent-based simulators (Byrd et al., 2020; Frey et al., 2023); the closest in ambition, MarS, tokenizes inter-order intervals, leaving no conditional intensity or rate bound (Appendix A.8). Simulators are scored on stylized facts (Cont, 2001; Vyetrenko et al., 2020) and LOB-Bench's KS/Wasserstein suites, whose authors report error accumulation over long roll-outs (Nagy et al., 2025); Kumar et al. (2025) argue for checking calibration first, separately. We hold one likelihood-based model to prediction and closed-loop realism and, unlike these simulators, verify its calibrated mean rate by a full-scale roll-out.

3 Methods

We reconstruct typed events from Coinbase Level-2 order-book records and trades, fit a simulator by likelihood, and sample it in closed loop (Figure 1).

3.1 Data collection

We obtain full-depth Level-2 order-book records and trades for the Coinbase BTC-USDT, ETH-USDT, and SOL-USDT spot markets for 1–7 January 2026 from Kaiko (2025), one file per asset

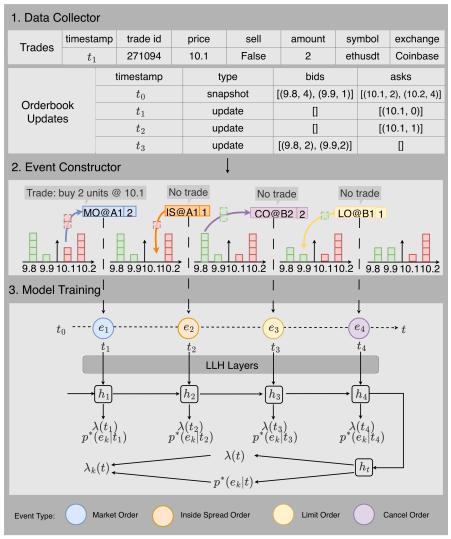

Figure 1: Pipeline: (1) trades and Level-2 book updates; (2) atomic typed events; (3) backbone state h_t decoded into a bounded rate $\lambda(t)$ and marks $p(e \mid t)$. We read event-time intensities at the left limit, before the event's own update.

and UTC day. A book record, stamped to the microsecond, is a snapshot of the whole book or an update that sets the aggregate resting size at listed prices (size zero removes the level); a trade carries a price, a size, an aggressor side, and a millisecond timestamp.

3.2 Event Construction

Per asset and day, we replay the book records on a full-depth book (reset by the day's first record and every snapshot), assign each record at t_j the trades in $(t_{j-1}, t_j]$, stamp its events with t_j, and keep per-price volume changes within the post-update top $K = 10$ levels per side. A type $e = (c, s, \delta) \in \mathcal{E}$ has a class $c \in \{\text{LO}, \text{CO}, \text{MO}, \text{IS}\}$, side $s \in \{b, a\}$, and level δ, assigned in three steps (Figure 2). (i) Each trade is a *market order* (MO) at $\delta = 1$ on the side it consumed (identical trades in an interval count once), netted against any decrease at its price on that side. (ii) A remaining decrease is a *cancel order* (CO). (iii) An increase strictly inside the previous spread is an *inside-spread order* (IS), with δ its improvement in ticks of 0.01 over the previous best quote on its side; any other increase is a *limit order* (LO). For LO and CO, $\delta \in \{1, \dots, K\}$ is the price's post-update rank ($\delta = 1$ best). Changes that empty a level (no rank), snapshot records, and records repeating the previous timestamp yield no book event.

Each book record with events is one arrival at t_i: the backbone input averages its types' embeddings, the mark term scores its first type e_i in vocabulary order (LO, CO, MO, IS; bid before ask; then level), and simulated arrivals carry one type. At load time we drop the type of IS improvements deeper than K ticks, so $|\mathcal{E}| = 6K + 2 = 62$; a record with only such events stays as an untyped arrival that advances the state and adds compensator mass but has no mark. Rates, windows, and test-event counts count arrivals. We split each day's arrivals chronologically 70/15/15% into training, validation, and test, discarding the first 1025 (one window plus its target) of validation and test.

3.3 SS2P2: A Bounded Rate and Rate-Neutral Marks

Both heads read the layer-normalized output $h_t \in \mathbb{R}^H$ ($H = 64$, two layers) of the unchanged backbone of Section 2, following the ground-intensity factorization of set-valued MTPPs (Chang et al., 2024) but with a categorical mark.

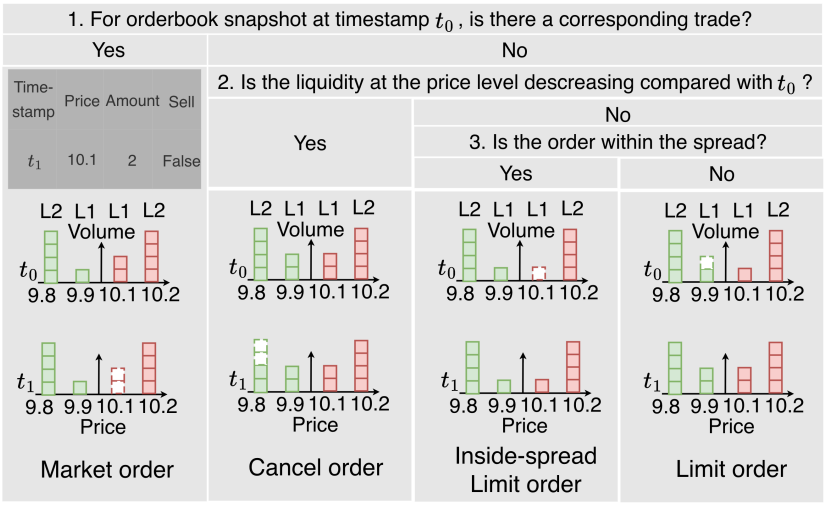

Figure 2: Event construction: we classify each price-level change between consecutive Level-2 book states ($t_0 \to t_1$) into the four order classes with three questions, given that interval's trades.

Softmin-bounded rate head. A gate maps h_t to a bounded g, and a smooth minimum (softmin) with fixed $z_{\max} = 6$ caps a linear readout of g:

$$o = \sigma(W_o h_t + b_o) \in (0,1)^H, \qquad g = o \odot \tanh(h_t) \in (-1,1)^H, \qquad z_{\text{raw}} = w^\top g + b,$$
$$z = z_{\max} - \text{softplus}\big(z_{\max} - z_{\text{raw}}\big), \qquad \lambda(t) = \varsigma\, \text{softplus}(z).$$

We initialize the learned scale $\varsigma > 0$ at the training-split rate over $\ln 2$, so with $b = 0$ and small initial w ($z_{\text{raw}} \approx 0$) the rate starts near that rate and the ceiling at $\text{softplus}(6)/\ln 2 \approx 8.7$ times it.

- **Hard ceiling.** As $\text{softplus} > 0$, $z < z_{\max}$, so $\lambda(t) \le \bar{\lambda} := \varsigma\, \text{softplus}(z_{\max})$ for every state and (W_o, b_o, w, b), by construction (unlike the soft no-arbitrage penalties of volatility smoothing (Wiedemann et al., 2025)), giving Poisson domination and exact thinning (Proposition 1, Appendix A) but not stationarity, ergodicity, or realism.
- **Bounded input.** As $g \in (-1,1)^H$, $|z_{\text{raw}} - b| < \|w\|_1$ bounds the rate for fixed parameters (Proposition 2); the cap adds a user-chosen ceiling on z, independent of w and b.
- **Vanishing gain.** $\partial z / \partial z_{\text{raw}} = \sigma(z_{\max} - z_{\text{raw}})$ is below one, $1/2$ at the cap, and decays exponentially beyond it (reachable only if $b + \|w\|_1 > z_{\max}$).
- **One-sided cap.** Well below the cap the head reduces to $\varsigma\, \text{softplus}(z_{\text{raw}})$, passing quiet-period gradients nearly unattenuated; bounded g also leaves a strictly positive, parameter-dependent floor (Proposition 2).

Rate-neutral mark head. A one-hidden-layer MLP of width H on the same h_t gives $p(e \mid t) = \text{softmax}(\text{MLP}(h_t))_e$ and $\lambda_e = \lambda\, p(e \mid t)$, so $\sum_e \lambda_e = \lambda$ for any mark logits: a larger mark network, or one with exogenous inputs such as an agent's quotes, changes neither the total rate at a given state nor the ceiling. In closed loop, marks still steer future states and can shift the realized mean rate (possibly requiring recalibration). Only the timing terms ($\log \lambda$, compensator) train the rate head, only the mark term the mark head, and both the shared backbone.

3.4 Training, simulation, and rate calibration

Sampling and rescaling the simulated intensity by a constant κ to the validation-split mean rate rely on three properties of the SS2P2 rate head that the S2P2 rate lacks (Appendix A.10):

(P1) the ceiling $\bar{\lambda}$, which holds at every state and depends on no weight but ς (Proposition 1);
(P2) a rate linear in ς, which the mark head does not use;
(P3) a gain $\partial\lambda / \partial z_{\text{raw}} = \varsigma\, \sigma(z)\, \sigma(z_{\max} - z_{\text{raw}})$ that vanishes as $\lambda \to \bar{\lambda}$.

Likelihood. We minimize the negative log-likelihood of Section 2, with a categorical mark term, over windows of $W = 1024$ arrivals plus the next one, scoring arrivals at left limits; order sizes and book state are not inputs. The compensator uses one evaluation per gap at its right end, $\sum_i \lambda(t_i^-)(t_i - t_{i-1})$, which undercharges gaps where λ decays: were only the level c of $c\,g(t)$ free, the fitted level would be $\int g\,dt \,/\, \sum_i g(t_i^-)(t_i - t_{i-1})$ times the exact one, above one when g decays in every gap, inflating the trained rate. The next-arrival term, sampler, and evaluation quadrature hold each upper layer's input at the lower layer's output just before the last event (Section 2); scoring updates it at each gap's end, so event-time intensities differ, increasingly with gap length.

Stateful (TBPTT) training. Cold-starting every window scores early arrivals without history and never warms slow modes as carried roll-outs do, so we train SS2P2 with truncated backpropagation through time (Williams & Peng, 1990): 64 contiguous lanes of non-overlapping windows, one window per lane per step, each starting from its predecessor's detached final state, resetting at lane starts each epoch and at day boundaries. Only gradients are truncated (carried states use earlier weights), at unchanged per-step cost.

Optimization. We train for 12 epochs with AdamW, keep the best-validation checkpoint (Appendix A.5), and run the backbone recursion as a sequential loop or an equivalent Hillis–Steele parallel scan (Hillis & Steele, 1986).

Carried-state simulation. A roll-out encodes one real W-arrival window from the initial state, then advances the state at $O(1)$ cost per simulated arrival, algebraically equal to re-encoding from the warm-up's start (Appendix A.9), for 600 s of simulated time or $40{,}000$ arrivals, whichever comes first.

Exact thinning (P1). We sample SS2P2 by Ogata thinning: candidates arrive at the constant rate $\kappa\bar{\lambda}$ and are accepted with probability $\lambda(t)/\bar{\lambda}$. By (P1), each step is exact for the intensity it queries, with no per-event bound search and $O(1)$ cost per proposal; the one exception is that we force-accept the 200th consecutive proposal, capping gaps near $200/(\kappa\bar{\lambda})$. The weight-dependent bound of S2P2 can far exceed realized rates, making thinning slow; an estimated bound $\hat{M}$ can be exceeded, and thinning then samples $\min(\lambda, \hat{M})$, drawing gaps that are too long. Models without a closed-form ceiling therefore invert the compensator, solving $\Lambda(\tau) = E \sim \mathrm{Exp}(1)$ on a log-spaced grid (exact only up to the grid).

Rate calibration (P2). The ceiling bounds but does not set a roll-out's mean rate, which the compensator approximation inflates, so we scale each checkpoint's simulated intensity by a constant κ chosen to match its validation-split rate r^*. By (P2), $\kappa\lambda$ is the same head with $\varsigma \to \kappa\varsigma$: still an SS2P2 model, with exact ceiling $\kappa\bar{\lambda}$ and unchanged $p(e \mid t)$ at every state. For S2P2, $\kappa\,\mathrm{softplus}(\cdot)$ leaves the model family; for S2P2_{pub}, calibration is equivalent to rescaling each type's readout by κ and sharpness by $1/\kappa$. Any decoder keeps λ_e/λ at a fixed history, but not its marginal mark frequencies. One constant fixes only the mean rate, not the intensity's shape.

Why calibration succeeds for SS2P2 (P1, P3). Let $r(\kappa)$ be the expected closed-loop mean rate. Propositions 1 and 2 give $\kappa\underline{\lambda} \le r(\kappa) \le \kappa\bar{\lambda}$ (floor $\underline{\lambda} > 0$), so every solution of $r(\kappa) = r^*$ lies in $[r^*/\bar{\lambda},\ r^*/\underline{\lambda}]$, and quartering κ from 1 brings $r(\kappa)$ below r^* within $\lceil \log_4(\bar{\lambda}/r^*) \rceil$ steps. The difficulty is sensitivity: for a linear Hawkes process with baseline μ and branching ratio n, $r(\kappa) = \kappa\mu/(1 - \kappa n)$ for $\kappa n < 1$, and $d\log r / d\log\kappa = 1/(1 - \kappa n)$ diverges as $\kappa n \to 1$, where small changes in κ or the sampling seed move the rate greatly. The SS2P2 head damps this feedback: the ceiling caps any runaway at $\kappa\bar{\lambda}$ (P1), and near it the vanishing gain (P3) makes each further event raise the rate less, whereas the slope of the S2P2 softplus tends to one. This is a mechanism, not a proof (the continuity and monotonicity of r are unproven); Section 4.4 tests it.

Calibration protocol. From $\kappa = 1$ we bracket r^* by factors of 4, bisect in $\log\kappa$ until a probe lands within 5%, and verify with a full-scale validation roll-out within 15%, recalibrating once after a miss; a checkpoint fails if the search does not converge or the second verification misses. Probes and verification use the inversion sampler; thinning reuses the verified κ without re-verification (Appendix A.2).

	BTC			ETH			SOL		
model	NLL↓	ACC↑	MAE↓	NLL↓	ACC↑	MAE↓	NLL↓	ACC↑	MAE↓
NHP	**-1.07±0.01**	**0.449±0.003**	**0.029±0.0001**	-0.81±0.03	0.341±0.006	0.030±0.0003	0.53±0.01	0.184±0.004	0.052±0.001
LSTM	-0.44±0.40	0.411±0.015	0.032±0.002	0.15±0.04	0.288±0.003	0.034±0.002	1.34±0.25	0.141±0.005	0.057±0.004
SAHP †	-0.93±0.01	0.412±0.001	0.034±0.005	-0.60±0.01	0.297±0.004	0.038±0.006	0.67±0.02	0.155±0.003	0.060±0.002
PCT-LSTM	-0.97±0.03	0.380±0.009	0.029±0.0004	-0.84±0.05	0.272±0.014	**0.028±0.0002**	0.38±0.08	0.130±0.011	**0.048±0.0004**
S2P2	-0.39±0.25	0.439±0.002	2.3±5.9	-0.38±0.22	0.331±0.008	10.2±31.0	0.67±0.49	0.182±0.002	46.2±96.0
$S2P2_{pub}$	0.71±2.04	0.327±0.272	0.030±0.009	5.96±12.25	0.192±0.245	0.028±0.001	2.59±3.49	0.140±0.116	0.049±0.0002
SS2P2	-0.87±0.12	0.448±0.001	0.030±0.001	**-0.87±0.08**	**0.345±0.003**	0.031±0.001	**0.27±0.05**	**0.192±0.001**	0.050±0.001

Table 1: Prediction per genuine test event (mean ± 95% CI over three training seeds): negative log-likelihood (NLL; time + mark), next-type accuracy (ACC) at the observed event time, and expected-time MAE in seconds (60 s horizon).

4 EXPERIMENTS

We compare SS2P2 with six baselines on prediction, closed-loop realism, verified rate calibration, and efficiency.

4.1 EXPERIMENTAL SETUP

Datasets. Each asset's seven days give seven daily segments per split (Section 3.2); test segments cover roughly the last two to six hours of each day. Validation-split rates are 38.3, 47.9, and 24.0 events/s (BTC, ETH, SOL): production rates, not subsampled streams.

Baselines. The baselines (Appendix A.11) are a standard LSTM decoder; NHP (Mei & Eisner, 2017), the neural Hawkes continuous-time LSTM; SAHP (Zhang et al., 2020), a transformer adaptation (not a faithful reproduction) with a softplus intensity piecewise-constant between events; PCT-LSTM (Shi & Cartlidge, 2022), CT4LSTM-PPP without the market-state input that closed loop cannot supply (possibly costing accuracy); S2P2 (Chang et al., 2025), our factorized, uncapped variant of Section 2; and $S2P2_{pub}$, a port of the published configuration verified value-level against the official code. All models share the input embedding, the likelihood with its right-endpoint compensator, the optimizer, the windows, and the number of gradient steps (Section 3.4), so $S2P2_{pub}$ does not follow the published training recipe. Only SS2P2 trains with carried state; the baselines draw the windows in shuffled order and start each from the initial state. S2P2, LSTM, NHP, and SAHP factorize rate times mark like SS2P2, but each of their heads reads half of the hidden state, whereas SS2P2 feeds all of h_t to both heads.

Training and evaluation protocol. Each model trains per asset with three seeds. At these rates a 1024-arrival window spans only $\sim$20–40 s of market time, so long-memory structure leans entirely on carried state.

- **Prediction**: a streaming evaluator with carried state (one cold start per genuine segment) scores every genuine test event; NLL integrates the compensator by a 32-point log-spaced quadrature per gap (Appendix A).
- **Simulation**: at the fixed calibrated κ, three roll-out seeds per checkpoint each run 32 sequences of 600 s, warm-started from the first 32 test windows and scored against equal-duration bootstrap samples of real segments; the test-split rate is reported, not used as a gate. All models carry state except SAHP (sliding 1024-arrival window). Stylized facts use an order-flow proxy (signed counts of price-moving events per 1 s bucket), built identically for real and simulated streams.
- **Calibration**: every model except SAHP (†; Section 4.3) is calibrated to the validation-split rate and verified within 15% (Section 3.4), with the shared inversion sampler so outcomes are comparable; failing checkpoints are excluded and reported as results.
- **Statistics**: we average roll-out seeds within each checkpoint and report t-based 95% CIs over three training seeds, indicative only at this sample size.

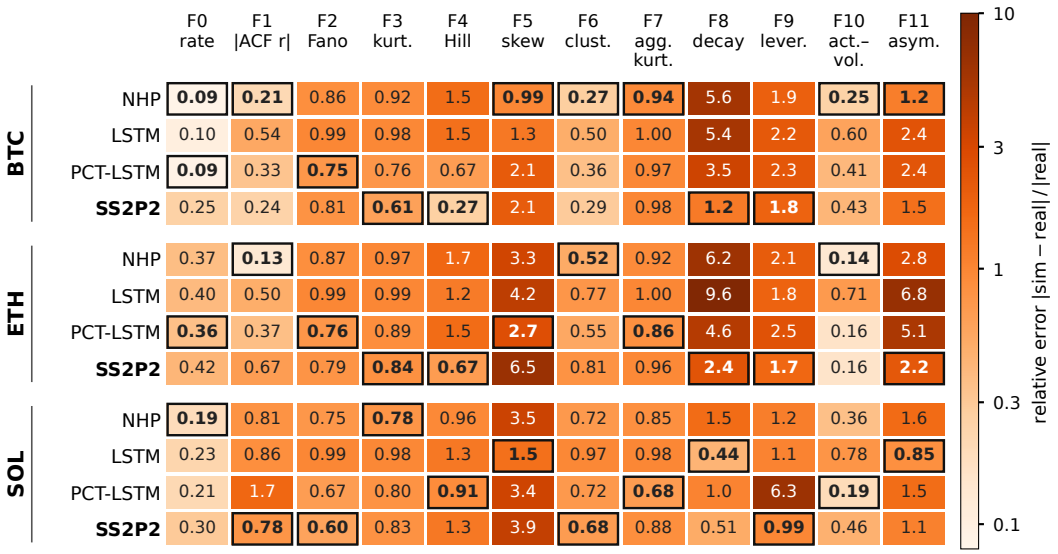

Figure 3: Stylized-fact relative errors on the order-flow proxy, averaged over roll-out seeds, then over three training seeds. F0 event rate; F1 |ACF| of proxy returns; F2 Fano factor (mean over six scales); F3 kurtosis; F4 Hill index; F5 skewness; F6 volatility clustering; F7 aggregated kurtosis; F8 $|r|$-ACF decay exponent; F9 leverage; F10 activity–volatility correlation; F11 timescale asymmetry. Log color scale (lighter is closer to real); boxes mark the lowest error per asset and fact (displayed-precision ties all boxed). Omitted: S2P2 and S2P2_{pub} (fewer than three calibrated checkpoints on some asset) and the uncalibrated SAHP.

4.2 Prediction Quality

SS2P2 leads NLL and type accuracy on two of three assets, and bounding the intensity did not reduce predictive performance in these experiments (Table 1): it wins both on SOL (0.27 vs. 0.38; 0.192 vs. 0.184) and ETH (-0.87 vs. -0.84; 0.345 vs. 0.341) and ties NHP on BTC marks (0.448 vs. 0.449) while conceding BTC likelihood. Its expected-time MAE (0.030–0.050 s) is about 1–3 ms behind the best model per asset (PCT-LSTM on ETH and SOL); the uncapped S2P2 is pathological (2.3, 10.2, 46.2 s), consistent with near-zero-intensity states whose expected-time integral hits the horizon clamp. S2P2_{pub} is erratic across seeds under this budget (ETH NLL spans 0.28–9.1): a protocol-matched reference rather than the published model's attainable accuracy. Appendix A.12 discusses the per-asset pattern.

4.3 Simulation Quality

We run each calibrated model in closed loop and score its roll-outs against real order flow on the twelve stylized facts of Figure 3. SAHP's free-running rate collapses to 5–16 ev/s against real segments at 20–35 ev/s, under half on every asset, so per protocol we did not calibrate it († in Table 1). S2P2 and S2P2_{pub} instead drift hot and fail verified calibration on 6 and 3 of their 9 checkpoints (Section 4.4); the three passing S2P2 checkpoints are erratic, with roll-out Fano relative errors of 0.27, 2.2, and 6.0.

Among the four models that pass calibration on every checkpoint, SS2P2 has the lowest relative error in 13 of the 36 asset–fact cells, more than any baseline (NHP 12, PCT-LSTM 9, LSTM 3; a tie counts for each tied model). It leads the tail and memory facts F3, F4, F8, and F9 on both BTC and ETH; on SOL it keeps F9 and leads the dispersion and clustering facts (F1, F2, F6) but trails on F4 (1.3 vs. PCT-LSTM's 0.91). This is not dominance: at three seeds the 95% intervals of the best and runner-up models overlap in all 36 cells, so the figure supports only broad competitiveness.

All models share two misses: simulated proxy returns are far lighter-tailed than real ones (F3 and F7 errors 0.61–1.00 on every asset), and F0 is floored by the validation–test rate gap (on ETH we calibrate to 47.9 ev/s, while the test segments run at 34.9). F2 separates the model families most

	verified calibration			cost (one GPU)		
model	BTC	ETH	SOL	step (s)	vs. loop	sim. ev/s
NHP	3/3	3/3	3/3	4.35	–	1253
LSTM	3/3	3/3	3/3	0.019	–	2083
SAHP †	*not calibrated*			0.059	–	583
PCT-LSTM	3/3	3/3	3/3	0.79	–	1919
S2P2	**2/3**	**1/3**	**0/3**	0.055	64×	1163
$S2P2_{pub}$	**2/3**	3/3	**1/3**	–	–	–
SS2P2	3/3	3/3	3/3	0.059	60×	879 / 353

Table 2: Verified calibration (checkpoints passing, of three seeds per asset; bold: failures) and cost on one GPU: training step on 64×1024 events (scan for the S2P2 family, speedup over its loop) and simulated events per wall-second (8×60 s roll-outs; SS2P2: inversion / thinning). † Not calibrated (protocol); $S2P2_{pub}$ was not timed.

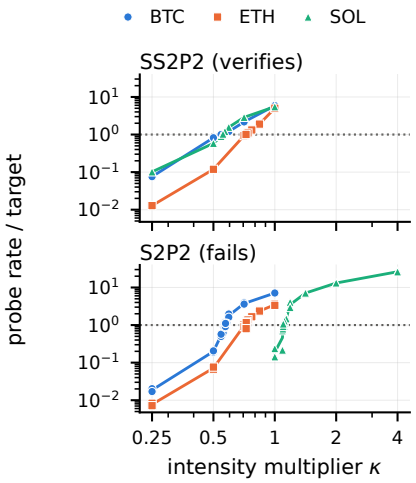

Figure 4: Seed-1 probe ladders: probe rate/target vs. κ (dotted: target). Failing S2P2 checkpoints respond near-discontinuously, worst on SOL.

sharply: the LSTM pins at ≈ 0.99 on every asset, a near-Poisson stream at the trivially stable end of the trade-off, while real Fano rises from $\sim$40–70 at 1 s to $\sim$340–1400 at 50 s. NHP leads the short-lag facts on BTC and ETH (F1, F6, F10) but misses decay there (F8 errors 5.6 and 6.2); PCT-LSTM is closest on F2 for BTC and ETH and on F7 for ETH and SOL, yet has the weakest type accuracy of the four. Prediction and simulation quality are thus separate axes: no baseline wins both, and SS2P2 is the only model we tested that tracks burst structure while predicting at the top of the table.

4.4 Calibratability

SS2P2 verifies on *all nine* checkpoints and S2P2 on 3 of 9, none on SOL (Table 2); every failure we report reproduces under deterministic seeds with bit-identical retry brackets.

The SS2P2 checkpoints behave as Section 3.4 predicts: at $\kappa = 1$ every probe ran at 2.6–5.9× its target, every checkpoint calibrated to $\kappa \in [0.545, 0.723]$, below one (the direction of the right-endpoint compensator's rate inflation), and every SS2P2 probe ladder we ran was monotone in κ.

Figure 4 shows the S2P2 failures. On the failing BTC checkpoint, moving κ by 8×10^{-4} carried the probe rate across the whole band ($34.8 \to 42.3$ ev/s, target 38.3); repeat probes at the same κ differed by tens of percent, and full-scale verification missed by 22%. On SOL the probe landed within tolerance (25.1 ev/s, target 24.0), yet the full-scale roll-out at the same κ ran at 58.2 ev/s. This sensitivity is consistent with the near-critical mechanism of Section 3.4. No multiplier we tried passed verification, while the SS2P2 response crossed the same targets smoothly on every asset. The faithful $S2P2_{pub}$ port free-runs 14–19× hot at $\kappa = 1$ and fails on 3 of 9 with the same signatures (a non-monotone BTC ladder; SOL probe–verification mismatches up to 44%), so the instability is not an artifact of our variant. Neither these failures nor the S2P2 expected-time tail (Section 4.2) appeared with SS2P2 in any run we report; this is an empirical regularity, not a theorem, and without separate ablations of the gate, the $\tanh$ bound, the cap, the ς initialization, and carried-state training we cannot attribute it to any one of them.

4.5 Efficiency Analysis

The parallel scan of Section 3.4 matches the S2P2-family's sequential loop up to floating-point order (maximum observed discrepancy 2×10^{-4}) and cuts the training step from 3.5 s to $\sim$0.06 s, 60× over the loop and 73× over NHP (Table 2; ETH checkpoints). We trained the reported ETH and SOL SS2P2 and S2P2 checkpoints with the loop and their BTC counterparts with the scan. Thinning

under the constant ceiling accepts 0.54 of proposals on ETH (~ 1.9 per event), so exactness costs $\sim 2.5\times$ vs. the shared inversion sampler, both far above real time (353 vs. 48 events/s).

5 Discussion and Conclusion

On three assets, SS2P2 has the best NLL and accuracy on SOL and ETH, ties the best BTC marks, and leads five of twelve stylized facts. It calibrated smoothly and verified on all nine checkpoints; the uncapped S2P2 decoders reproducibly fail this process-fidelity gate (Kumar et al., 2025), with near-critical rescaling sensitivity: S2P2 on 6 of 9 (all three on SOL) and the faithful port S2P2_{pub} on 3 of 9. The bound was nearly free: prediction held and exact thinning runs far above real time.

Limits: three streams from one crypto venue; we score order-flow statistics, not price paths; train-on-synthetic utility (Ang et al., 2026) is untested. The guarantee is boundedness, not realism: the LSTM is stable yet near-Poisson, SS2P2 leaves a Fano gap, NHP keeps the BTC likelihood lead, and component attribution, including carried-state training, awaits ablations. Agent-conditioned marks, which cannot violate the ceiling, and Chang et al. (2024)'s queries remain open.

AI use statement

In this work, we used generative AI tools (large language models) for five tasks: to aid and polish writing; for retrieval and discovery, such as finding related work; for research ideation and execution, such as auditing our implementation against the published S2P2 code and writing analysis and plotting scripts; to draft sections of the paper; and to prove and check mathematical claims, such as the bounds in Section 3.4 and Appendix A. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Reproducibility statement

Appendix A states the rate-bound propositions with proofs, the full calibration protocol with the per-checkpoint multipliers κ, the metric definitions (NLL quadrature, conditional type accuracy, expected-time MAE), the order-flow proxy behind the stylized-fact battery, the complete model and training configuration (optimizer, schedule, seeds, $z_{\max}$, ς initialization), and the dataset facts (assets, splits, event vocabulary, test-event counts). Every roll-out is seeded and deterministic, and scripts that regenerate every table and figure from the raw results accompany the submission.

References

Yihao Ang, Qiang Wang, Qiang Huang, Yifan Bao, Xinyu Xi, Anthony K. H. Tung, Chen Jin, and Zhiyong Huang. CTBench: Cryptocurrency Time Series Generation Benchmark. In *International Conference on Learning Representations*, 2026.

Anonymous. TFOW: Detecting Anomalous Order Flow in Crypto Limit Order Books. Companion manuscript, under review, 2026.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

David Byrd, Maria Hybinette, and Tucker Hybinette Balch. ABIDES: Towards High-Fidelity Multi-Agent Market Simulation. In *Proceedings of the 2020 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, pp. 11–22, 2020.

Yuxin Chang, Alex Boyd, and Padhraic Smyth. Probabilistic Modeling for Sequences of Sets in Continuous-Time. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 238 of *PMLR*, 2024.

Yuxin Chang, Alex Boyd, Cao Xiao, Taha Kass-Hout, Parminder Bhatia, Padhraic Smyth, and Andrew Warrington. Deep continuous-time state-space models for marked event sequences. In *Advances in Neural Information Processing Systems 38*, 2025. arXiv:2412.19634.

Andrea Coletta, Matteo Prata, Michele Conti, Emanuele Mercanti, Novella Bartolini, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. Towards Realistic Market Simulations: A Generative Adversarial Networks Approach. In *Proceedings of the Second ACM International Conference on AI in Finance*, 2021.

Andrea Coletta, Aymeric Moulin, Svitlana Vyetrenko, and Tucker Balch. Learning to simulate realistic limit order book markets from data as a world agent. In *Proceedings of the 3rd ACM International Conference on AI in Finance*, 2022.

Rama Cont. Empirical Properties of Asset Returns: Stylized Facts and Statistical Issues. *Quantitative Finance*, 1(2):223–236, 2001.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent Marked Temporal Point Processes: Embedding Event History to Vector. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1555–1564, 2016.

Sascha Frey, Kang Li, Peer Nagy, Silvia Sapora, Chris Lu, Stefan Zohren, Jakob Foerster, and Anisoara Calinescu. JAX-LOB: A GPU-accelerated limit order book simulator to unlock large-scale reinforcement learning for trading. In *Proceedings of the 4th ACM International Conference on AI in Finance*, 2023.

Alan G. Hawkes. Spectra of Some Self-Exciting and Mutually Exciting Point Processes. *Biometrika*, 58(1):83–90, 1971.

Alan G. Hawkes. Hawkes processes and their applications to finance: a review. *Quantitative Finance*, 18(2):193–198, 2 2018. ISSN 14697696. doi: 10.1080/14697688.2017.1403131. URL `/doi/pdf/10.1080/14697688.2017.1403131?download=true`.

W. Daniel Hillis and Guy L. Steele. Data parallel algorithms. *Communications of the ACM*, 29(12): 1170–1183, 1986.

Hongbin Huang, Minghua Chen, and Xiao Qiao. Generative Learning for Financial Time Series with Irregular and Scale-Invariant Patterns. In *International Conference on Learning Representations*, 2024.

Yu-Hao Huang, Chang Xu, Yang Liu, Weiqing Liu, Wu-Jun Li, and Jiang Bian. Controllable financial market generation with diffusion guided meta agent. In *Proceedings of the 40th AAAI Conference on Artificial Intelligence*, 2026.

Konark Jain, Nick Firoozye, Jonathan Kochems, and Philip Treleaven. Limit Order Book dynamics and order size modelling using Compound Hawkes Process. *Finance Research Letters*, 69:106157, 11 2024a. ISSN 1544-6123. doi: 10.1016/J.FRL.2024.106157. URL `https://www.sciencedirect.com/science/article/pii/S1544612324011863`.

Konark Jain, Nick Firoozye, Jonathan Kochems, and Philip Treleaven. Limit Order Book Simulations: A Review. *arXiv preprint arXiv:2402.17359*, 2024b.

Kaiko. Kaiko Cryptocurrency Market Data. `https://www.kaiko.com/`, 2025. Institutional digital-asset market data provider.

Maxime Kawawa-Beaudan, Srijan Sood, Kassiani Papasotiriou, Daniel Borrajo, and Manuela Veloso. TradeFM: A generative foundation model for trade-flow and market microstructure. *arXiv preprint arXiv:2602.23784*, 2026.

Harshit Kumar, Beomseok Kang, Biswadeep Chakraborty, and Saibal Mukhopadhyay. Has the Deep Neural Network Learned the Stochastic Process? An Evaluation Viewpoint. In *International Conference on Learning Representations*, 2025.

Peter A. W. Lewis and Gerald S. Shedler. Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413, 1979.

Junjie Li, Yang Liu, Weiqing Liu, Shikai Fang, Lewen Wang, Chang Xu, and Jiang Bian. MarS: A Financial Market Simulation Engine Powered by Generative Foundation Model. In *International Conference on Learning Representations*, 2025.

Junyi Li, Xintong Wang, Yaoyang Lin, Arunesh Sinha, and Michael P. Wellman. Generating realistic stock market order streams. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pp. 727–734, 2020.

Hongyuan Mei and Jason M Eisner. The Neural Hawkes Process: A Neurally Self-Modulating Multivariate Point Process. In I Guyon, U Von Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/6463c88460bd63bbe256e495c63aa40b-Paper.pdf.

Maxime Morariu-Patrichi and Mikko S. Pakkanen. State-dependent Hawkes processes and their application to limit order book modelling. *Quantitative Finance*, 22(3):563–583, 3 2022. ISSN 1469-7688. doi: 10.1080/14697688.2021.1983199.

Peer Nagy, Sascha Frey, Silvia Sapora, Kang Li, Anisoara Calinescu, Stefan Zohren, and Jakob Foerster. Generative AI for end-to-end limit order book modelling: A token-level autoregressive generative model of message flow using a deep state space network. In *Proceedings of the 4th ACM International Conference on AI in Finance*, 2023.

Peer Nagy, Sascha Frey, Kang Li, Bidipta Sarkar, Svitlana Vyetrenko, Stefan Zohren, Anisoara Calinescu, and Jakob Foerster. LOB-Bench: Benchmarking generative AI for finance – an application to limit order book data. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Yosihiko Ogata. On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31, 1981.

Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural Temporal Point Processes: A Review. *IJCAI International Joint Conference on Artificial Intelligence*, pp. 4585–4593, 4 2021. ISSN 10450823. doi: 10.24963/ijcai.2021/623. URL https://arxiv.org/pdf/2104.03528.

Zijian Shi and John Cartlidge. State Dependent Parallel Neural Hawkes Process for Limit Order Book Event Stream Prediction and Simulation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1607–1615, New York, NY, USA, 8 2022. ACM. ISBN 9781450393850. doi: 10.1145/3534678.3539462.

Yejun Soun, Hosung Lee, Suyoung Park, and U Kang. STABLE: Shift-Tolerant Allocation via Black-Litterman Using Conditional Diffusion Estimates. In *International Conference on Learning Representations*, 2026.

Svitlana Vyetrenko, David Byrd, Nick Petosa, Mahmoud Mahfouz, Danial Dervovic, Manuela Veloso, and Tucker Balch. Get Real: Realism Metrics for Robust Limit Order Book Market Simulations. In *Proceedings of the First ACM International Conference on AI in Finance*, 2020.

Ruben Wiedemann, Antoine Jacquier, and Lukas Gonon. Operator Deep Smoothing for Implied Volatility. In *International Conference on Learning Representations*, 2025.

Ronald J. Williams and Jing Peng. An efficient gradient-based algorithm for on-line training of recurrent network trajectories. *Neural Computation*, 2(4):490–501, 1990.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-Attentive Hawkes Process. In *International Conference on Machine Learning*, pp. 11183–11193, 2020.

Shuai Zhang, Chuan Zhou, Yang Liu, Peng Zhang, Xixun Lin, and Zhi-Ming Ma. Neural jump-diffusion temporal point processes. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

A TECHNICAL APPENDIX

A.1 THE ARCHITECTURAL RATE BOUND

Proposition 1 (global ceiling). For the head $g = o \odot \tanh(h_t)$, $z_{\text{raw}} = w^\top g + b$, $z = z_{\max} - \text{softplus}(z_{\max} - z_{\text{raw}})$, $\lambda = \varsigma\,\text{softplus}(z)$ with $\varsigma > 0$:

$$\lambda(t) \;\le\; \bar{\lambda} := \varsigma\,\text{softplus}(z_{\max})$$

for every h_t and every (W_o, b_o, w, b). *Proof.* $\text{softplus}(x) > 0$ for all x, so $z = z_{\max} - \text{softplus}(z_{\max} - z_{\text{raw}}) < z_{\max}$; softplus is increasing, so $\lambda < \varsigma\,\text{softplus}(z_{\max})$.

(i) The conditional intensity is globally bounded, so for any finite horizon T the compensator satisfies $\Lambda(T) \le \bar{\lambda}T < \infty$ and the process is non-explosive on $[0, T]$. (ii) The event count $N(T)$ is stochastically dominated by a homogeneous Poisson count with mean $\bar{\lambda}T$: the process can be constructed by thinning a homogeneous Poisson process of rate $\bar{\lambda}$, and thinning only removes points. (iii) $\bar{\lambda}$ is an exact dominating rate for Ogata thinning (Lewis & Shedler, 1979; Ogata, 1981). The bound establishes nothing beyond (i)–(iii); stationarity, ergodicity, and realism are empirical matters.

Proposition 2 (parameter-dependent floor and pre-cap bound). Since $g \in (-1, 1)^H$, for fixed trained parameters $b - \|w\|_1 < z_{\text{raw}} < b + \|w\|_1$. Hence, even without the cap, $\lambda \le \varsigma\,\text{softplus}(b + \|w\|_1)$ is a computable post-training ceiling; and with the cap,

$$\lambda \;\ge\; \varsigma\,\text{softplus}\big(z_{\max} - \text{softplus}(z_{\max} - b + \|w\|_1)\big) \;>\; 0,$$

a strictly positive, typically numerically negligible floor. The intensity therefore has zero infimum only in the limit of unbounded parameters, not for any fixed trained model. The cap's contribution over the pre-cap bound is that $z_{\max}$ is user-chosen and fixed in training, so $\bar{\lambda}$ depends on no trained weight but ς.

A.2 CALIBRATION PROTOCOL

Targets are validation-split rates (38.27, 47.92, 23.98 ev/s for BTC, ETH, SOL). A probe runs 8 sequences of 600 s, each warm-started from one of the first 8 validation windows, under one fixed seed. Starting from $\kappa = 1$, we multiply or divide κ by 4 until the target is bracketed, then bisect in $\log \kappa$, for at most ten steps, until a probe lands within 5% of the target. A full-scale validation roll-out (32 sequences from the first 32 validation windows, under a second fixed seed) must then land within 15%; if it misses, we recalibrate once with 32-sequence probes and verify again. A checkpoint fails calibration if the bracketing or bisection does not converge or the second verification misses; failing checkpoints are excluded and reported. Probes and verification use the compensator-inversion sampler; thinning roll-outs reuse the verified κ without re-verification. Probe seeds are deterministic; retried failures reproduce their bisection brackets bit-identically. We neither assume nor prove that the closed-loop rate is monotone in κ; it was monotone on every SS2P2 probe ladder we ran. Calibrated multipliers κ (seeds 1/2/3): SS2P2 BTC $0.545/0.569/0.648$, ETH $0.723/0.707/0.582$, SOL $0.557/0.595/0.621$; S2P2 (passing checkpoints only) BTC 0.534 (s2), 0.751 (s3), ETH 1.102 (s3); S2P2_{pub} (passing) BTC 0.608 (s1), 0.439 (s2), ETH $0.322/0.310/0.439$, SOL 0.534 (s2).

A.3 METRIC DEFINITIONS

NLL is per-event negative log-likelihood with the compensator integrated by a 32-point log-spaced quadrature per inter-event gap (horizon 60 s; beyond the grid the intensity is extended at its terminal value). **Type accuracy** is top-1 accuracy of $p(e \mid t^*)$ evaluated at the *observed* next-event time t^* (accuracy conditional on the observed time). **Expected-time MAE** computes $\hat{\mathbb{E}}[\Delta t]$ by survival quadrature on the same log-spaced grid, adds the tail term $S(T_{\max})/\lambda(T_{\max})$, clamps the result to $[10^{-6}, 2T_{\max}]$ with $T_{\max} = 60$ s, and reports the mean absolute error against the true gap. All prediction metrics are computed by a streaming evaluator with carried state, one cold start per genuine stream segment, which hard-fails if any test event goes unscored.

A.4 STYLIZED-FACT CONSTRUCTION (ORDER-FLOW PROXY)

The models emit event times and types only, so the battery operates on an order-flow proxy applied identically to real and simulated streams. Events are bucketed at 1 s. Per bucket, *activity* a_t is the to-

tal event count and the *proxy return* r_t is the signed count of price-moving events (unit tick): market orders and best-level inside-spread improvements, bid side positive, ask side negative. F0 is the event rate; F2 is the Fano factor $\mathrm{Var}/\mathrm{Mean}$ of bucket counts aggregated at scales $\{1, 2, 5, 10, 20, 50\}$ s, averaged over scales; F1, F3–F9, and F11 are standard stylized-fact statistics of r_t (autocorrelation, kurtosis, Hill index, skewness, clustering, aggregated kurtosis, power-law decay of $|r|$ autocorrelation, leverage, timescale asymmetry); F10 correlates a_t with $|r_t|$ (activity–volatility). None of these are statistics of asset price returns; we label them accordingly.

A.5 Model and training configuration

All models share the input embedding (channel embedding 64, time embedding 64, unused by the S2P2-family decoders, where elapsed time enters only through the backbone's decay), the likelihood, the optimizer, the windows, and the number of gradient steps; they differ in the decoder, and only SS2P2 trains with carried state (TBPTT, detached state passed between windows). Backbone/decoder hidden size 64, two layers. We train for 12 epochs with AdamW (learning rate 2×10^{-3}, weight decay 10^{-6}), gradient-norm clipping at 0.5, TF32 matrix multiplies, and no dropout, and we halve the learning rate after six epochs without validation improvement (ReduceLROnPlateau, factor 0.5, patience 5). Each step averages the window loss over 64 windows of 1024 arrivals (stride 1024). After every epoch we evaluate the loss on all validation windows, each started from the initial state, and keep the best checkpoint; we train three seeds $\{1, 2, 3\}$. We evaluate the backbone recursion either with a sequential per-event loop or with an equivalent Hillis–Steele parallel prefix scan (Hillis & Steele, 1986). SS2P2 rate head: $z_{\max} = 6$; ς softplus-parameterized and initialized from the training-split rate ($\varsigma_0 = \text{rate}/\ln 2$); gate bias and rate bias initialized to zero; rate weights Xavier-uniform (gain 0.5). Simulation: carried-state roll-outs, 32 sequences $\times$ 600 s, three roll-out seeds, thinning for SS2P2 (constant ceiling $\varsigma\,\mathrm{softplus}(z_{\max})$), compensator-inversion sampling for baselines. The S2P2$_{\text{pub}}$ port sizes are benchmark-matched (H=64, two layers, complex state dimension P=64 per layer); its architecture follows the official example configuration (`Int_Backward_LLH`, `relative_time`, post-norm, GELU) and was verified value-level against the official classes.

A.6 Dataset

Coinbase BTC-USD, ETH-USD, and SOL-USD trades and Level-2 book updates, seven days per asset, encoded into $|\mathcal{E}| = 62$ atomic types ($20 + 20 + 2 + 20$ over LO/CO/MO/IS $\times$ side $\times$ level). Genuine test events scored: 3,226,821 (BTC), 3,623,575 (ETH), 2,000,638 (SOL). Trades are attributed to book changes by inter-update timestamp interval and netted against liquidity changes at the traded price; residual decreases are cancellations. Multiple atomic events from one update share its timestamp and are ordered deterministically (MO, CO, LO, IS; bid before ask; then by level). Order sizes are retained in the dataset but unused by the models in this paper.

A.7 Backbone reimplementation and the uncapped benchmark variant

The published S2P2 decoder (Section 2) uses a learnable per-type sharpness β_e in $\lambda_e = \mathrm{softplus}(\beta_e a_e)/\beta_e$. Each β_e reshapes the curvature near zero, but the map is asymptotically linear with unit slope, so the head carries no explicit output ceiling. Our factorized, uncapped benchmark variant shares the published head's operative property, a softplus rate with no explicit ceiling, though its head is factorized rather than per-type. It is built on our real-valued reimplementation of the backbone, which keeps the published configuration's block structure (backward-ZOH input drive, input-dependent dynamics, post-norm GELU residual readout, event impulses, left-limit likelihood) and deviates in four respects: real diagonal states with log-spaced timescale initialization (vs. the complex DPLR spectrum); a learned skip projection (vs. identity residual with diagonal feedthrough); dynamics gates conditioned at each interval's end (vs. start); and between-event queries holding inter-layer inputs at last-event values. SS2P2 keeps everything up to and including h_t and replaces only the heads; it also differs in training, since only SS2P2 trains with carried state (Section 4.1).

A.8 Extended related work

Neural TPPs for event streams. Neural MTPPs learn history-dependent intensities with recurrent (Du et al., 2016; Mei & Eisner, 2017), attention-based (Zhang et al., 2020; Zuo et al., 2020), jump-diffusion (Zhang et al., 2024), and continuous-time state-space (Chang et al., 2025) encoders (survey: Shchur et al., 2021). Shi & Cartlidge (2022) specialize neural Hawkes models to LOB streams and evaluate both prediction and simulation; the state-space line, however, is evaluated teacher-forced only, with state-of-the-art held-out likelihood but no roll-outs (Chang et al., 2025).

Hawkes modeling of order flow. Parametric Hawkes processes are the classical LOB event model (Hawkes, 1971; 2018; Jain et al., 2024a;b), with state-dependent extensions (Morariu-Patrichi & Pakkanen, 2022): exact stationarity certificates, limited expressiveness. SS2P2 takes the complementary route: full neural expressiveness with a closed-form rate bound restoring non-explosiveness, the bound being a modeling decision rather than a thinning device.

Generative market simulators. Return-level generators synthesize daily or hourly return series, one shot from noise or segment by segment (Huang et al., 2024), and the crypto benchmark CTBench scores them by downstream trading utility rather than marginal similarity, with order-book data excluded by design (Ang et al., 2026). Returns are bounded and stationary, so these models never meet the intensity feedback that governs event-level simulation, though stylized-fact audits still find that off-the-shelf diffusion and GAN generators underestimate kurtosis (Soun et al., 2026). Event-level order-flow generators span GANs (Li et al., 2020; Coletta et al., 2021; 2022), token-level autoregressive state-space models (Nagy et al., 2023), diffusion-guided agents (Huang et al., 2026), and foundation-model-scale transformers (Li et al., 2025; Kawawa-Beaudan et al., 2026); agent-based simulators (Byrd et al., 2020; Frey et al., 2023) remain the RL workhorse. MarS, the closest in ambition, tokenizes inter-order intervals into 16 bins, so it has no conditional intensity and no notion of a rate bound. It validates roll-outs only over anchored horizons of 5–25 minutes re-seeded from real orders and reports no event-rate diagnostic, and its authors note that stylized facts alone do not certify simulation quality (Li et al., 2025).

Evaluating simulators. Order-flow generators are scored on distributional realism: stylized facts (Cont, 2001; Vyetrenko et al., 2020) and the KS/Wasserstein suites of LOB-Bench (Nagy et al., 2025), whose authors report error accumulation over long roll-outs. Kumar et al. (2025) make the general point for stochastic systems: any metric scored on a single observed path, likelihood included, tests fidelity to a realization rather than to the process, and because a proper score rewards sharpness, calibration should be checked first as a separate gate. Our verified mean-rate calibration and stylized-fact battery play that role for closed-loop event streams. SS2P2 thus differs from the simulators above on two counts: one likelihood-based model is held to *both* prediction and closed-loop realism, and the simulated mean rate is calibrated and checked by a full-scale verification roll-out, a closed-loop test none of them reports.

A.9 Carried-state simulation: why not re-encode

Carried-state roll-outs advance the backbone state with each simulated arrival at $O(1)$ cost. Re-encoding only the last W arrivals would cost $O(W)$ per arrival and let older history act only through them.

A.10 The S2P2 rate lacks (P1)–(P3)

The S2P2 softplus rate has none of the properties (P1)–(P3) of Section 3.4: its only bound, from the LayerNorm on h_t, grows with the trained weights; rescaling it by κ changes no single parameter; and its slope tends to one as the rate grows.

A.11 Baseline and protocol details

SAHP (Zhang et al., 2020) is a two-layer, four-head causal transformer encoder with a softplus intensity held piecewise-constant between events; the published decay parameterization is not

ported. **PCT-LSTM** (Shi & Cartlidge, 2022) is the state-dependent parallel neural Hawkes process (CT4LSTM-PPP) without its exogenous market-state input; the event-driven cell is ported unchanged. **S2P2$_{\text{pub}}$** ports the published configuration (complex DPLR states, backward-ZOH drive, relative-time dynamics, post-norm, per-type ScaledSoftplus head); the published training recipe differs from ours (larger rate, 300 epochs, MC compensator).

Any batch failure or non-finite value aborts the run.

A.12 PER-ASSET PATTERN

The assets where SS2P2 leads (ETH, SOL) and the one where it trails (BTC) differ in market character, and one reading is that SOL and ETH order flow on Coinbase carries more repeated manipulation footprints, of the kind the companion detector of Anonymous (2026) targets, while BTC flow is closer to natural trading; a decoder that fits recurring structure would then gain on ETH and SOL and lose on BTC. Two facts bound this reading. The BTC deficit is confined to timing, since SS2P2 ties NHP on next-type accuracy (0.448 vs. 0.449) and concedes only the likelihood, and expected-time MAE does not follow the split, since PCT-LSTM is lowest on ETH and SOL. We report the asset split as an observation; testing the manipulation reading requires scoring each stream's anomaly fraction with the detector, which we leave open.