# SURF: LEARNING A TIME-RESCALING MAP FOR EVENT-STREAM FORECASTING ACROSS DATASETS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Irregularly sampled multivariate event streams are hard to model with forecasters built for regularly sampled series: inter-event intervals span orders of magnitude, and time units differ from one corpus to the next. We propose Survival Flows or **SurF**, an event forecasting model built on the fact that the Time Rescaling Theorem (TRT) maps *every* point process to the same i.i.d. unit-rate exponential target. SurF learns the cumulative intensity that realizes this map; its output is unit-free comprising a single set of weights, with no per-dataset parameters, can be learned across heterogeneous event-stream datasets. We build a Transformer encoder to represent three monotone parameterizations of the cumulative intensity function that require no Monte Carlo integration to evaluate the likelihood (two closed-form, one a deterministic per-interval quadrature rule) enabling fast and accurate gradient estimation to learn the map. On six real-world benchmarks, SurF attains the best next-event time RMSE on $5/6$ and closely tails the best on Taxi. Under a leave-one-out protocol, in which the target corpus is excluded from training and model selection, SurF-GLQ attains lower time RMSE than these baselines trained the same way on $5/6$ datasets and lower held-out NLL than DTPP on all six; on Amazon, Earthquake and Taobao it even matches or beats the baselines trained on the target corpus.

## 1 INTRODUCTION

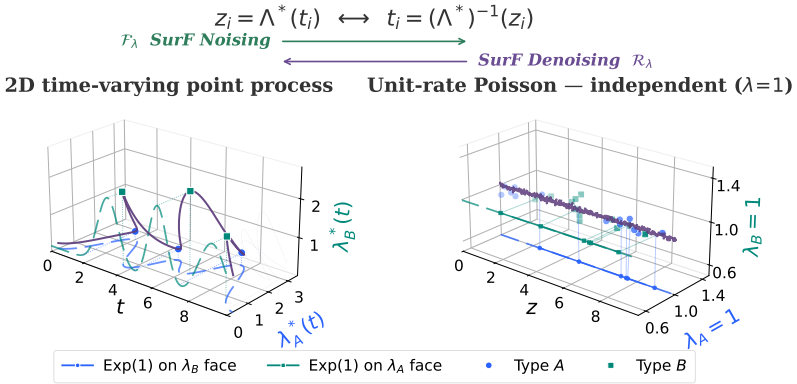


Figure 1: **The SurF forward and inverse maps in the joint intensity space.** A 2D point process with two anti-correlated event types. **(Left)** In the original time span, type $A$ activity suppresses type $B$ and vice versa. **(Right)** After the forward map $\mathcal{F}_\lambda$ (Theorem 1), the trajectory collapses to i.i.d. $\mathrm{Exp}(1)$ inter-arrival times along the rescaled axis $z$; the inverse $\mathcal{R}_\lambda$ recovers the original dynamics.

Generative models of regularly sampled time-series data such as TimesFM (Das et al., 2024), Chronos (Ansari et al., 2024), and Lag-Llama (Rasul et al., 2024) share a key assumption that data arrive at a fixed rate along synchronized channels. This makes them a poor fit for *irregularly sampled multivariate event streams*, where events arrive asynchronously, streams update at different frequencies, and observation timing itself carries semantic meaning; applying them means discarding event types and treating inter-arrival times as a regular series (Section 4). Yet such data arise throughout



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.

high-stakes domains: clinical records and neural data with variable vital-sign intervals (Rubanova et al., 2019; Rezaei et al., 2022), asynchronous financial order flows (Bacry et al., 2015; Hawkes, 2018), event-driven sensor measurements (Shukla & Marlin, 2021), and coordinated neural spike trains (Pillow et al., 2008). The standard mathematical object for these data is the *temporal point process* (TPP) (Rezaei et al., 2021; Truccolo et al., 2005), which models a stream of events through its conditional intensity $\lambda^*(t \mid \mathcal{H}_t)$; the instantaneous rate of events at time $t$ given history $\mathcal{H}_t$. Its integral, the cumulative intensity $\Lambda^*(t) = \int_0^t \lambda^*(s \mid \mathcal{H}_s)\, ds$, governs how long to wait for the next event. Fitting a TPP amounts to choosing a parameterization of $\lambda^*$ and estimating its parameters from the data.

A recurring difficulty in this fitting problem is evaluation: standard regression diagnostics do not apply to event data, and since every process has its own intensity, timescales, and shape, there is no obvious reference distribution to compare a fitted model against. In the early 2000s, the Time Rescaling Theorem (TRT) (Brown et al., 2002) resolved this: if the fitted $\Lambda$ is correct, the rescaled gaps $\Delta z_i = \Lambda(t_i) - \Lambda(t_{i-1})$ are i.i.d. $\mathrm{Exp}(1)$, so a Q–Q plot or Kolmogorov–Smirnov statistic (Truccolo et al., 2005; Brown et al., 2002) suffices for goodness-of-fit. The TRT became standard practice as an *evaluation* tool. Two decades later, neural TPPs rose in popularity as a means to leverage the representational flexibility of neural networks for event-stream modeling. Most neural TPPs are fit by maximum likelihood, which at every step requires the intensity $\lambda_\theta$ at each event and its integral $\int_0^T \lambda_\theta(s)\, ds$ over the observation window. Intensity-first methods parameterize $\lambda_\theta$ directly and estimate the integral by Monte Carlo sampling, typically with a fixed number of points per inter-event interval, which injects sampling noise into every gradient step. The cumulative-hazard family (Omi et al., 2020) avoids this by parameterizing $\Lambda_\theta$ with a monotone neural network and recovering $\lambda_\theta$ by differentiation, collapsing the window integral to a single evaluation $\Lambda_\theta(t)$. These models, however, use $\Lambda_\theta$ only for likelihood evaluation and point prediction rather than as a mapping, and, like prior neural TPPs, are fit one dataset at a time.

To address these gaps we propose SurF, built on a novel observation: the TRT sends *every* point process to the same $\mathrm{Exp}(1)$ target; can we use a neural network to learn this process. SurF learns this map with a shared Transformer encoder $\phi_\theta$ over the history and a monotone head that produces a per-interval cumulative intensity $\Lambda_\theta$. We design a training procedure that minimizes the negative log-likelihood (NLL) through $\Lambda_\theta$ without Monte Carlo integration - enabling fast gradient based parameter estimation. The model can sample draws $z \sim \mathrm{Exp}(1)$ and inverts $\Lambda_\theta$ via safeguarded Newton. This enables us to obtain a single set of weights $(\phi_\theta, \Lambda_\theta)$ trained jointly on six corpora (Table 9) with no per-dataset time parameters unlocking transfer to unseen corpora(Section 4.3).

**Contributions. (1) Learning the time-rescaling map across corpora.** The TRT has been used for goodness-of-fit, likelihood evaluation, and per-dataset flows; we learn the map across corpora. SurF's head outputs the cumulative intensity, whose rescaled gaps are unit-free and, for a correct model, $\mathrm{Exp}(1)$ on every corpus, and we train one set of weights on the union of six corpora with the summed NLL and no per-dataset time parameters (Section 3.2). **(2) An architecture for this map.** A shared Transformer encoder feeds a per-interval monotone $\Lambda_\theta$ head in three parameterizations: MoE and CSB are fully closed-form; GLQ uses an unconstrained positive network with a deterministic $O(Q)$ per-interval rule of negligible practical error (Appendix G.3). **(3) Transfer learning for event streams forecasting.** Under a leave-one-out protocol, SurF-GLQ beats AttNHP, THP and DTPP trained on the same five corpora in time RMSE on $5/6$ datasets, and has lower held-out NLL than DTPP on all six (Table 10); these baselines can collapse on the held-out corpus (THP: $1.922$ vs. $0.200$ for SurF-GLQ on Taobao). On Amazon, Earthquake and Taobao, SurF-GLQ even matches or beats AttNHP, THP and DTPP trained on the target corpus itself (Appendix J.1). In-corpus, SurF attains the best next-event time RMSE on $5/6$ benchmarks and closely tails the strongest baseline on Taxi; given the same multi-corpus pretraining and finetuning, AttNHP, THP and DTPP are beaten on all six in multi-step forecasting (Table 3).

## 2 Background

Let $\mathcal{D} = \{(t_i, k_i)\}_{i=1}^N$ denote a sequence of $N$ events in $[0, T]$ with $0 < t_1 < \cdots < t_N \leq T$ and types $k_i \in \{1, \ldots, K\}$. A temporal point process (TPP) is specified by its conditional intensity $\lambda^*(t, k \mid \mathcal{H}_t)$, where $\mathcal{H}_t = \{(t_j, k_j) : t_j < t\}$ (Guttorp, 1990). Writing the intensity $\lambda^*(t \mid \mathcal{H}_t) =$

$\sum_k \lambda^*(t, k \mid \mathcal{H}_t)$, the likelihood factorizes as

$$p(\mathcal{D}) = \Big[\prod_{i=1}^{N} \lambda^*(t_i, k_i \mid \mathcal{H}_{t_i})\Big] \exp\Big(-\int_0^T \lambda^*(s \mid \mathcal{H}_s)\, ds\Big), \tag{1}$$

with the exponential term the survival probability over $[0, T]$. The name *Survival Flow* is inspired by this structure: SurF learns a bijective flow between observed event times and the canonical survival form. Marks factorize as $\lambda^*(t, k) = \lambda^*(t)\, p(k \mid t, \mathcal{H}_t)$, which decomposes the joint NLL additively (Appendix G.1); the rest of this section develops the ground-intensity machinery.

**Time Rescaling Theorem (TRT).** The TRT describes the existence of a common mapping that SurF will map event sequences onto using a learnable flow.

*Theorem* 1 (Time Rescaling (Brown et al., 2002)). Let $\{t_i\}_{i=1}^N$ be a realization of a point process with intensity $\lambda^*(t \mid \mathcal{H}_t) > 0$, and define the cumulative intensity $\Lambda^*(t) = \int_0^t \lambda^*(s \mid \mathcal{H}_s)\, ds$, with $\Lambda^*(t) \to \infty$ as $t \to \infty$. The transformed times $z_i = \Lambda^*(t_i)$ form a unit-rate Poisson process; equivalently, the gaps $\Delta z_i = z_i - z_{i-1}$ are i.i.d. $\mathrm{Exp}(1)$ (Appendix A).

The theorem maps *any* temporal point process to the same canonical distribution, $\mathrm{Exp}(1)$, enabling the creation of a flow via a generative model. To be able to generate valid samples after learning, we need to be able to learn the reverse direction, which the next section states. While Theorem 1 establishes the existence of a forward map $t \mapsto \Lambda^*(t)$ that maps a point process to a unit-rate Poisson reference, three problems remain. First, it concerns the forward direction only: it does not establish that mapping unit-rate Poisson noise back through $(\Lambda^*)^{-1}$ recovers the original process, which *sampling* requires; it does not establish *smoothness* of the inverse, which we need for gradient-based training; and it does not provide the *Jacobian* required to evaluate likelihoods under a change of variables. The following closes these gaps under a mild positivity condition on $\lambda^*$; we do that based on the triangular-map view of Shchur et al. (2020).

*Theorem* 2 (Reverse Rescaling and Bijectivity; restated from Guttorp, 1990; Shchur et al., 2020). Assume $\lambda^*(\cdot \mid \mathcal{H}_\cdot)$ is continuous between events and satisfies $\lambda^*(t \mid \mathcal{H}_t) \geq \lambda_{\min} > 0$ for all $t \geq 0$. Then $\Lambda^* : \mathbb{R}^+ \to \mathbb{R}^+$ is a strictly increasing continuous bijection, $C^1$ with $C^1$ inverse between events; if in addition $\lambda^*$ is $C^k$ there ($k \geq 0$), then both $\Lambda^*$ and $(\Lambda^*)^{-1}$ are $C^{k+1}$ there. Conversely, if $\{z_i\}$ is a realization of a unit-rate Poisson process, then $\{t_i = (\Lambda^*)^{-1}(z_i)\}$ is a realization of the original point process with conditional intensity $\lambda^*$, and the change of variables $\boldsymbol{\tau} \mapsto \boldsymbol{\Delta z}$ has lower-triangular Jacobian $J = \partial \boldsymbol{\Delta z} / \partial \boldsymbol{\tau}$ with $\det J = \prod_{i=1}^N \lambda^*(t_i \mid \mathcal{H}_{t_i}) > 0$.

Proof sketch: between events $(\Lambda^*)' = \lambda^* \geq \lambda_{\min} > 0$, so the continuous map $\Lambda^*$ is strictly increasing and piecewise $C^1$, with $\Lambda^*(0) = 0$ and $\Lambda^*(t) \geq \lambda_{\min}\, t \to \infty$; this gives a bijection $\mathbb{R}^+ \to \mathbb{R}^+$, and the inverse function theorem yields a $C^1$ inverse between events. The reverse direction is the converse of Theorem 1 (the inverse method (Guttorp, 1990)): inverting the history-conditioned cumulative intensity one interval at a time maps i.i.d. $\mathrm{Exp}(1)$ gaps to event times whose conditional survival functions are those of $\lambda^*$ (Appendix A). The two theorems define mutually inverse maps, the classical rescaling and inverse-method pair, which we refer to as the SurF bijection (Figure 1). The forward map $\mathcal{F}_\lambda(t) = \Lambda^*(t)$ warps time so that high-intensity regions stretch and low-intensity regions compress, yielding i.i.d. $\mathrm{Exp}(1)$ gaps. The inverse $\mathcal{R}_\lambda(z) = (\Lambda^*)^{-1}(z)$ reconstructs temporal structure and can be obtained by solving $dt/dz = 1/\lambda^*(t \mid \mathcal{H}_t)$ with $t(0) = 0$: high intensity makes time advance slowly (event clusters), low intensity makes it advance quickly (gaps). A detailed comparison to prior neural TPPs with a tractable cumulative intensity (FullyNN (Omi et al., 2020), log-normal mixture (Shchur et al., 2019)) is in Appendix G.2.

Theorems 1 and 2 serve as a proof of concept, establishing that—under a mild positivity assumption—there exists a smooth, invertible transformation between event times and i.i.d. $\mathrm{Exp}(1)$ noise. Here we aim to be the first to leverage these results to learn a time-rescaling map over event-stream datasets: SurF amortizes $\Lambda^*$ via a learned $\Lambda_\theta$, converting this bijection into a trainable, sampleable model, as developed in the next section.

## 3 SurF: Survival Flow for Temporal Point Processes

Previous section establishes that the forward map $\Lambda^*$ (Theorem 1) and its inverse $(\Lambda^*)^{-1}$ (Theorem 2) exists and is smooth under a mild positivity condition on $\lambda^*$ (the bijection support for the

SurF). SurF shows, for the first time, how to amortize the process of learning this bijection using a neural network.

### 3.1 Training Objective: Amortized SurF Loss

To learn the SurF bijection, we *directly amortize the cumulative intensity*: a neural network learns $\Lambda_\theta(\Delta t \mid \mathbf{h})$, and the instantaneous intensity is recovered by differentiation. As in FullyNN (Omi et al., 2020), this inverts the usual pipeline— instead of *model $\lambda$ then integrate*, we *model $\Lambda$ then differentiate*; yielding a maximum-likelihood objective whose intensity integral needs no Monte Carlo approximation. We parameterize $\Lambda_\theta : \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^+$ using a monotonically increasing function of $\Delta t = t - t_{i-1}$ conditioned on the encoded history $\mathbf{h}_{i-1}$, with $\Lambda_\theta(0 \mid \mathbf{h}) = 0$. The intensity is then

$$\lambda_\theta^\dagger(\Delta t \mid \mathbf{h}) \;=\; \frac{\partial \Lambda_\theta(\Delta t \mid \mathbf{h})}{\partial(\Delta t)} \;=\; \lambda_{\text{floor}} + \lambda_\theta(\Delta t \mid \mathbf{h}) \;\geq\; \lambda_{\text{floor}} \;>\; 0, \tag{2}$$

with positivity guaranteed by monotonicity and by the floor. Below, $\lambda_\theta$ denotes the unfloored variant-specific term and $\lambda_\theta^\dagger$ the quantity used in the objective and in our implementation (every variant carries $\lambda_{\text{floor}}\Delta t$ in $\Lambda_\theta$, so the identity above holds exactly.)

**SurF-MoE** (mixture of exponentials, closed-form). With $w_j, \gamma_j > 0$ output by the history encoder, $\lambda_\theta(\Delta t \mid \mathbf{h}) = \sum_{j=1}^J w_j e^{-\gamma_j \Delta t}, \Lambda_\theta(\Delta t \mid \mathbf{h}) = \lambda_{\text{floor}}\Delta t + \sum_{j=1}^J \frac{w_j}{\gamma_j}\big(1 - e^{-\gamma_j \Delta t}\big)$.

**SurF-CSB** (cumulative softplus basis, closed-form). $\Lambda_\theta$ is a sum of shifted softplus primitives with positive $\alpha_m, \beta_m$,$\Lambda_\theta(\Delta t \mid \mathbf{h}) = \lambda_{\text{floor}}\Delta t + \sum_{m=1}^M \alpha_m \big[\log(1 + e^{\beta_m \Delta t + \delta_m}) - \log(1 + e^{\delta_m})\big]$, whose derivative $\lambda_\theta = \sum_m \alpha_m \beta_m \, \sigma(\beta_m \Delta t + \delta_m)$ is also closed-form.

**SurF-GLQ** (unconstrained positive with per-interval quadrature). $\lambda_\theta = \mathrm{softplus}(f_\theta(\Delta t, \mathbf{h}))$ for an unconstrained network $f_\theta$; $\Lambda_\theta$ is obtained by applying fixed $Q$-point Gauss–Legendre quadrature to $\lambda_\theta^\dagger = \lambda_{\text{floor}} + \lambda_\theta$ on $[0, \tau_i]$ with $Q = 8$, so the floor is integrated exactly. This quadrature is (i) applied per inter-event interval, (ii) deterministic, of cost $O(NQ)$ with $Q$ fixed, and (iii) of negligible error at $Q = 8$ for the smooth integrands it is applied to (see Appendix D.2.2 for more details). At the same $O(NQ')$ cost, standard THP/SAHP implementations use a stochastic Monte Carlo estimate with $Q'$ points per interval, whereas GLQ is deterministic and MoE/CSB are exact. Variant-selection guidelines are in Appendix G.3. Substituting equation 2 into equation 1 and applying the exact change of variables $\boldsymbol{\tau} \mapsto \boldsymbol{\Delta z}$ with the Jacobian of Theorem 2 (Appendix B) gives the standard TPP negative log-likelihood written entirely through $\Lambda_\theta$, which we call the amortized SurF loss:

$$\mathcal{L}_{\text{SurF}}(\theta) \;=\; \sum_{i=1}^N \Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1}) + \Lambda_\theta(\Delta T \mid \mathbf{h}_N) - \sum_{i=1}^N \log \lambda_\theta^\dagger(\tau_i \mid \mathbf{h}_{i-1}), \tag{3}$$

where $\tau_i = t_i - t_{i-1}$ and $\Delta T = T - t_N$. The first two terms are forward passes of the monotone network and the third its derivative, so no term requires Monte Carlo integration. For marked sequences $\mathcal{L}_{\text{SurF}}$ is added to the mark cross-entropy $\mathcal{L}_{\text{type}}$ with scale-balancing weight $\beta_{\text{type}}$ (Appendix G.1).

**Generative Sampling.** SurF generates events by inverting the time rescaling: sample exponential noise and solve for event times under the learned cumulative intensity. Given history $\mathbf{h}$, draw $U \sim \mathrm{Uniform}(0, 1)$, set $z = -\log U$ so that $z \sim \mathrm{Exp}(1)$, and solve $\Lambda_\theta(\Delta t \mid \mathbf{h}) = z$ for $\Delta t$. Because $\Lambda_\theta$ is $C^1$ and strictly increasing, with derivative at least $\lambda_{\text{floor}} > 0$, the root exists and is unique. We use a safeguarded Newton method (Nocedal & Wright, 2006); the update $\Delta t^{(n+1)} = \Delta t^{(n)} - \frac{\Lambda_\theta(\Delta t^{(n)} \mid \mathbf{h}) - z}{\lambda_\theta^\dagger(\Delta t^{(n)} \mid \mathbf{h})}$ is accepted when it remains inside the current bracket, and bisection is used otherwise. This converges globally from any valid bracket, with local quadratic rate once inside the Newton–Kantorovich region (Appendix E.1). The mark is drawn from the softmax of $\mathrm{TypeNet}_\theta(\mathbf{h})$ and the history is updated; full procedure in Algorithm 3. Table 1 summarizes the parameterization of SurF against existing neural TPPs.

### 3.2 Multi-Corpus Training and Zero-Shot Evaluation

We train a set of weights $(\phi_\theta, \Lambda_\theta)$, with history encoder $\phi_\theta$ (Appendix I), on the union of corpora by minimizing the loss equation 3. The MLE-consistency (Proposition G.4) shows that every population minimizer recovers the true cumulative intensity of each training corpus, so the rescaled gaps

Table 1: **Parameterization comparison across neural TPPs.** "Intensity integral" is how $\int \lambda$ is obtained inside the training likelihood: exact (closed form), Monte Carlo (MC; standard implementations draw a fixed number $Q'$ of points per interval, $O(NQ')$ and stochastic), or a deterministic $O(Q)$ Gauss–Legendre rule on each $[0, \tau_i]$. The Sampling column reports the procedure used to generate event times given a learned model. *The original FullyNN inverts its cumulative hazard at $\Phi(\tau) = \log 2$ by bisection for a median point prediction and specifies no sampler over $z \sim \mathrm{Exp}(1)$; in our evaluation all models use the predictive mean (Section 4).

| Method | Parameterizes | Likelihood form | Intensity integral | Sampling |
|---|---|---|---|---|
| RMTPP (Du et al., 2016) | $\lambda$ (RNN, exp. decay) | Closed-form (parametric) | Exact | Inverse CDF |
| NHP (Mei & Eisner, 2017) | $\lambda$ (CT-LSTM) | MC integral | MC, $O(NQ')$ | Thinning |
| THP (Zuo et al., 2020) | $\lambda$ (Transformer) | MC / Simpson | MC, $O(NQ')$ | Thinning |
| SAHP (Zhang et al., 2020) | $\lambda$ (self-attention) | MC | MC, $O(NQ')$ | Thinning |
| IFTPP (Shchur et al., 2019) | Density (log-normal mixture) | Closed-form | Implicit ($-\log S$) | Sample density |
| *FullyNN (Omi et al., 2020) | $\Lambda$ (monotone FFN) | Closed-form | Exact | Bisection at $\Phi = \log 2$ |
| **SurF-MoE** (ours) | $\Lambda$ (mixture of exponentials) | **Closed-form** | **Exact** | Newton on $\Lambda^{-1}$ |
| **SurF-CSB** (ours) | $\Lambda$ (cumulative softplus basis) | **Closed-form** | **Exact** | Newton on $\Lambda^{-1}$ |
| **SurF-GLQ** (ours) | $\lambda = \mathrm{softplus}(f_\theta)$, $\Lambda$ via GLQ | Per-interval $O(Q)$ | Deterministic $O(NQ)$ | Newton on $\Lambda^{-1}$ |

are $\mathrm{Exp}(1)$ on every corpus in the training mixture (Corollary 1); we make no such guarantee beyond it. Testing transfer outside the training mixture is nonetheless reasonable because the target the model fits does not depend on the corpus: whatever a corpus's time units, the rescaled gaps $\Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1})$ are unit-free and, for a correct model, $\mathrm{Exp}(1)$. Section 4.3 (Appendix J.1) tests this empirically under leave-one-out, where the target corpus is withheld from training. Zero-shot evaluation is restricted to event times, since a held-out corpus's event types lie outside the type vocabulary of the training corpora.

# 4 Experiments

We assess SurF for irregular event streams by asking: **(i)** Can the SurF bijection reconstruct rich intensity patterns? (§4.1). **(ii)** Can a single shared $\Lambda_\theta$ match or outperform dataset-specific models on next-event prediction across six diverse corpora? (§4.2). **(iii)** Do event-time predictions transfer zero-shot to unseen datasets? (§4.3). To investigate these questions, we use six benchmarks covering time scales from seconds to days: A) **Amazon** (Ni et al., 2019), B) **Retweet** (Zhou et al., 2013), C) **Taxi** (Whong, 2014), D) **Earthquake** (Xue et al., 2024), E) **StackOverflow** (Leskovec & Krevl, 2014), and F) **Taobao** (Xue et al., 2022) ($\sim$1.3M events, $\sim$32K sequences, and 75 event types in total across the six corpora; Appendix H). We use RMSE for time prediction and accuracy for type prediction. **Protocol.** SurF's time prediction is the predictive mean over 20 samples from the safeguarded Newton solver (Appendix E); every baseline likewise uses the predictive mean of its own sampler. All baselines are rerun by us on the EasyTPP splits and preprocessing (Xue et al., 2024), and all RMSEs are in the same EasyTPP time units. TimesFM is applied zero-shot to each sequence's inter-arrival series (marks ignored). Hyperparameters and checkpoints are selected on validation RMSE, and the test split is used only for reporting. Multi-horizon predictions are generated autoregressively, feeding the sampled (not ground-truth) event back into the encoder.

## 4.1 Proof of Concept: Recovering Oscillatory Firing Rates

Before turning to benchmarks, we verify on a controlled synthetic process what SurF promises to learn. We generated synthetic spike trains from $\lambda(t) = 0.5 + 19.5\big((1 + \cos(\pi t))/2\big)^3$ (base rate 0.5 Hz, peak 20 Hz) and trained on 200 trials of 6 s. SurF-MoE accurately recovers the oscillatory peaks and matches the ground-truth inter-spike interval (ISI) distribution (Figure 4). A Hawkes fit is shown as a classical parametric reference; its self-exciting form cannot express an exogenous periodic rate, which is exactly the structure SurF recovers.

## 4.2 Next-Event Prediction

Table 2 reports next-event RMSE and type accuracy for per-dataset baselines and for the joint and finetuned models of all SurF variants. We evaluate SurF in two regimes: **joint** (a single checkpoint

trained on the union of all six training corpora, $\{A - F\}_{train}$ datasets, and evaluated per-test-set, $\{A - F\}_{test}$; the target's training split is included in the training mix) and **finetuned** (the same checkpoint finetuned per dataset, e.g., finetune on $A_{train}$ and test on $A_{test}$). SurF (finetuned) variants achieve the best time RMSE on $5/6$ and are close to the strongest baseline on Taxi; SurF-GLQ alone is best on $4/6$ (all except Amazon, where MoE is best, and Taxi, where DTPP is).

Standard TPP evaluation trains and tests a separate model per corpus (Table 2); the shared, unit-free target of Section 3.2 instead lets a single checkpoint be evaluated in a new regime, zero-shot on a corpus it never trained on. We test this under a strict leave-one-out protocol (Table 3): e.g. we train on $\{A - E\}_{train}$ and test on $F_{test}$, with $F$'s training *and* validation splits excluded from checkpoint and hyperparameter selection and no finetuning on the target. AttNHP, THP and DTPP are also run under all three protocols of Table 3 (joint, per-dataset, leave-one-out). Figure 6 compares finetuned SurF against the DTPP, NHP, and NJDTPP baselines on next-event RMSE and type accuracy.

Table 2: Next-event prediction: time RMSE / type accuracy (%). **Bold** = best, underline = second-best. "–" = not available: FullyNN and TimesFM do not model marks, and MHP, NJDTPP and GRUwE produced no result on the corpora left blank. † DTPP conditions its mark prediction on the ground-truth next-event time, which biases its type accuracy upward.

| Model | Amazon | Retweet | Taxi | Earthquake | StackOverflow | Taobao |
|---|---|---|---|---|---|---|
| MHP (Hawkes, 1971) | 0.635 / 24.1 | 22.92 / 44.3 | 0.382 / 90.5 | – / – | 1.388 / 35.0 | 0.539 / 31.9 |
| RMTPP (Du et al., 2016) | 0.620 / 31.9 | 22.31 / 55.9 | 0.371 / 90.5 | 2.087 / 45.1 | 1.376 / 35.0 | 0.531 / 44.2 |
| NHP (Mei & Eisner, 2017) | 0.621 / 32.9 | 21.90 / 60.0 | 0.369 / 91.5 | 2.294 / 45.2 | 1.372 / 45.0 | 0.531 / 44.2 |
| SAHP (Zhang et al., 2020) | 0.619 / 32.3 | 22.40 / 58.4 | 0.372 / 90.3 | 1.832 / 44.3 | 1.375 / 43.9 | 0.532 / 45.4 |
| AttNHP (Yang et al., 2022) | 0.621 / 34.7 | 22.19 / 59.9 | 0.370 / 91.3 | 1.853 / 45.1 | 1.374 / 45.0 | 0.531 / 46.4 |
| THP (Zuo et al., 2020) | 0.621 / 33.9 | 22.01 / 58.5 | 0.370 / 91.3 | 2.357 / 45.0 | 1.374 / 45.0 | 0.531 / 46.4 |
| ODETPP (Chen et al., 2021) | 0.620 / 34.2 | 22.48 / 56.8 | 0.371 / 89.5 | 2.532 / 44.8 | 1.374 / 43.2 | 0.533 / 44.6 |
| FullyNN (Omi et al., 2020) | 0.615 / – | 21.92 / – | 0.373 / – | 2.330 / – | 1.375 / – | 0.529 / – |
| IFTPP (Shchur et al., 2019) | 0.618 / 32.5 | 22.18 / 60.3 | 0.377 / 91.4 | 2.550 / 46.7 | 1.374 / 43.2 | 0.531 / 44.6 |
| DTPP (Panos, 2024) | 0.346 / **40.9**† | 17.89 / 60.0† | **0.283** / **92.9**† | 1.372 / **47.3**† | 1.034 / **49.6**† | 0.224 / 59.9† |
| NJDTPP (Zhang et al., 2024) | – / – | 17.50 / **60.8** | 0.296 / 90.8 | 1.420 / 47.2 | 1.112 / 46.9 | 0.130 / 48.6 |
| GRUwE (Joshi & Hauskrecht, 2026) | 0.611 / 38.8 | 22.21 / 59.2 | 0.369 / 91.4 | – / – | 1.369 / 44.7 | – / – |
| TimesFM (Das et al., 2024) | 0.386 / – | 17.69 / – | 0.316 / – | 1.367 / – | 1.149 / – | 0.131 / – |
| SurF-MoE (joint) | 0.343 / 34.4 | 16.04 / 59.2 | 0.491 / 89.7 | 1.541 / 47.1 | 1.255 / 44.8 | 0.231 / 60.0 |
| SurF-MoE (fine) | **0.336** / 34.7 | 16.23 / 59.7 | 0.311 / 89.5 | 1.258 / 47.0 | 1.077 / 45.1 | 0.189 / **60.3** |
| SurF-CSB (joint) | 0.410 / 30.4 | 15.89 / 56.8 | 0.456 / 90.3 | 1.294 / 47.1 | 1.104 / 44.5 | 0.369 / 51.5 |
| SurF-CSB (fine) | 0.409 / 33.1 | 15.81 / 59.5 | 0.288 / 90.5 | 1.256 / 47.2 | 1.039 / 44.7 | 0.224 / 57.2 |
| SurF-GLQ (joint) | 0.477 / 33.3 | 15.80 / 59.1 | 0.330 / 88.2 | 1.294 / 47.2 | 1.098 / 44.4 | 0.127 / 53.4 |
| SurF-GLQ (fine) | 0.409 / 34.2 | **15.78** / 59.8 | 0.286 / 90.9 | **1.228** / **47.3** | **1.018** / 44.6 | **0.126** / 56.2 |

On Earthquake, SurF-GLQ (finetuned) improves $\sim$10% over TimesFM and DTPP and $\sim$13.5% over NJDTPP; all three finetuned variants beat every baseline, and the jointly-trained SurF-CSB and SurF-GLQ checkpoints already do so without any per-dataset finetuning. On Retweet, SurF-GLQ (finetuned) improves $\sim$10% over NJDTPP and $\sim$12% over DTPP; its jointly-trained counterpart again leads all baselines. SurF-GLQ (finetuned) attains the best Taobao time RMSE, and SurF-MoE (finetuned) the top Taobao type accuracy among TPP methods. On Amazon and StackOverflow, SurF beats DTPP on time RMSE; on Taxi the gap is within 1.1%. The best SurF variant outperforms the much larger foundation model TimesFM (applied zero-shot) on every dataset. Without any per-dataset finetuning, the best jointly-trained checkpoint already beats every baseline in time RMSE on $4/6$ datasets (Amazon, Retweet, Earthquake, Taobao). Finetuning further improves time RMSE on at least $5/6$ datasets per variant (CSB on $6/6$) and type accuracy on $6/6$ for CSB and GLQ. Figure 2 reports the Kolmogorov–Smirnov (KS) statistic $D = \sup_x |G(x) - F(x)|$ between the $\mathrm{Exp}(1)$ CDF $G$ and the empirical CDF $F$ of a variant's residuals $\Delta z_i$, which are i.i.d. $\mathrm{Exp}(1)$ under a perfectly specified model (TRT (Brown et al., 2002)). All three SurF variants reach $D \leq 0.10$ on all three datasets shown (Taxi, StackOver-

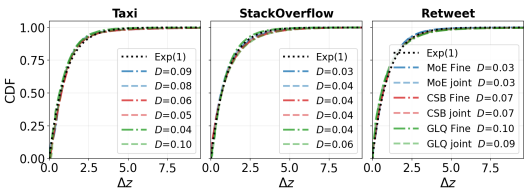


Figure 2: Time-rescaling goodness-of-fit on Taxi, StackOverflow and Retweet. Empirical CDF of the residuals $\Delta z_i$ on the test split. A perfectly calibrated model produces residuals that are i.i.d. $\mathrm{Exp}(1)$; smaller $D$ indicates better calibration.

flow and Retweet). Because the KS test flags even small deviations at our test-set sizes, we use $D$ as a relative measure of calibration across variants and datasets.

**Goodness-of-fit via NLL.** We report held-out NLL for the SurF variants in Appendix C: MoE is lowest on $4/6$ datasets (Taxi, StackOverflow, Earthquake, Taobao) and GLQ on the two heaviest-tailed ones (Retweet, Amazon), and the jointly-trained checkpoint differs from the finetuned one by $0.06$ nats/event on average.

### 4.3 Multi-Horizon Forecasting

We test whether event-time predictions transfer zero-shot to held-out datasets by generating the next $h \in \{1, \ldots, 10\}$ events under three protocols (joint, finetuned, leave-one-out) and measuring per-horizon RMSE and type accuracy. Figure 3 shows that iterated autoregressive decoding remains

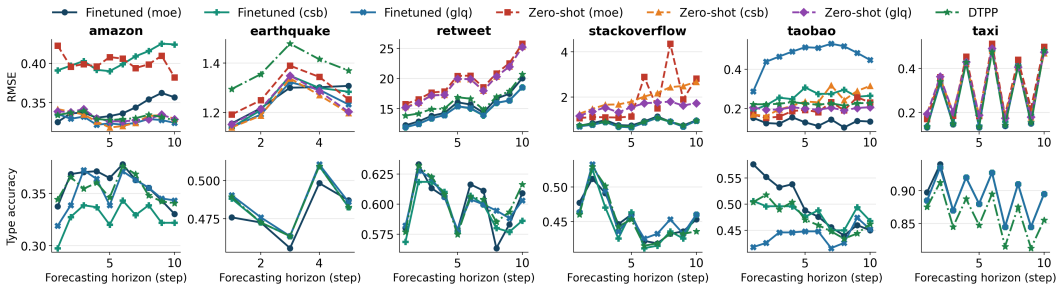


Figure 3: Per-horizon inter-arrival RMSE (top) and type accuracy (bottom) for finetuned (solid) and zero-shot (dashed) SurF variants and DTPP (dash-dotted) over 1–10 future events (Earthquake to 5). Zero-shot curves report time RMSE only (Section 3.2).

stable: finetuned RMSE grows sub-linearly with the horizon, most visibly on Retweet. On Taxi, every model, DTPP included, alternates between low and high error at successive horizons, so this pattern reflects the data rather than SurF. Type accuracy remains close to its one-step value on most datasets.

Finetuned SurF matches or beats DTPP, the strongest baseline, on time RMSE on all six datasets, with margins from a tie on Taxi (where SurF additionally leads on type accuracy) to $\sim 41\%$ on Taobao. This is not only a data advantage: AttNHP, THP and DTPP given the same joint pretraining and per-dataset finetuning (the "fine" rows) are beaten on all six, e.g. $0.285$ vs. $0.321$ for DTPP (fine) on Taxi. On type accuracy, finetuned SurF leads on $4/6$ datasets among non-$^\dagger$ comparisons, though on Retweet and StackOverflow the margin is under $0.05$ points. GLQ wins outright on four datasets, ties DTPP on Taxi, and is beaten only on Taobao, where MoE wins.

**Zero-shot vs. baselines.** Under the leave-one-out protocol, SurF-GLQ alone beats every leave-one-out baseline in time RMSE on $5/6$ datasets, with the largest gaps where baselines collapse, and is within $0.032$ of DTPP on StackOverflow; its held-out NLL is lower than DTPP's on all six, by at least $0.26$ nats/event (Table 10). Across variants, GLQ transfers best on $3/6$, CSB on $2/6$, and MoE on Taobao. Relative to each model's own best in-corpus time RMSE in Table 3 (the minimum over its in-corpus rows), every SurF variant's leave-one-out RMSE stays below $1.6\times$ on all six corpora, whereas AttNHP, THP and DTPP each exceed $3.4\times$ on at least one (AttNHP $4.6\times$ on Amazon; THP $7.9\times$ and DTPP $3.6\times$ on Taobao). On Amazon, Earthquake and Taobao, the held-out checkpoint also matches or beats every baseline in Table 3 that was trained on the target corpus (Appendix J.1); for SurF-CSB on these three datasets it even outperforms its own joint checkpoint.

**Joint training.** In the joint training setting (top block of Table 3), SurF attains the best time RMSE on $5/6$ datasets; among variants, MoE leads on $3/6$, CSB on $2/6$, and GLQ on $1/6$.

### 4.4 Ablations

Here, we ablate two design choices of SurF and report a hyperparameter sweep: (i) the *conformer* block inserted before multi-head attention (Appendix I); (ii) the *loss schedule*: *Equal* keeps the time and type losses at fixed weights throughout, *Staged* drops the SurF likelihood and finetunes

Table 3: Multi-time-horizon forecasting: inter-event time RMSE / type accuracy (%). Blocks: joint (trained on all six training splits); per-dataset (baselines trained on the target only, then joint-then-finetuned models); leave-one-out (target corpus excluded from training and from model selection). **Bold** = best within block, underline = second-best. "–" = not applicable (Section 3.2). † DTPP conditions its mark prediction on the ground-truth next-event time.

| Model | Amazon | Retweet | Taxi | Earthquake | StackOverflow | Taobao |
|---|---|---|---|---|---|---|
| AttNHP (joint) (Yang et al., 2022) | 1.508 / 34.3 | 19.00 / 59.72 | 0.944 / 58.58 | 1.912 / 48.12 | 1.616 / 42.65 | 0.823 / 46.54 |
| THP (joint) (Zuo et al., 2020) | 1.053 / 31.91 | 21.77 / 54.88 | 0.364 / 45.00 | 1.426 / 48.27 | 0.989 / 43.80 | 0.243 / 43.62 |
| DTPP (joint) (Panos, 2024) | **0.332 / 36.14** | 16.32 / **59.89** | 0.321 / 89.50 | 1.383 / 48.33 | 0.861 / 44.84 | 0.224 / 47.28 |
| SurF-MoE (joint) | 0.336 / 35.66 | **15.26** / 59.38 | 0.323 / 89.20 | **1.349** / 48.16 | 0.866 / 45.40 | 0.132 / **49.78** |
| SurF-CSB (joint) | 0.402 / 31.91 | 15.27 / 56.09 | **0.319** / 89.10 | 1.380 / 48.29 | **0.836** / 45.92 | 0.361 / 47.38 |
| SurF-GLQ (joint) | 0.582 / 34.61 | 19.73 / 59.15 | 0.326 / **89.70** | 1.438 / **48.34** | 0.846 / **46.06** | **0.129** / 49.04 |
| AttNHP (Yang et al., 2022) | 0.657 / **36.48** | 18.18 / 58.91 | 0.397 / 85.35 | 1.911 / 48.34 | 1.235 / 45.89 | 0.427 / 48.20 |
| THP (Zuo et al., 2020) | 0.657 / 35.53 | 18.51 / 59.65 | 0.371 / 42.90 | 2.746 / 46.84 | 1.218 / 43.00 | 0.357 / 43.62 |
| DTPP (Panos, 2024) | 0.332 / 35.45† | 16.20 / **60.19**† | **0.285** / 86.20† | 1.382 / 48.35† | 0.851 / 45.19† | 0.223 / 47.34† |
| AttNHP (fine) (Yang et al., 2022) | 0.431 / 29.52 | 17.28 / 60.07 | 0.543 / 82.95 | 2.790 / **48.63** | 2.102 / 45.12 | 2.518 / 44.66 |
| THP (fine) (Zuo et al., 2020) | 0.594 / 36.30 | 21.78 / 59.83 | 0.379 / 89.20 | 1.821 / 48.25 | 1.102 / 45.47 | 0.279 / 47.32 |
| DTPP (fine) (Panos, 2024) | 0.332 / 34.22† | 16.31 / 59.72† | 0.321 / 86.25† | 1.383 / 48.40† | 0.861 / 44.80† | 0.223 / 46.86† |
| SurF-MoE (fine) | 0.341 / 35.91 | 15.52 / 59.94 | 0.325 / 89.30 | 1.258 / 47.80 | 0.871 / 45.51 | **0.132** / **49.68** |
| SurF-CSB (fine) | 0.405 / 32.75 | 14.82 / 59.43 | 0.320 / **89.38** | 1.258 / 48.34 | 0.837 / 45.75 | 0.261 / 48.20 |
| SurF-GLQ (fine) | **0.327** / 35.10 | **14.71** / 60.10 | **0.285** / 89.10 | **1.236** / 48.50 | **0.825** / **45.90** | 0.468 / 44.00 |
| AttNHP (zero-shot (leave-one-out)) | 1.975 / – | 19.79 / – | 0.698 / – | 1.482 / – | 1.007 / – | 0.771 / – |
| THP (zero-shot (leave-one-out)) | 0.913 / – | 19.23 / – | 2.759 / – | 1.447 / – | 3.676 / – | 1.922 / – |
| DTPP (zero-shot (leave-one-out)) | 1.135 / – | 19.90 / – | 0.440 / – | 1.338 / – | **0.861** / – | 0.795 / – |
| SurF-MoE (zero-shot (leave-one-out)) | 0.401 / – | 19.86 / – | 0.349 / – | 1.287 / – | 0.984 / – | **0.198** / – |
| SurF-CSB (zero-shot (leave-one-out)) | **0.330** / – | 19.35 / – | 0.333 / – | **1.229** / – | 0.963 / – | 0.243 / – |
| SurF-GLQ (zero-shot (leave-one-out)) | 0.332 / – | **19.10** / – | **0.310** / – | 1.241 / – | 0.893 / – | 0.200 / – |

on the type loss only, and *Hybrid* keeps the SurF likelihood in every stage but raises $\beta_{\text{type}}$ during finetuning; and (iii) a sweep of SurF-GLQ hyperparameters. Full tables are in Appendix F. Full SurF uses the Hybrid schedule for robustness: it is best on Retweet time RMSE and tied-best on Taobao type accuracy, and it is one of only two settings never worst on time RMSE, beating the other (conformer-free Equal) on $4/6$ datasets. Staged is competitive on several datasets but collapses on Retweet (RMSE $\sim 16 \rightarrow 22.3$) because dropping the SurF likelihood during finetuning discards the gradient signal that anchors long-tail inter-arrivals. The conformer block gives its largest gains under the Staged schedule (Earthquake and Taxi). The 100-trial SurF-GLQ sweep (Taobao validation split, raw time units, hence the different RMSE scale from Table 2) serves as a robustness check (Appendix F). Time RMSE varies in $[8.725, 9.242]$ with $79\%$ of trials within $1\%$ of the best run, and the top-10 configurations span $0.011$ RMSE units across several settings, so there is no single fragile optimum. The $Q{=}8$ default is supported by the Gauss–Legendre error analysis of Appendix D.2.2.

**Computational Efficiency** Based on Table 4 and against NJDTPP, the most directly comparable quadrature-based method, SurF-MoE trains $10.9\times$ faster on Amazon and $12.4\times$ on StackOverflow, with inference gaps above $50\times$; it is also $\sim 5.8\times$ faster than TimesFM at inference. Against DTPP, SurF-MoE trains $\sim 1.4\times$ faster on both datasets at $\sim 36\%$ lower peak memory. Lighter models such as THP and IFTPP train faster but are markedly less accurate (Table 2); within SurF's accuracy tier (DTPP, NJDTPP), SurF-MoE and SurF-CSB are the cheapest to train, while SurF-GLQ trades training cost for an unrestricted intensity shape.

Table 4: Training and inference cost on Amazon and StackOverflow (H100, batch 512, inference averaged over 20 sequences). Train: ms per batch. Infer.: ms per sequence.

| | Amazon | | | StackOverflow | | |
|---|---|---|---|---|---|---|
| Model | Train | Infer. | Mem (GB) | Train | Infer. | Mem (GB) |
| AttNHP | 90.9 | 9.91 | 13.1 | 81.3 | 10.3 | 11.3 |
| IFTPP | 5.96 | 2.73 | 0.26 | – | – | – |
| ODETPP | 240 | 102 | 0.37 | 248 | 125 | 0.27 |
| THP | 6.61 | 4.77 | 0.81 | 7.51 | 5.92 | 0.59 |
| DTPP | 104 | 5.57 | 8.93 | 115 | 7.42 | 9.84 |
| NJDTPP | 797 | 373 | 0.544 | 1000 | 387 | 0.77 |
| TimesFM | – | 39.7 | – | – | 40.4 | – |
| **SurF-MoE** | 72.9 | 6.82 | 5.71 | 80.8 | 6.80 | 6.23 |
| **SurF-CSB** | 73.3 | 6.79 | 5.71 | 87.4 | 7.21 | 6.71 |
| **SurF-GLQ** | 297 | 13.5 | 19.4 | 314 | 13.7 | 21.1 |

## 5 Related Work

**Generative models for time series.** Foundation models have advanced rapidly in language (Brown et al., 2020; Touvron et al., 2023), vision (Radford et al., 2021; Oquab et al., 2024), and regularly sampled time series (TimesFM (Das et al., 2024), Chronos (Ansari et al., 2024), Lag-Llama (Rasul et al., 2024), Moirai (Woo et al., 2024)). All as-

sume fixed sampling rates and synchronized channels, an assumption that fails for asynchronous event streams where inter-arrival times span orders of magnitude and timing itself carries meaning. SurF targets this regime directly.

**Neural temporal point processes.** Neural TPPs (Shchur et al., 2021; Rezaei et al., 2021) offer flexible event modeling but have not reached foundation-model scale; pretrained event models so far adapt a TPP to streaming tasks (Xue et al., 2023) or couple it with a pretrained language model over textual event descriptions (Shi et al., 2023; Liu & Quan, 2024), rather than pretraining a TPP on raw (time, mark) streams from several corpora. RNN-based methods (RMTPP (Du et al., 2016; Rezaei et al., 2018), Neural Hawkes (Mei & Eisner, 2017)) suffer from gradient pathologies and sequential bottlenecks; Transformer variants (SAHP (Zhang et al., 2020), THP (Zuo et al., 2020), and extensions (Li & Sun, 2023; Gu, 2021; Xue et al., 2022)) parallelize training but still estimate the intensity integral by Monte Carlo sampling. Intensity-free methods (IFTPP (Shchur et al., 2019)) avoid the integral by modeling the inter-event density directly.

**Cumulative-hazard neural TPPs.** Closest to SurF is FullyNN (Omi et al., 2020), which parameterizes $\Lambda_\theta$ directly and recovers $\lambda_\theta$ by differentiation; the log-normal mixture of Shchur et al. (2019) also has a closed-form cumulative intensity. We differ on three fronts. *Conceptually,* using $\Lambda$ as a bijection in both directions is the classical inverse method, made explicit for neural TPPs by the triangular-map view (likelihood via the Jacobian, sampling via the inverse; Shchur et al., 2020; Guttorp, 1990), while FullyNN uses $\Lambda_\theta$ only to avoid the intensity integral. What SurF adds is to train one model of this map across corpora through its *shared* $\mathrm{Exp}(1)$ target. *Architecturally,* prior work uses single-dataset feedforward or recurrent encoders; SurF pairs the cumulative-intensity head with a Transformer encoder and multi-dataset pretraining, and offers three monotone parameterizations (MoE, CSB, GLQ) with distinct expressiveness–cost trade-offs (Appendix G.3). *Empirically,* prior cumulative-hazard TPPs are trained and evaluated one dataset at a time; SurF is evaluated on held-out corpora. The jointly-trained SurF-GLQ checkpoint alone outperforms per-dataset FullyNN on all six benchmarks, and under the leave-one-out protocol of Table 3 the held-out checkpoint (which never sees the target corpus) still beats every leave-one-out baseline on $5/6$ datasets. See Appendix G.2 for a full comparison.

**Relation to intensity-free and flow-based models.** IFTPP (Shchur et al., 2019) fits the conditional inter-event density with a log-normal mixture. TriTPP (Shchur et al., 2020) maps to the same unit-rate Poisson base as SurF but composes learnable triangular layers instead of conditioning a per-interval cumulative intensity on a history encoder. Both are fitted per dataset. FullyNN (Omi et al., 2020) does have the cumulative intensity and does use the $\mathrm{Exp}(1)$ property, but only to solve $\Lambda_\theta^{\mathrm{FNN}}(\tau \mid \mathbf{h}_i) = \log 2$ by bisection for a median point prediction, also per dataset. The TRT fixes the same target for every corpus; SurF uses it to train one set of weights on several real corpora and evaluates the result leave-one-out (Table 3).

## 6 Conclusion

We introduced **SurF**, a forecasting model for irregularly sampled multivariate event streams that learns the Time Rescaling Theorem's invertible map between event sequences and canonical $\mathrm{Exp}(1)$ noise. SurF builds on classical point-process theory, the cumulative-hazard lineage of neural TPPs, and the triangular-map view of Shchur et al. (2020). Our contributions are multi-corpus training of this map through its shared $\mathrm{Exp}(1)$ target, a Transformer-amortized per-interval cumulative intensity with three monotone parameterizations (Appendix G.3), and a leave-one-out evaluation of cross-dataset transfer against baselines trained under the same protocol. With the target corpus held out from training, SurF-GLQ beats every leave-one-out baseline in time RMSE on $5/6$ benchmarks and DTPP in held-out NLL on $6/6$, and on Amazon, Earthquake and Taobao it also matches or beats every baseline in Table 3 that saw the target corpus.

**Limitations and future work.** Bijectivity (Theorem 2) relies on a the intensity having a floor whose effect on the likelihood is negligible on our benchmarks; for processes with genuine dead-time, the learnable-floor and masking schemes of Appendix A.1 apply. The closed-form variants trade shape flexibility for exactness (decaying between-event intensity for MoE, increasing for CSB), while GLQ is unrestricted; summing the MoE and CSB bases would give a closed-form variant covering both. Extending zero-shot prediction to marks, and scaling multi-corpus pretraining to larger and more diverse event corpora, are natural next steps.

## GENERATIVE AI USAGE DISCLOSURE

In our paper, we used generative AI tools to help edit and polish the writing (grammar, phrasing, clarity) and to assist in the implementation of our methods (i.e., writing or updating code, with all changes reviewed by the authors before being accepted). We did not use generative AI for any other required-disclosure research tasks (e.g., developing models/frameworks, formulating or proving mathematical claims, designing methodology/experiments, generating or cleaning data, interpreting results); translation and qualitative/thematic analysis are not applicable to this work. We have reviewed all AI-assisted edits and AI-assisted code for accuracy, correctness, and fidelity to the original meaning, and we take full responsibility for the final content of this paper.

## REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance, 2015. URL https://arxiv.org/abs/1502.04592.

Emery N. Brown, Riccardo Barbieri, Valérie Ventura, Robert E. Kass, and Loren M. Frank. The time-rescaling theorem and its application to neural spike train data analysis. *Neural Computation*, 14(2):325–346, 02 2002. ISSN 0899-7667. doi: 10.1162/08997660252741149. URL https://doi.org/10.1162/08997660252741149.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Ricky T. Q. Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes, 2021. URL https://arxiv.org/abs/2011.04583.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/das24c.html.

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 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016. URL https://www.kdd.org/kdd2016/papers/files/rpp1081-duA.pdf.

Yulong Gu. Attentive neural point processes for event forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 7592–7600, 2021.

Peter Guttorp. An introduction to the theory of point processes (d. j. daley and d. vere-jones). *SIAM Review*, 32(1):175–176, 1990. doi: 10.1137/1032025. URL https://doi.org/10.1137/1032025.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58:83–90, 1971. URL https://api.semanticscholar.org/CorpusID:14122089.

Alan G. Hawkes. Hawkes processes and their applications to finance: a review. *Quantitative Finance*, 18(2):193–198, 2018. doi: 10.1080/14697688.2017.1403131. URL https://doi.org/10.1080/14697688.2017.1403131.

Ankitkumar Joshi and Milos Hauskrecht. Still competitive: Revisiting recurrent models for irregular time series prediction, 2026. URL https://arxiv.org/abs/2510.16161.

Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection, 2014. URL https://snap.stanford.edu/data/.

Zhuoqun Li and Mingxuan Sun. Sparse transformer hawkes process for long event sequences. In *Machine Learning and Knowledge Discovery in Databases: Research Track: European Conference, ECML PKDD 2023, Turin, Italy, September 18–22, 2023, Proceedings, Part V*, pp. 172–188, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-43423-5. doi: 10.1007/978-3-031-43424-2_11. URL https://doi.org/10.1007/978-3-031-43424-2_11.

Zefang Liu and Yinzhu Quan. Tpp-llm: Modeling temporal point processes by efficiently fine-tuning large language models. *ArXiv*, abs/2410.02062, 2024. URL https://api.semanticscholar.org/CorpusID:273098350.

Hongyuan Mei and Jason Eisner. The neural Hawkes process: A neurally self-modulating multivariate point process. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1612.09328.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 188–197, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1018. URL https://aclanthology.org/D19-1018/.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 2006.

Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes, 2020. URL https://arxiv.org/abs/1905.09690.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. URL https://arxiv.org/abs/2304.07193.

Aristeidis Panos. Decomposable transformer point processes. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Jonathan W. Pillow, Jonathon Shlens, Liam Paninski, Alexander Sher, Alan M. Litke, E. J. Chichilnisky, and Eero P. Simoncelli. Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature*, 454:995 – 999, 2008. URL https://api.semanticscholar.org/CorpusID:306143.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *ArXiv*, abs/2103.00020, 2021. URL https://api.semanticscholar.org/CorpusID:231591445.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-llama: Towards foundation models for probabilistic time series forecasting, 2024. URL https://arxiv.org/abs/2310.08278.

Mohammad R Rezaei, Anna K Gillespie, Jennifer A Guidera, Behzad Nazari, Saeid Sadri, Loren M Frank, Uri T Eden, and Ali Yousefi. A comparison study of point-process filter and deep learning performance in estimating rat position using an ensemble of place cells. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 4732–4735. IEEE, 2018.

Mohammad R Rezaei, Alex E Hadjinicolaou, Sydney S Cash, Uri T Eden, and Ali Yousefi. Direct discriminative decoder models for analysis of high-dimensional dynamical neural data. *Neural Computation*, 34(5):1100–1135, 2022.

Mohammad Reza Rezaei, Kensuke Arai, Loren M Frank, Uri T Eden, and Ali Yousefi. Real-time point process filter for multidimensional decoding problems using mixture models. *Journal of neuroscience methods*, 348:109006, 2021.

Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf.

Oleksandr Shchur, Marin Bilos, and Stephan Günnemann. Intensity-free learning of temporal point processes. *ArXiv*, abs/1909.12127, 2019. URL https://api.semanticscholar.org/CorpusID:202889230.

Oleksandr Shchur, Nicholas Gao, Marin Biloš, and Stephan Günnemann. Fast and flexible temporal point processes with triangular maps. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural temporal point processes: A review, 2021. URL https://arxiv.org/abs/2104.03528.

Xiaoming Shi, Siqiao Xue, Kangrui Wang, Fan Zhou, James Y Zhang, Jun Zhou, Chenhao Tan, and Hongyuan Mei. Language models can improve event prediction by few-shot abductive reasoning. *arXiv preprint arXiv:2305.16646*, 2023. URL https://arxiv.org/abs/2305.16646.

Satya Narayan Shukla and Benjamin M. Marlin. Multi-time attention networks for irregularly sampled time series, 2021. URL https://arxiv.org/abs/2101.10318.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aur'elien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *ArXiv*, abs/2302.13971, 2023. URL https://api.semanticscholar.org/CorpusID:257219404.

Wilson Truccolo, Uri T. Eden, Matthew R. Fellows, John P. Donoghue, and Emery N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of Neurophysiology*, 93(2):1074–1089, 2005. doi: 10.1152/jn.00697.2004. URL https://doi.org/10.1152/jn.00697.2004. PMID: 15356183.

A.W. van der Vaart. *Asymptotic Statistics*. Asymptotic Statistics. Cambridge University Press, 2000. ISBN 9780521784504. URL https://books.google.ca/books?id=UEuQEM5RjWgC.

Chris Whong. FOILing NYC's taxi trip data, 2014. URL https://chriswhong.com/open-data/foil_nyc_taxi/.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Siqiao Xue, Xiaoming Shi, Y James Zhang, and Hongyuan Mei. Hypro: A hybridly normalized probabilistic model for long-horizon prediction of event sequences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2210.01753.

Siqiao Xue, Yan Wang, Zhixuan Chu, Xiaoming Shi, Caigao Jiang, Hongyan Hao, Gangwei Jiang, Xiaoyun Feng, James Zhang, and Jun Zhou. Prompt-augmented temporal point process for streaming event sequence. *Advances in Neural Information Processing Systems*, 36:18885–18905, 2023.

Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao Jiang, Chen Pan, James Y. Zhang, Qingsong Wen, Jun Zhou, and Hongyuan Mei. Easytpp: Towards open benchmarking temporal point processes. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2307.08097.

Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2201.00044.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive Hawkes process. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2020. URL http://proceedings.mlr.press/v119/zhang20q/zhang20q.pdf.

Shuai Zhang, Chuan Zhou, Yang Aron Liu, Peng Zhang, Xixun Lin, and Zhi-Ming Ma. Neural jump-diffusion temporal point processes. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 60541–60557. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhang24cm.html.

Ke Zhou, Hongyuan Zha, and Le Song. Learning triggering kernels for multi-dimensional hawkes processes. In *International Conference on Machine Learning*, 2013. URL https://api.semanticscholar.org/CorpusID:6191094.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *International Conference on Machine Learning*, pp. 11692–11702. PMLR, 2020. URL https://arxiv.org/pdf/2002.09291.pdf.

# CONTENTS


**1 Introduction** **1**

**2 Background** **2**

**3 SurF: Survival Flow for Temporal Point Processes** **3**

3.1 Training Objective: Amortized SurF Loss . . . . . 4

3.2 Multi-Corpus Training and Zero-Shot Evaluation . . . . . 4

**4 Experiments** **5**

4.1 Proof of Concept: Recovering Oscillatory Firing Rates . . . . . 5

4.2 Next-Event Prediction . . . . . 5

4.3 Multi-Horizon Forecasting . . . . . 7

4.4 Ablations . . . . . 7

**5 Related Work** **8**

**6 Conclusion** **9**

**A Time Rescaling Theorem and Bijectivity Proofs** **16**

A.1 Near-Zero Intensity Regimes and Likelihood Consistency . . . . . 17

**B Likelihood Derivation via Change of Variables** **17**

B.1 Jacobian Analysis . . . . . 18

B.2 Change of Variables Formula . . . . . 18

B.3 Observation Window Correction . . . . . 19

**C Held-Out Negative Log-Likelihood** **19**

**D Amortized Event Intensity: Detailed Derivations** **20**

D.1 SurF-CSB: Cumulative Softplus Basis . . . . . 20

D.2 SurF-GLQ: Gauss–Legendre Quadrature . . . . . 20

D.2.1 Gradient Computation under Gauss–Legendre Quadrature . . . . . 21

D.2.2 Quadrature Error in the Parameter Gradient . . . . . 21

D.3 SurF-MoE: Mixture-of-Exponentials (Closed-Form) . . . . . 21

D.4 Closed-Form Loss under MoE . . . . . 22

**E Sampling Algorithms** **23**

E.1 Safeguarded Newton: Formal Convergence Guarantee . . . . . 23

**F Extended Hyperparameter Sensitivity Study** **24**

**G Additional Theoretical Material** **27**

G.1 Joint Modeling of Times and Marks . . . . . 27
G.2 Relationship to Prior Cumulative-Hazard Neural TPPs . . . . . 27
G.3 When to Use Which Variant, and the Role of Quadrature . . . . . 28
G.4 Shared Canonical Target . . . . . 28

**H Datasets** **28**

**I Model Architecture and Hyperparameters** **29**
I.1 History Encoding . . . . . 29
I.2 Cumulative-Intensity Head . . . . . 29
I.3 Type Prediction Head . . . . . 29
I.4 Hyperparameter Search . . . . . 30

**J Additional Empirical Analysis** **30**
J.1 Leave-One-Out Transfer Against In-Corpus Baselines . . . . . 30

# A TIME RESCALING THEOREM AND BIJECTIVITY PROOFS

Consider a temporal point process defined on $\mathbb{R}^+$ generating event times $\{t_1, t_2, \ldots, t_N\}$ with a conditional intensity function $\lambda^*(t \mid \mathcal{H}_t) > 0$ for all $t \geq 0$, where $\mathcal{H}_t$ represents the history of events up to time $t$. The intensity function characterizes the instantaneous rate of event occurrence and fully describes the probabilistic structure of the process.

*Theorem* 1 (Time Rescaling Theorem, restated). Let $\{t_i\}_{i=1}^N$ be a realization of a point process with conditional intensity function $\lambda^*(t \mid \mathcal{H}_t) > 0$. Define the cumulative intensity function $\Lambda^*(t) = \int_0^t \lambda^*(s \mid \mathcal{H}_s)\, ds$, assumed to satisfy $\Lambda^*(t) \to \infty$ as $t \to \infty$, and the rescaled event times $z_i = \Lambda^*(t_i)$. Then $\{\Delta z_i = z_i - z_{i-1}\}_{i=1}^N$ with $z_0 = 0$ are i.i.d. exponential random variables with unit rate.

*Proof.* Fix $i \in \{1, \ldots, N\}$ and condition on $\mathcal{H}_{t_i}$, the first $i-1$ events, with $t_0 = 0$. By definition of the conditional intensity, the conditional survival function of the $i$-th event time is, for $t \geq t_{i-1}$,

$$\Pr(t_i > t \mid \mathcal{H}_{t_i}) = \exp\Big(-\int_{t_{i-1}}^{t} \lambda^*(s \mid \mathcal{H}_s)\, ds\Big) = \exp\big(-[\Lambda^*(t) - \Lambda^*(t_{i-1})]\big). \tag{4}$$

Since $\lambda^* > 0$, $\Lambda^*$ is strictly increasing and hence invertible on its range, and $\Lambda^*(t) \to \infty$ drives the right-hand side to 0, so $t_i < \infty$ almost surely. For $u \geq 0$ set $t = (\Lambda^*)^{-1}(\Lambda^*(t_{i-1}) + u)$; monotonicity gives $\{\Delta z_i > u\} = \{t_i > t\}$, so

$$\Pr(\Delta z_i > u \mid \mathcal{H}_{t_i}) = \exp(-u), \qquad u \geq 0. \tag{5}$$

The conditional law of $\Delta z_i$ is thus $\mathrm{Exp}(1)$ whatever $\mathcal{H}_{t_i}$ is, so $\Delta z_i$ is independent of $\mathcal{H}_{t_i}$ and in particular of $\Delta z_1, \ldots, \Delta z_{i-1}$, which are $\mathcal{H}_{t_i}$-measurable. Applying this for $i = 1, \ldots, N$ gives that the $\Delta z_i$ are i.i.d. $\mathrm{Exp}(1)$. □

The crucial aspect of this transformation is its bijectivity. The condition $\lambda^*(t \mid \mathcal{H}_t) > 0$ ensures that $\Lambda^*(t)$ is strictly increasing.

**Corollary 1.** *Let $\{t_i\}_{i=1}^N$ be a realization of a point process with conditional intensity function $\lambda_0(t \mid \mathcal{H}_t) > 0$. For any $t > 0$, let $t' = \max(\{t_j : (t_j, k_j) \in \mathcal{H}_t\} \cup \{0\})$, and write $\tilde{\lambda}(\Delta t \mid \mathcal{H}_t) = \lambda_0(t \mid \mathcal{H}_t)$, where $\Delta t = t - t'$. Define the cumulative intensity functions $\Lambda_0(t) = \int_0^t \lambda_0(s \mid \mathcal{H}_s)\, ds$, assumed to satisfy $\Lambda_0(t) \to \infty$ as $t \to \infty$, and $\tilde{\Lambda}(\Delta t \mid \mathcal{H}_t) = \int_0^{\Delta t} \tilde{\lambda}(s \mid \mathcal{H}_t)\, ds$. Then $\{\tilde{\Lambda}(\tau_i \mid \mathcal{H}_{t_i})\}_{i=1}^N$ are i.i.d. exponential random variables with unit rate, where $\tau_i = t_i - t_{i-1}$ and $t_0 = 0$. Note that under the convention $\mathcal{H}_t = \{(t_j, k_j) : t_j < t\}$ we have $\mathcal{H}_{t_i} = \{(t_j, k_j)\}_{j=1}^{i-1}$, the history available for predicting the $i$-th event, which is what the encoding $\mathbf{h}_{i-1}$ represents.*

*Proof.* We know from Theorem 1 that $\Lambda_0(t_i) - \Lambda_0(t_{i-1})$ are i.i.d. Exp(1) random variables. Observe that:

$$\begin{aligned}
\Lambda_0(t_i) - \Lambda_0(t_{i-1}) &= \int_{t_{i-1}}^{t_i} \lambda_0(s \mid \mathcal{H}_s) ds && (6)\\
&= \int_0^{\tau_i} \lambda_0(s + t_{i-1} \mid \mathcal{H}_{s+t_{i-1}}) ds && (7)\\
&= \int_0^{\tau_i} \tilde{\lambda}(s \mid \mathcal{H}_{t_i}) ds && (8)\\
&= \tilde{\Lambda}(\tau_i \mid \mathcal{H}_{t_i}) && (9)
\end{aligned}$$

□

**Lemma 1** (Invertibility and Regularity)**.** *Assume $\lambda^*(\cdot \mid \mathcal{H}_\cdot)$ is continuous on each inter-event interval and $\lambda^*(t \mid \mathcal{H}_t) \geq \lambda_{\min} > 0$ for all $t \in \mathbb{R}^+$. Then $\Lambda^*$ is a strictly increasing continuous bijection from $\mathbb{R}^+$ to $\mathbb{R}^+$ that is $C^1$, with $C^1$ inverse, on each inter-event interval. If, in addition, $\lambda^*$ is $C^k$ there for some $k \geq 0$, then $\Lambda^*$ and $(\Lambda^*)^{-1}$ are $C^{k+1}$ there.*

*Proof.* As an integral of a locally bounded function, $\Lambda^*$ is continuous on $\mathbb{R}^+$; by the fundamental theorem of calculus it is $C^1$ on each inter-event interval with $(\Lambda^*)'(t) = \lambda^*(t \mid \mathcal{H}_t)$. The lower bound $\lambda^* \geq \lambda_{\min} > 0$ implies $\Lambda^*$ is strictly increasing (hence injective) and $\Lambda^*(t) \geq \lambda_{\min} t \to \infty$ as $t \to \infty$. Together with $\Lambda^*(0) = 0$ and continuity, this gives surjectivity onto $\mathbb{R}^+$, so $\Lambda^*$ is a continuous bijection. The Inverse Function Theorem applied at every point inside an inter-event interval (where $(\Lambda^*)' \neq 0$) yields a $C^1$ inverse there. If $\lambda^*$ is $C^k$ there, then $\Lambda^*$ is $C^{k+1}$ there by repeated differentiation, and $(\Lambda^*)^{-1}$ inherits this regularity via IFT. □

**Proposition** (Reverse-direction realization)**.** *Under the hypotheses of Lemma 1, let $\{z_i\}$ be a realization of a unit-rate Poisson process on $\mathbb{R}^+$. Then $\{t_i = (\Lambda^*)^{-1}(z_i)\}$ is a realization of the original point process with conditional intensity $\lambda^*$.*

*Proof.* Because $\lambda^*$ is history-dependent, $\Lambda^*$ is not a single fixed map on $\mathbb{R}^+$ but is determined recursively by the realization, so we construct the inverse sequentially rather than pushing a law through one bijection. Set $t_0 = 0$, $z_0 = 0$, and suppose $t_1 < \cdots < t_{i-1}$ have been defined, fixing the first $i-1$ events, denoted $\mathcal{F}_{i-1}$. On $(t_{i-1}, \infty)$ the intensity $s \mapsto \lambda^*(s \mid \mathcal{H}_s)$ is determined by $\mathcal{F}_{i-1}$ alone until the next event; writing it as $\lambda^*(s \mid \mathcal{F}_{i-1})$, $G_i(u) = \int_{t_{i-1}}^{t_{i-1}+u} \lambda^*(s \mid \mathcal{F}_{i-1})\, ds$ is a fixed $C^1$ bijection $\mathbb{R}^+ \to \mathbb{R}^+$ by Lemma 1. Define $t_i = t_{i-1} + G_i^{-1}(\Delta z_i)$ with $\Delta z_i = z_i - z_{i-1}$. Since the $\Delta z_i$ are i.i.d. $\mathrm{Exp}(1)$ and $\mathcal{F}_{i-1}$ is a function of $\Delta z_1, \ldots, \Delta z_{i-1}$, $\Delta z_i$ is independent of $\mathcal{F}_{i-1}$ and $\Pr(t_i > t \mid \mathcal{F}_{i-1}) = \Pr(\Delta z_i > G_i(t - t_{i-1})) = \exp(-[\Lambda^*(t) - \Lambda^*(t_{i-1})])$, which is exactly the conditional survival function of the $i$-th event under $\lambda^*$. The map $(\Delta z_1, \ldots, \Delta z_N) \mapsto (\tau_1, \ldots, \tau_N)$ is therefore lower-triangular and bimeasurable, and matching conditional laws at every $i$ identifies, by induction, the joint law of $\{t_i\}$ with that of the point process with conditional intensity $\lambda^*$. □

*Remark* 1. The lower bound $\lambda^* \geq \lambda_{\min} > 0$ — rather than mere positivity $\lambda^* > 0$ — is essential. Without it, $\Lambda^*$ can have a finite limit and fail to be surjective onto $\mathbb{R}^+$: e.g., $\lambda^*(t) = 1/(1+t)^2$ is strictly positive but $\Lambda^*(t) = 1 - 1/(1+t) \to 1$, in which case $\Lambda^*$ is a $C^1$ bijection only onto $[0, 1)$. In SurF, the floor $\lambda_{\text{floor}}$ enforces exactly the uniform lower bound required here (Appendix A.1).

This bijectivity is what makes SurF's loss and sampler well-defined: every learned $\Lambda_\theta$ is invertible, so sampling through $\Lambda_\theta^{-1}$ is exact up to the inversion tolerance $\epsilon_{\text{tol}} = 10^{-4}$.

### A.1 Near-Zero Intensity Regimes and Likelihood Consistency

The bijectivity in Theorem 2 requires $\lambda_{\min} > 0$. Real processes (neural refractoriness, market-closure windows, after-hours IoT) can exhibit regimes where the true intensity is effectively zero. We handle this structurally rather than with ad hoc clamping.

All three variants are structurally lower-bounded. SurF-MoE uses $\gamma_j = \mathrm{softplus}(\hat{\gamma}_j) + \varepsilon$ with $\varepsilon = 10^{-4}$; SurF-CSB and SurF-GLQ apply analogous $\mathrm{softplus} + \varepsilon$ to their positive parameters. Every variant additionally carries $\lambda_{\text{floor}}\Delta t$ in $\Lambda_\theta$ (equation 2), giving a hard floor $\lambda_{\text{floor}} > 0$ that ensures the conditions of Theorem 2.

Let $\lambda_\theta^\dagger = \lambda_\theta + \lambda_{\text{floor}}$. The floor raises the cumulative intensity by $\tau_i \lambda_{\text{floor}}$ per interval and simultaneously raises every log-intensity term, so the net likelihood bias is

$$b(\lambda_{\text{floor}}) = T\lambda_{\text{floor}} - \sum_{i=1}^{N} \log\Big(1 + \tfrac{\lambda_{\text{floor}}}{\lambda_\theta(\tau_i \mid \mathbf{h}_{i-1})}\Big) \;\leq\; T\lambda_{\text{floor}},$$

strictly below the stated bound whenever the intensity is finite, and vanishing as $\lambda_{\text{floor}} \to 0$, so the estimator is consistent in the limit. For $\lambda_{\text{floor}} = 10^{-4}$ below the NLL floor on all six benchmarks.

## B Likelihood Derivation via Change of Variables

The result is the standard TPP likelihood equation 1 (Guttorp, 1990); we include the derivation only to make explicit the triangular Jacobian underlying the SurF loss. We derive the likelihood for inter-arrival times $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_N)$ through the TRT transformation to exponential space $\boldsymbol{\Delta z} = (\Delta z_1, \ldots, \Delta z_N)$:

$$\Delta z_i = \Lambda_\theta(t_i) - \Lambda_\theta(t_{i-1}) = \int_{t_{i-1}}^{t_i} \lambda_\theta^\dagger(s \mid \mathcal{H}_s)\, ds, \tag{10}$$

where $t_i = \sum_{j=1}^{i} \tau_j$ are cumulative event times.

### B.1 Jacobian Analysis

The Jacobian $J = \partial \boldsymbol{\Delta z} / \partial \boldsymbol{\tau}$ is lower triangular in general, but its determinant depends only on the diagonal entries.

**General case (lower triangular).** Since $t_i = \sum_{j=1}^{i} \tau_j$, both limits of $\Delta z_i = \int_{t_{i-1}}^{t_i} \lambda_\theta^\dagger \, ds$ depend on $\tau_1, \ldots, \tau_i$, and the history $\mathcal{H}_s$ for $s \in (t_{i-1}, t_i]$ may depend on all earlier event times through the encoding. Hence

$$\frac{\partial \Delta z_i}{\partial \tau_j} = \begin{cases} \neq 0 & j \leq i, \\ 0 & j > i, \end{cases} \tag{11}$$

making $J$ lower triangular. By Leibniz's rule applied to the upper limit of integration,

$$\frac{\partial \Delta z_i}{\partial \tau_i} = \lambda_\theta^\dagger(t_i \mid \mathcal{H}_{t_i}), \tag{12}$$

regardless of how the history encoding is computed. The determinant of a lower-triangular matrix is the product of its diagonal entries, so

$$\det J = \prod_{i=1}^{N} \lambda_\theta^\dagger(t_i \mid \mathcal{H}_{t_i}). \tag{13}$$

Since $\lambda_\theta^\dagger \geq \lambda_{\text{floor}} > 0$ everywhere, $\det J > 0$ and the transformation is locally invertible.

**Under our architecture.** In SurF the encoding $\mathbf{h}_{i-1}$ is held fixed within $(t_{i-1}, t_i]$, so $\Delta z_i = \int_0^{\tau_i} \lambda_\theta^\dagger(u \mid \mathbf{h}_{i-1}) \, du = \Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1})$, whether $\Lambda_\theta$ is closed-form (MoE, CSB) or computed by quadrature (GLQ). Because $\mathbf{h}_{i-1}$ is computed by the encoder from $\tau_1, \ldots, \tau_{i-1}$, the entries $\partial \Delta z_i / \partial \tau_j$ for $j < i$ are in general nonzero, so $J$ is lower triangular, not diagonal. Since $\mathbf{h}_{i-1}$ does not depend on $\tau_i$, the diagonal entry is $\partial \Delta z_i / \partial \tau_i = \lambda_\theta^\dagger(\tau_i \mid \mathbf{h}_{i-1})$, and equation 13 holds unchanged, as do the change-of-variables likelihood below and the SurF loss.

### B.2 Change of Variables Formula

SurF *defines* its density by declaring the base law to be i.i.d. $\mathrm{Exp}(1)$ and pushing it through the map $\boldsymbol{\Delta z} \mapsto \boldsymbol{\tau}$ with components $\tau_i = \Lambda_\theta^{-1}(\Delta z_i \mid \mathbf{h}_{i-1})$. By Lemma 1, and because every variant carries the floor $\lambda_{\text{floor}} \Delta t$ in $\Lambda_\theta$ (Appendix A.1), this map is a $C^1$ diffeomorphism for every $\theta$, with inverse $\Delta z_i = \Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1})$. Change of variables for the transformation $\boldsymbol{\tau} \mapsto \boldsymbol{\Delta z}$ then gives

$$p_\theta(\boldsymbol{\tau}) = p_{\boldsymbol{\Delta z}}(\boldsymbol{\Delta z}(\boldsymbol{\tau})) \cdot |\det J|, \tag{14}$$

$$\log p_\theta(\boldsymbol{\tau}) = \log p_{\boldsymbol{\Delta z}}(\boldsymbol{\Delta z}) + \log |\det J|. \tag{15}$$

Because the base law is $\mathrm{Exp}(1)^{\otimes N}$ *by construction*, $\log p_{\boldsymbol{\Delta z}}(\boldsymbol{\Delta z}) = -\sum_i \Delta z_i$ holds identically in $\theta$, so

$$\log p_\theta(\boldsymbol{\tau}) = -\sum_{i=1}^{N} \Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1}) + \sum_{i=1}^{N} \log \lambda_\theta^\dagger(\tau_i \mid \mathbf{h}_{i-1}), \tag{16}$$

a valid log-density for every $\theta$. The direction of this argument matters: at an arbitrary $\theta$ the residuals $\Delta z_i = \Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1})$ computed on observed data are *not* $\mathrm{Exp}(1)$ — their departure from it is what the loss penalises and what Figure 2 measures. The step above holds by construction; the role of the TRT is to guarantee that the base is *attainable*, since by Corollary 1 the true cumulative intensity makes the pushforward exact. The $\mathrm{Exp}(1)$ reference therefore introduces no misspecification of its own, and it carries no dataset-specific constants.

### B.3 Observation Window Correction

For data observed in $[0, T]$, we include the survival probability of the final gap $[t_N, T]$:

$$\Pr(\text{no event in } (t_N, T]) = \exp\big(-\Lambda_\theta(\Delta T \mid \mathbf{h}_N)\big). \tag{17}$$

The complete negative log-likelihood is

$$\boxed{\mathcal{L}_{\text{SurF}}(\theta) = \sum_{i=1}^{N} \Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1}) + \Lambda_\theta(\Delta T \mid \mathbf{h}_N) - \sum_{i=1}^{N} \log \lambda^{\dagger}_\theta(\tau_i \mid \mathbf{h}_{i-1}).} \tag{18}$$

This is the unified amortized loss. The first two terms are forward passes of $\Lambda_\theta$; the third is computed via automatic differentiation (or closed form for MoE and CSB).

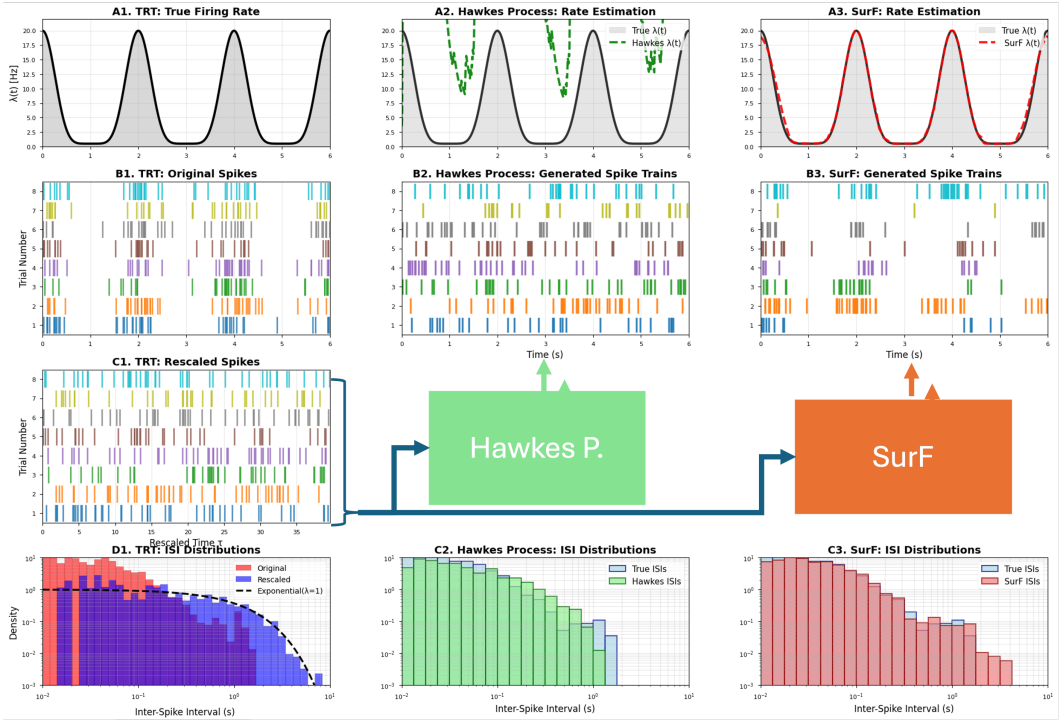


Figure 4: SurF recovers the true oscillatory intensity (A3) and ISI distribution (C3). The Hawkes fit (A2, C2), a classical parametric reference, cannot express this exogenous periodic rate.

## C Held-Out Negative Log-Likelihood

Table 5 reports per-event NLL of the joint inter-arrival/type density $p_\theta(\tau, k \mid \mathbf{h}_{i-1})$ for all three variants.

SurF-MoE attains the best NLL on four of six datasets (Taxi, StackOverflow, Earthquake, Taobao), consistent with its exponential basis matching their fast-decaying, near-Markovian dynamics. GLQ wins on the two heaviest-tailed datasets, Amazon (Joint, $-1.61$) and Retweet (Fine, $+2.55$), where unconstrained log-intensity flexibility pays off in the tail.

The jointly-trained checkpoint differs from the finetuned one by $0.06$ nats/event on average (mean absolute difference; net change $-0.01$), so joint training costs little in likelihood; the larger gaps, such as GLQ on Amazon, arise because checkpoints are selected on validation RMSE rather than NLL (Appendix I).

Table 5: Held-out per-event NLL of SurF variants (lower is better). **Bold** = best within row across all six configurations. "Joint" denotes the checkpoint trained on the union of all six training splits and evaluated on the target test set; it is *not* zero-shot. "Fine" denotes that checkpoint adapted per dataset. Strictly zero-shot numbers appear only under the leave-one-out protocol (Table 3).

| | MoE | | GLQ | | CSB | |
|---|---|---|---|---|---|---|
| **Dataset** | **Joint** | **Fine** | **Joint** | **Fine** | **Joint** | **Fine** |
| Taxi | $-0.9341$ | $\mathbf{-0.9503}$ | $-0.6914$ | $-0.7670$ | $-0.6407$ | $-0.6835$ |
| StackOverflow | $\mathbf{+0.4687}$ | $+0.4834$ | $+0.6423$ | $+0.5833$ | $+0.6969$ | $+0.7059$ |
| Retweet | $+2.5651$ | $+2.5557$ | $+2.6809$ | $\mathbf{+2.5502}$ | $+3.1440$ | $+3.1383$ |
| Earthquake | $-0.5986$ | $\mathbf{-0.6594}$ | $+0.0490$ | $-0.0257$ | $+0.1382$ | $+0.2134$ |
| Amazon | $-0.3520$ | $-0.3576$ | $\mathbf{-1.6062}$ | $-1.2922$ | $-0.9344$ | $-0.9326$ |
| Taobao | $\mathbf{-2.7897}$ | $-2.7765$ | $-2.5403$ | $-2.5011$ | $-1.5407$ | $-1.6960$ |

# D Amortized Event Intensity: Detailed Derivations

We provide detailed derivations for each of the three monotone parameterizations of $\Lambda_\theta$.

## D.1 SurF-CSB: Cumulative Softplus Basis

The cumulative intensity is parameterized as

$$\Lambda_\theta(\Delta t \mid \mathbf{h}) = \lambda_{\text{floor}}\,\Delta t + \sum_{m=1}^{M} \alpha_m(\mathbf{h})\big[\log(1 + e^{\beta_m(\mathbf{h})\Delta t + \delta_m(\mathbf{h})}) - \log(1 + e^{\delta_m(\mathbf{h})})\big], \tag{19}$$

with $\alpha_m = \text{softplus}(\hat{\alpha}_m(\mathbf{h})) > 0$ and $\beta_m = \text{softplus}(\hat{\beta}_m(\mathbf{h})) + \epsilon > 0$.

**Boundary condition.** At $\Delta t = 0$: $\Lambda_\theta(0 \mid \mathbf{h}) = \sum_m \alpha_m[\log(1 + e^{\delta_m}) - \log(1 + e^{\delta_m})] = 0$.

**Monotonicity.** The derivative is

$$\lambda_\theta^\dagger(\Delta t \mid \mathbf{h}) = \frac{\partial \Lambda_\theta}{\partial(\Delta t)} = \lambda_{\text{floor}} + \sum_{m=1}^{M} \alpha_m \beta_m\, \sigma(\beta_m \Delta t + \delta_m), \tag{20}$$

where $\sigma$ is the sigmoid. Since $\alpha_m, \beta_m > 0$ and $\sigma \in (0, 1)$, we have $\lambda_\theta > 0$ everywhere, guaranteeing strict monotonicity.

## D.2 SurF-GLQ: Gauss–Legendre Quadrature

The intensity is a free-form positive MLP,

$$\tilde{\lambda}_\theta(\Delta t \mid \mathbf{h}) = \text{softplus}(f_\theta(\Delta t, \mathbf{h})), \tag{21}$$

with $f_\theta : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ unconstrained. Here $\tilde{\lambda}_\theta$ is the unfloored integrand of equation 2; the floored intensity is $\lambda_\theta^\dagger = \lambda_{\text{floor}} + \tilde{\lambda}_\theta$. The cumulative is approximated by $Q$-point Gauss–Legendre quadrature on $[0, \tau_i]$:

$$\Lambda_\theta(\tau_i \mid \mathbf{h}) \approx \lambda_{\text{floor}}\,\tau_i + \frac{\tau_i}{2} \sum_{q=1}^{Q} w_q\, \tilde{\lambda}_\theta\Big(\tfrac{\tau_i}{2}(1 + x_q),\ \mathbf{h}\Big), \tag{22}$$

where $\{(x_q, w_q)\}_{q=1}^{Q}$ are the $Q$-point nodes and weights on $[-1, 1]$. The floor contributes the exact term $\lambda_{\text{floor}}\tau_i$ (a constant integrand is integrated exactly by any Gauss rule), so it adds nothing to the quadrature error analysed below.

The $Q$-point Gauss–Legendre error on $[0, \tau]$ is

$$\big|\Lambda_\theta(\tau) - \Lambda_\theta^{(Q)}(\tau)\big| \le \frac{(Q!)^4 \tau^{2Q+1}}{(2Q+1)[(2Q)!]^3} \sup_{u \in [0,\tau]} \big|\tfrac{d^{2Q}}{du^{2Q}} \tilde{\lambda}_\theta(u)\big|. \tag{23}$$

The factorially decaying prefactor makes this bound small already at $Q{=}8$ for the smooth integrand $\tilde{\lambda}_\theta$; the consequences for the parameter gradients are analysed in Appendix D.2.2. The quadrature cost $O(Q)$ is *per interval*, not per sequence.

#### D.2.1 GRADIENT COMPUTATION UNDER GAUSS–LEGENDRE QUADRATURE

Because SurF-GLQ is the variant whose likelihood involves a numerical step, we give an explicit derivation of how parameter gradients are computed.

The nodes $\{x_q\}_{q=1}^{Q}$ and weights $\{w_q\}_{q=1}^{Q}$ in equation 22 are the $Q$ roots of the Legendre polynomial $P_Q$ on $[-1, 1]$ and the associated Gauss weights — *data- and parameter-independent constants*, computed once at initialization and stored. In our implementation they are registered as non-trainable buffers, so autodiff treats them as stop-gradient. We do **not** backpropagate through the nodes or weights, and we use no differentiable-quadrature surrogate.

With this convention, differentiating equation 22 with respect to $\theta$ (with $\tau$, $\mathbf{h}$, and $(x_q, w_q)$ all independent of $\theta$) gives

$$\nabla_\theta \Lambda_\theta^{(Q)}(\tau \mid \mathbf{h}) \;=\; \frac{\tau}{2} \sum_{q=1}^{Q} w_q \, \nabla_\theta \, \tilde{\lambda}_\theta\Big(\tfrac{\tau}{2}(1 + x_q),\ \mathbf{h}\Big). \tag{24}$$

The gradient of the quadrature is the quadrature of the gradient, with the same nodes and weights. No derivatives of $x_q$ or $w_q$ appear.

#### D.2.2 QUADRATURE ERROR IN THE PARAMETER GRADIENT

Subtracting the exact parameter-gradient integral from equation 24, the residual $\nabla_\theta \Lambda_\theta^{(Q)} - \nabla_\theta \Lambda_\theta$ is *itself* a Gauss–Legendre quadrature error, applied to the integrand $g := \nabla_\theta \tilde{\lambda}_\theta$. The standard Gauss–Legendre error bound on $[0, \tau]$ for a $C^{2Q}$ integrand gives

$$\Big| \int_0^\tau g(u)\, du - \tfrac{\tau}{2} \sum_q w_q \, g\big(\tfrac{\tau}{2}(1 + x_q)\big) \Big| \;\le\; \frac{(Q!)^4 \, \tau^{2Q+1}}{(2Q+1)[(2Q)!]^3} \sup_{u \in [0,\tau]} |g^{(2Q)}(u)|. \tag{25}$$

Because $\tilde{\lambda}_\theta = \mathrm{softplus}(f_\theta)$ with $f_\theta$ a smooth MLP, $g$ and its higher derivatives exist and are bounded on the finite interval $[0, \tau]$. The claim "error does not enter the gradient" therefore holds under two conditions: (i) the integrand is sufficiently smooth (guaranteed by our architecture), and (ii) $Q$ is large enough that equation 25 is negligible at the scale of the optimization signal. For a smooth integrand the prefactor in equation 25 decays factorially in $Q$, so (ii) is already met at the modest $Q = 8$ we use throughout.

$\mathcal{L}_{\text{SurF}}^{(Q)}$ is a perturbation of $\mathcal{L}_{\text{SurF}}$ uniform in $\theta$ on any compact parameter set, with perturbation size bounded by equation 25 applied to $\tilde{\lambda}_\theta$ itself. Standard M-estimator theory (van der Vaart, 2000, Thm. 5.7) gives $\theta^{(Q)} \to \theta^\star$ as $Q \to \infty$ under the usual identifiability and equicontinuity conditions. At $Q = 8$ the size of this perturbation is controlled by equation 25; for SurF-MoE and SurF-CSB the question is moot since $\Lambda_\theta$ is closed-form.

Two scenarios re-introduce quadrature error into the gradient: (a) *learning* the quadrature nodes/weights, which would require $\partial/\partial\theta$ of $x_q, w_q$ (we do not do this); (b) using $Q$ too small for the smoothness of $f_\theta$, e.g. replacing the softplus-MLP with a non-smooth layer. Neither applies to SurF as described.

### D.3 SURF-MOE: MIXTURE-OF-EXPONENTIALS (CLOSED-FORM)

Between-event intensity is modeled as

$$\lambda_\theta(\Delta t \mid \mathbf{h}) = \sum_{j=1}^{J} w_j \, \exp(-\gamma_j \Delta t), \tag{26}$$

with $w_j = \mathrm{softplus}(\hat{w}_j(\mathbf{h})) > 0$ and $\gamma_j = \mathrm{softplus}(\hat{\gamma}_j(\mathbf{h})) + \epsilon > 0$. The cumulative admits an exact closed form,

$$\Lambda_\theta^{\text{MoE}}(\Delta t \mid \mathbf{h}) = \lambda_{\text{floor}} \, \Delta t + \sum_{j=1}^{J} \frac{w_j}{\gamma_j} \big(1 - e^{-\gamma_j \Delta t}\big). \tag{27}$$

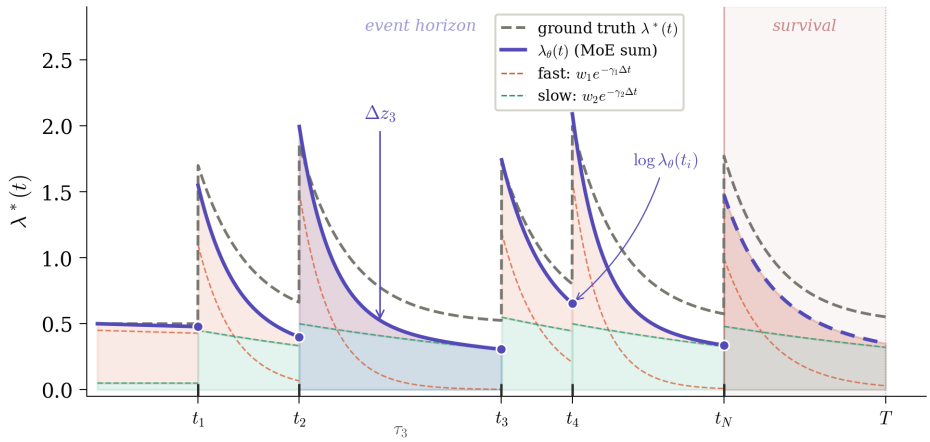


Figure 5: SurF-MoE on a toy example. Learned intensity $\lambda_\theta(t) = \sum_j w_j e^{-\gamma_j \Delta t}$ resets at each event as a sum of fast- (coral) and slow-decay (teal) components. The shaded region between $t_2$ and $t_3$ is $\Delta z_3$, one cumulative-intensity increment; dots mark $\log \lambda_\theta(t_i)$. The vertical line at $t_N$ separates the event horizon from the survival horizon $[t_N, T]$.

Each MoE component $(w_j/\gamma_j)(1 - e^{-\gamma_j \Delta t})$ is monotone and concave in $\Delta t$, so between two events $\lambda_\theta^{\text{MoE}}$ is a sum of decaying exponentials: a multi-scale excitation-and-relaxation bias that matches the fast-decaying dynamics of most of our benchmarks (MoE attains the lowest NLL on $4/6$, Table 5). Because $(w_j, \gamma_j)$ are re-emitted from $\mathbf{h}_{i-1}$ at every event, this shape constraint applies within an interval, not to the process-level intensity, which can rise and oscillate across events (Figure 4). CSB is the mirror case: with $\alpha_m, \beta_m > 0$ its derivative $\lambda_\theta^{\text{CSB}} = \sum_m \alpha_m \beta_m \sigma(\beta_m \Delta t + \delta_m)$ is strictly *increasing* in $\Delta t$, so CSB represents delayed excitation and saturating build-up. GLQ places no shape restriction on the between-event intensity. Summing the MoE and CSB bases, $\lambda_\theta = \sum_j w_j e^{-\gamma_j \Delta t} + \sum_m \alpha_m \beta_m \sigma(\beta_m \Delta t + \delta_m)$, gives a closed-form variant covering both directions, with cumulative intensity equal to the sum of the two closed forms.

### D.4 Closed-Form Loss under MoE

Under MoE all terms of $\mathcal{L}_{\text{SurF}}$ admit closed forms. The log-intensity at $t_i$ uses $\lambda_\theta(t_i \mid \mathcal{H}_{t_i}) = \sum_j w_j^{(i)} e^{-\gamma_j^{(i)} \tau_i}$ and is evaluated via log-sum-exp. The survival term is

$$\Lambda_\theta^{\text{MoE}}(\Delta T \mid \mathbf{h}_N) = \lambda_{\text{floor}}\, \Delta T + \sum_{j=1}^{J} \frac{w_j^{(N)}}{\gamma_j^{(N)}} \big(1 - e^{-\gamma_j^{(N)} \Delta T}\big). \tag{28}$$

Substituting into the unified loss gives

$$\mathcal{L}_{\text{SurF}}^{\text{MoE}}(\theta) = \sum_{i=1}^{N} \Big[ \lambda_{\text{floor}} \tau_i + \sum_{j=1}^{J} \frac{w_j^{(i)}}{\gamma_j^{(i)}} \big(1 - e^{-\gamma_j^{(i)} \tau_i}\big) \Big] + \lambda_{\text{floor}} \Delta T + \sum_{j=1}^{J} \frac{w_j^{(N)}}{\gamma_j^{(N)}} \big(1 - e^{-\gamma_j^{(N)} \Delta T}\big) - \sum_{i=1}^{N} \log\Big( \lambda_{\text{floor}} + \sum_{j=1}^{J} w_j^{(i)} e^{-\gamma_j^{(i)} \tau_i} \Big). \tag{29}$$

Under a piecewise-linear parameterization $\lambda_\theta(s) = \lambda_{i-1} + \alpha_{i-1}(s - t_{i-1})$, the integral is $\lambda_{i-1}\tau_i + \frac{1}{2}\alpha_{i-1}\tau_i^2$. While also closed-form, this form has three drawbacks: (i) positivity requires clamping $\max(\lambda_{\min}, \cdot)$, creating non-smooth gradients; (ii) the intensity is monotonic within each interval, preventing multimodal behavior; and (iii) $\log \lambda_i$ is unstable when $\lambda_i$ is near zero. MoE avoids (i) and (iii) structurally; GLQ additionally avoids (ii).

**Algorithm 1** SurF Training (Unified Amortized Framework)

**Input:** Training sequences $\{\mathcal{T}^{(k)}\}_{k=1}^{K}$ with observation windows $\{T^{(k)}\}_{k=1}^{K}$, learning rate $\eta$, iterations $M$, type weight $\beta_{\text{type}}$, variant $\in \{\text{MoE}, \text{CSB}, \text{GLQ}\}$, learned initial-history embedding $\mathbf{h}_0$
Initialize parameters $\theta$
**for** iteration $m = 1$ to $M$ **do**
  Sample mini-batch $\mathcal{B}$ from training sequences
  $\mathcal{L}_{\text{batch}} \leftarrow 0$
  **for** each sequence $\mathcal{T} = \{t_1, \ldots, t_N\}$ with window $T$, set $t_0 := 0$ **do**
    Encode: $\{\mathbf{h}_i\}_{i=1}^{N} \leftarrow \text{Transformer}_\theta(\{t_i, k_i\}_{i=1}^{N})$
    **for** $i = 1$ to $N$ **do**
      $\tau_i \leftarrow t_i - t_{i-1}$ $\{\tau_1 = t_1$ since $t_0 = 0\}$
      $\Delta z_i \leftarrow \Lambda_\theta(\tau_i \mid \mathbf{h}_{i-1})$ {forward pass; $\mathbf{h}_0$ used for $i$=1}
      $\lambda_i \leftarrow \lambda_\theta^\dagger(\tau_i \mid \mathbf{h}_{i-1})$ {autodiff or closed-form; $= \partial\Lambda_\theta/\partial(\Delta t)$}
    **end for**
    $\Delta T \leftarrow T - t_N$
    $\mathcal{L}_{\text{MLE}} \leftarrow \sum_{i=1}^{N} \Delta z_i + \Lambda_\theta(\Delta T \mid \mathbf{h}_N) - \sum_{i=1}^{N} \log \lambda_i$
    $\{\boldsymbol{\ell}_i\}_{i=0}^{N-1} \leftarrow \text{TypeNet}_\theta(\{\mathbf{h}_i\}_{i=0}^{N-1})$
    $\mathcal{L}_{\text{type}} \leftarrow \sum_{i=1}^{N} \text{CrossEntropy}(\boldsymbol{\ell}_{i-1}, k_i)$
    $\mathcal{L}_{\text{batch}} \leftarrow \mathcal{L}_{\text{batch}} + \mathcal{L}_{\text{MLE}} + \beta_{\text{type}}\mathcal{L}_{\text{type}}$
  **end for**
  $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}_{\text{batch}}$
**end for**

# E Sampling Algorithms

## E.1 Safeguarded Newton: Formal Convergence Guarantee

The sampling step of Algorithm 3 requires solving $\Lambda_\theta(\Delta t \mid \mathbf{h}) = z$ for $\Delta t > 0$ given $z \sim \text{Exp}(1)$. Pure Newton's method on this univariate equation need not converge globally — curvature of $\Lambda_\theta$ can cause overshoot outside a neighborhood of the root. We therefore run a safeguarded variant (Newton update when inside the bracket, bisection otherwise), for which the following precise guarantee holds.

**Proposition** (Global convergence of safeguarded Newton)**.** *Let $\Lambda_\theta(\cdot \mid \mathbf{h}) : \mathbb{R}^+ \to \mathbb{R}^+$ be $C^2$ and satisfy: (i) $\Lambda_\theta(0 \mid \mathbf{h}) = 0$, (ii) $\lambda_\theta^\dagger(\Delta t \mid \mathbf{h}) = \partial\Lambda_\theta/\partial(\Delta t) \geq \lambda_{\text{floor}} > 0$ for all $\Delta t \geq 0$, and (iii) $\Lambda_\theta(\Delta t \mid \mathbf{h}) \to \infty$ as $\Delta t \to \infty$. For any $z > 0$, let $[a_0, b_0]$ be a bracket with $\Lambda_\theta(a_0 \mid \mathbf{h}) \leq z \leq \Lambda_\theta(b_0 \mid \mathbf{h})$. Then the safeguarded Newton iteration, Newton update when the iterate lies in the current bracket, bisection otherwise, with the bracket tightened each step, converges to the unique root $\Delta t^\star$. Moreover:*

- ***Global (at least linear) rate.*** *Bisection halves the bracket on any step the Newton update is rejected, giving $|\Delta t^{(n)} - \Delta t^\star| \leq 2^{-n_B}(b_0 - a_0)$ where $n_B$ is the number of bisection steps.*
- ***Local quadratic rate.*** *Let $L$ bound $|\partial\lambda_\theta^\dagger/\partial(\Delta t)|$ on the current bracket. Once an iterate satisfies the Newton–Kantorovich condition $L\,|z - \Lambda_\theta(\Delta t^{(n)})|/\lambda_\theta^\dagger(\Delta t^{(n)})^2 \leq \frac{1}{2}$, the Newton iterates converge quadratically to $\Delta t^\star$ (Nocedal & Wright, 2006, Ch. 11).*

*Sketch.* Existence and uniqueness of $\Delta t^\star$ follow from Theorem 2 under the stated hypotheses. Bisection convergence is standard. The Newton step is accepted iff it lies inside the current bracket, which remains a valid bracket throughout by construction. The local quadratic rate is the standard Newton–Kantorovich theorem applied to the smooth univariate residual $r(\Delta t) = \Lambda_\theta(\Delta t \mid \mathbf{h}) - z$: hypothesis (ii) gives $r' = \lambda_\theta^\dagger \geq \lambda_{\text{floor}} > 0$, and $C^2$ smoothness makes $r''$ bounded on the compact bracket, so $L < \infty$. □

**Bracket initialization.** We set $a_0 = \epsilon = 10^{-6}$ and $b_0 = 2z/\lambda_\theta^\dagger(0 \mid \mathbf{h})$ (twice the constant-intensity estimate); if $\Lambda_\theta(\epsilon \mid \mathbf{h}) > z$, the root lies in $(0, \epsilon)$ and the iterates approach $\epsilon$, an error below

**Algorithm 2** SurF Sampling via Inverse Time Rescaling (Safeguarded Newton, General)

**Input:** Trained parameters $\theta$, horizon $T$, initial type $k_1$, variant $\in \{\text{MoE}, \text{CSB}, \text{GLQ}\}$, tolerance $\epsilon_{\text{tol}} = 10^{-4}$, max iterations $M$, floor $\epsilon = 10^{-6}$
Initialize: $t \leftarrow 0$, $\mathcal{T} \leftarrow \emptyset$, $\mathcal{H}_t \leftarrow \emptyset$
**while** $t < T$ **do**
  $\Delta z \sim \text{Exponential}(1)$
  $\mathbf{h} \leftarrow \text{Transformer}_\theta(\mathcal{H}_t)$
  **Initialize bracket:** $a \leftarrow \epsilon$, $b \leftarrow 2\Delta z / \lambda^\dagger_\theta(0 \mid \mathbf{h})$
  **while** $\Lambda_\theta(b \mid \mathbf{h}) < \Delta z$ **do**
    $b \leftarrow 2b$
  **end while**
  $\Delta t \leftarrow (a + b)/2$
  **for** $n = 1$ to $M$ **do**
    $F \leftarrow \Lambda_\theta(\Delta t \mid \mathbf{h})$; $F' \leftarrow \lambda^\dagger_\theta(\Delta t \mid \mathbf{h})$
    **if** $|F - \Delta z| < \epsilon_{\text{tol}}$ **then**
      **break**
    **end if**
    $\Delta t_{\text{N}} \leftarrow \Delta t - (F - \Delta z)/\max(F', \epsilon)$
    **if** $\Delta t_{\text{N}} \in (a, b)$ **then**
      $\Delta t \leftarrow \Delta t_{\text{N}}$
    **else**
      $\Delta t \leftarrow (a + b)/2$ {bisection fallback}
    **end if**
    **if** $\Lambda_\theta(\Delta t \mid \mathbf{h}) < \Delta z$ **then**
      $a \leftarrow \Delta t$
    **else**
      $b \leftarrow \Delta t$
    **end if**
  **end for**
  $t_{\text{new}} \leftarrow t + \Delta t$
  **if** $t_{\text{new}} > T$ **then**
    **break**
  **end if**
  $k_{\text{new}} \sim \text{Categorical}(\text{softmax}(\text{TypeNet}_\theta(\mathbf{h})))$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(t_{\text{new}}, k_{\text{new}})\}$; $\mathcal{H}_t \leftarrow \mathcal{H}_t \cup \{(t_{\text{new}}, k_{\text{new}})\}$; $t \leftarrow t_{\text{new}}$
**end while**
**Return** $\mathcal{T}$

$10^{-6}$. If $\Lambda_\theta(b_0 \mid \mathbf{h}) < z$, we double $b_0$ until the upper bracket is reached; termination is guaranteed in at most $\lceil \log_2(z/(b_0\,\lambda_{\text{floor}})) \rceil$ doublings, since hypothesis (ii) implies $\Lambda_\theta(b) \geq b\,\lambda_{\text{floor}}$. Without the bracket safeguard, Newton can overshoot in regimes where $\lambda^\dagger_\theta$ changes rapidly relative to its magnitude — concretely, when $|\partial_{\Delta t}\lambda^\dagger_\theta(\Delta t^{(0)})|\,|z - \Lambda_\theta(\Delta t^{(0)})|/\lambda^\dagger_\theta(\Delta t^{(0)})^2 \gtrsim 1$. This arises in practice at the tail of a short-timescale MoE component shortly after an event; the safeguard catches exactly these cases. Across all six benchmarks, $M = 8$ iterations suffice to reach tolerance $10^{-4}$ in $> 99.9\%$ of inversions; bisection is triggered only on inputs near numerical underflow of $e^{-\gamma_j \Delta t}$, where it activates automatically and yields linear convergence on the final iteration. The total wall-clock fraction spent in Newton during sampling is under $1.2\%$ of Transformer encoder time on every dataset, consistent with the measured cost in Table 4.

## F Extended Hyperparameter Sensitivity Study

This appendix supplements the ablations of Section 4.4 with the full per-dataset breakdown of the conformer/loss-schedule ablation (Table 6) and the full results of a 100-trial hyperparameter sweep over SurF-GLQ (Tables 7 and 8).

**Algorithm 3** SurF-MoE Sampling (Safeguarded Newton, Closed-Form Specialization)

**Input:** Trained parameters $\theta$, horizon $T$, initial type $k_1$, intensity floor $\lambda_{\text{floor}}$, tolerance $\epsilon_{\text{tol}} = 10^{-4}$, max iterations $M = 8$, numerical guard $\epsilon = 10^{-6}$
Initialize: $t \leftarrow 0$, $\mathcal{T} \leftarrow \emptyset$, $\mathcal{H}_t \leftarrow \emptyset$
**while** $t < T$ **do**
  **if** $\mathcal{H}_t = \emptyset$ **then**
    use initial embedding for type $k_1$
  **else**
    $\mathbf{h} \leftarrow \text{Transformer}_\theta(\mathcal{H}_t)$
  **end if**
  $\{w_j\} \leftarrow \text{softplus}(\hat{w}_j(\mathbf{h}))$, $\{\gamma_j\} \leftarrow \text{softplus}(\hat{\gamma}_j(\mathbf{h})) + \epsilon$
  $\{\Lambda_\theta(\Delta t \mid \mathbf{h}) = \lambda_{\text{floor}}\Delta t + \sum_j (w_j/\gamma_j)(1 - e^{-\gamma_j \Delta t}),\ \lambda_\theta^\dagger = \lambda_{\text{floor}} + \sum_j w_j e^{-\gamma_j \Delta t}\}$
  $U \sim \text{Uniform}(0, 1)$; $z \leftarrow -\log U$
  **Initialize bracket:** $a \leftarrow \epsilon$, $b \leftarrow 2z/(\lambda_{\text{floor}} + \sum_j w_j)$
  **while** $\lambda_{\text{floor}}\, b + \sum_j (w_j/\gamma_j)(1 - e^{-\gamma_j b}) < z$ **do**
    $b \leftarrow 2b$ {expand upper bracket; terminates by Prop. E.1}
  **end while**
  $\Delta t \leftarrow (a + b)/2$ {initial guess}
  **for** $n = 1$ to $M$ **do**
    $F \leftarrow \lambda_{\text{floor}}\Delta t + \sum_j (w_j/\gamma_j)(1 - e^{-\gamma_j \Delta t})$; $F' \leftarrow \lambda_{\text{floor}} + \sum_j w_j\, e^{-\gamma_j \Delta t}$
    **if** $|F - z| < \epsilon_{\text{tol}}$ **then**
      **break** {converged}
    **end if**
    $\Delta t_{\text{N}} \leftarrow \Delta t - (F - z)/F'$ {candidate Newton step} [1]
    **if** $\Delta t_{\text{N}} \in (a, b)$ **then**
      $\Delta t \leftarrow \Delta t_{\text{N}}$ {accept Newton}
    **else**
      $\Delta t \leftarrow (a + b)/2$ {bisection fallback}
    **end if**
    $F_{\text{new}} \leftarrow \lambda_{\text{floor}}\Delta t + \sum_j (w_j/\gamma_j)(1 - e^{-\gamma_j \Delta t})$
    **if** $F_{\text{new}} < z$ **then**
      $a \leftarrow \Delta t$
    **else**
      $b \leftarrow \Delta t$
    **end if**{tighten bracket; sign of $F_{\text{new}} - z$ preserves invariant}
  **end for**
  $t_{\text{new}} \leftarrow t + \Delta t$
  **if** $t_{\text{new}} > T$ **then**
    **break**
  **end if**
  $k_{\text{new}} \sim \text{Categorical}(\text{softmax}(\text{TypeNet}_\theta(\mathbf{h})))$
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{(t_{\text{new}}, k_{\text{new}})\}$; $\mathcal{H}_t \leftarrow \mathcal{H}_t \cup \{(t_{\text{new}}, k_{\text{new}})\}$; $t \leftarrow t_{\text{new}}$
**end while**
**Return** $\mathcal{T}$

Table 6 reports time RMSE / type accuracy on each of the six benchmarks under each tested combination of conformer toggle and loss schedule.

The sweep was run on the Taobao validation split and is scored in raw time units, so its RMSE scale differs from Table 2; only comparisons within the sweep are meaningful. It jointly varied: number of Gauss–Legendre points $Q \in \{4, 8, 12\}$; number of log-intensity references $J \in \{3, 4, 5\}$; conformer pre-encoder (on/off); batch size $\in \{128, 256, 512\}$; learning rate $\in [3{\times}10^{-5},\ 8{\times}10^{-4}]$; and pretraining type-loss weight $\beta_{\text{type}} \in [0.51,\ 4.28]$.

**Sampling design.** Because the TPE sampler concentrates trials in promising regions, the sampled factors co-vary: trials fall into a few joint configurations (group sizes 9, 69 and 22). All $Q$=4 trials

Table 6: Conformer and loss-schedule ablation on SURF-MOE (time RMSE / type accuracy %). **Bold** = best per column; the **Hybrid** (full) row is the configuration reported as SURF-MOE (fine) in Table 2.

| Conformer | Schedule | Amazon | Retweet | Taxi | Earthquake | StackOverflow | Taobao |
|---|---|---|---|---|---|---|---|
| ✗ | Equal | 0.343 / 34.6 | 16.60 / 59.7 | 0.354 / 89.1 | 1.289 / **47.4** | **1.039** / 45.1 | **0.131** / 59.7 |
| ✗ | Staged | **0.330** / 34.4 | 22.33 / **60.0** | 0.304 / **91.6** | 1.556 / 47.2 | 1.054 / **45.3** | 0.136 / 57.9 |
| ✓ | Equal | 0.343 / **34.8** | 16.58 / 59.7 | 0.479 / 89.6 | 1.309 / 47.3 | 1.147 / 45.0 | 0.236 / **60.3** |
| ✓ | Staged | 0.349 / **34.8** | 22.33 / 59.3 | **0.283** / **91.6** | **1.224** / **47.4** | 1.049 / **45.3** | **0.131** / 59.8 |
| ✓ | **Hybrid** (full) | 0.336 / 34.7 | **16.23** / 59.7 | 0.311 / 89.5 | 1.258 / 47.0 | 1.077 / 45.1 | 0.189 / **60.3** |

use batch 128 without the conformer, all $Q{=}8$ trials use batch 256 with it, and $Q{=}12$ trials use batch 512 with either setting; the $J$ and $\beta_{\text{type}}$ groupings overlap similarly. We therefore read the groupings below as joint configurations and use the sweep as a robustness check rather than a per-factor ablation. The $Q{=}8$ default is supported by the Gauss–Legendre error analysis of Appendix D.2.2.

**Quadrature points and mixture size.** Table 7 aggregates the sweep by $Q$ and $J$; the $Q$ rows correspond to the three joint configurations described above.

Table 7: SurF-GLQ 100-trial hyperparameter sweep grouped by the number of Gauss–Legendre points $Q$ and log-intensity reference components $J$. We report the best, median, and trial-level standard deviation of time RMSE, and the best type accuracy within each group. **Bold** = best per metric. Groups co-vary with batch size and the conformer (see text).

| Param | Value | $n$ | RMSE (best) | RMSE (med) | $\sigma_{\text{RMSE}}$ | Acc % (best) |
|---|---|---|---|---|---|---|
| $Q$ | 4 | 9 | 8.751 | 8.825 | 0.170 | 48.96 |
| | **8** (default) | 69 | 8.729 | **8.749** | 0.058 | **49.94** |
| | 12 | 22 | **8.725** | 8.786 | 0.035 | 49.49 |
| $J$ | 3 | 38 | **8.725** | 8.771 | 0.069 | 49.51 |
| | **4** (default) | 56 | 8.729 | **8.748** | 0.053 | **49.94** |
| | 5 | 6 | 8.751 | 8.815 | 0.082 | 48.88 |

Table 8 bins the trials by type-loss weight $\beta_{\text{type}}$ (same sampling design). The best trial in every bin lies within 0.03 RMSE of the overall best, so performance is robust to $\beta_{\text{type}}$ across the searched range. The default pretraining weight $\beta_{\text{type}}{=}0.74$ used in Section 4 lies in the bin containing the best-accuracy trial.

Table 8: Loss-weight sensitivity on the SurF-GLQ sweep (same sampling design as Table 7).

| Bin | $n$ | RMSE (best) | RMSE (mean) | Acc % (best) |
|---|---|---|---|---|
| *Type weight* $\beta_{type}$ | | | | |
| $< 0.7$ | 38 | **8.725** | 8.786 | 49.51 |
| 0.7–1 | 46 | 8.729 | **8.766** | **49.94** |
| 1–2 | 10 | 8.744 | 8.858 | 49.87 |
| $> 2$ | 6 | 8.751 | 8.801 | 49.49 |

**Top configurations.** The top-10 trials by time RMSE all come from $Q{\in}\{8, 12\}$ (60% $Q{=}8$, 40% $Q{=}12$) and $J{\in}\{3, 4\}$, span both conformer settings (60% on, 40% off), and use either batch 256 or 512. None of the top-10 trials comes from the $Q{=}4$/batch-128/no-conformer configuration or uses $J{=}5$. The fact that several near-optimal HP regions exist suggests that training with equation 3 is not fragile to the exact configuration.

# G Additional Theoretical Material

## G.1 Joint Modeling of Times and Marks

For $K$ event types, SurF factorizes the joint conditional intensity as

$$\lambda^*(t, k \mid \mathcal{H}_t) = \lambda^*(t \mid \mathcal{H}_t) \cdot p(k \mid t, \mathcal{H}_t), \qquad \sum_{k=1}^{K} p(k \mid t, \mathcal{H}_t) = 1, \tag{30}$$

where $\lambda^*(t \mid \mathcal{H}_t)$ is the marginal ground intensity (modeled by $\Lambda_\theta$) and $p(k \mid t, \mathcal{H}_t)$ is the conditional mark distribution (modeled by $\text{TypeNet}_\theta$ with a softmax head). $\text{TypeNet}_\theta$ conditions on $\mathbf{h}_{i-1}$ only (Algorithm 1), i.e. $p(k \mid t, \mathcal{H}_t) = p(k \mid \mathcal{H}_t)$, so unlike DTPP it never sees the ground-truth event time. In joint training the types of the six corpora are mapped to disjoint global IDs (75 in total), one tied head scores all 75, and the logits are masked to the current corpus's types. A held-out corpus's types are therefore never seen, which is why zero-shot type accuracy is undefined. Substituting into the point-process likelihood, the joint NLL decomposes exactly:

$$-\log p(\mathcal{D}) = \underbrace{\mathcal{L}_{\text{SurF}}(\theta)}_{\text{time}} + \underbrace{\sum_{i=1}^{N} -\log p_\theta(k_i \mid t_i, \mathcal{H}_{t_i})}_{\mathcal{L}_{\text{type}}(\theta)}. \tag{31}$$

No variational bound or coupling term is required; the factorization is exact at the likelihood level. The weight $\beta_{\text{type}}$ in Algorithm 1 is introduced purely to balance the relative scales of the two NLL components during optimization, not as a variational coefficient.

Because $\lambda^*(t, k) = \lambda^*(t)\, p(k \mid t)$ is a strict product, the TRT applied to the ground intensity $\lambda^*(t) = \sum_k \lambda^*(t, k)$ yields the same $\text{Exp}(1)$ rescaling regardless of $K$, so Theorem 2 holds unchanged and Proposition G.4 is unaffected by the number of types. Marks are generated conditionally at sampling time (Algorithm 3).

A per-type formulation $\{\lambda_k^*(t)\}_{k=1}^K$ with separate cumulative intensities is also supported by letting the cumulative-intensity head output $K$ copies of the parameters. The TRT then applies per-type, giving $K$ independent $\text{Exp}(1)$ rescalings; the joint likelihood equals the sum of $K$ per-type instances of the unified loss.

## G.2 Relationship to Prior Cumulative-Hazard Neural TPPs

Parameterizing the cumulative hazard $\Lambda_\theta$ directly and recovering $\lambda_\theta$ via differentiation was introduced by Omi et al. (2020) in *FullyNN*, where $\Lambda_\theta$ is a feed-forward network with non-negative weights, and Shchur et al. (2019) use a log-normal mixture with a related closed-form cumulative intensity. SurF differs along three axes.

**(i) Conceptual framing.** FullyNN uses the monotone network to avoid the $\int_0^T \lambda\, ds$ integral. Using $\Lambda_\theta$ as a bijection between the point process and i.i.d. $\text{Exp}(1)$ noise in both directions is the classical inverse method, made explicit for neural TPPs by the triangular-map view of Shchur et al. (2020). What SurF adds is to use the shared $\text{Exp}(1)$ target for multi-corpus pretraining, whereas FullyNN and its descendants are trained per dataset.

**(ii) Parameterization.** FullyNN enforces monotonicity via non-negative-weight MLPs, which are ill-conditioned at depth. SurF-CSB uses a positive combination of shifted softplus basis functions that admits closed-form derivatives. SurF-MoE uses a mixture of exponentials with a fully closed-form loss; SurF-GLQ provides an unconstrained-MLP alternative with per-interval quadrature.

**(iii) Amortization.** FullyNN's hazard depends on history through an RNN hidden state shared across the monotone MLP input. SurF amortizes via a Transformer encoder whose output $\mathbf{h}_{i-1}$ parameterizes a fresh $\Lambda_\theta(\cdot \mid \mathbf{h}_{i-1})$ at each interval, yielding the lower-triangular Jacobian with closed-form diagonal of Appendix B rather than backpropagation through a recurrent hidden state. Table 2 includes FullyNN as a baseline; SurF outperforms it on all six benchmarks.

### G.3 When to Use Which Variant, and the Role of Quadrature

The three variants target different regimes and differ in how (if at all) they invoke quadrature.

**SurF-MoE — closed-form, fastest of the three.** Between-event intensity is monotone-decaying and multi-scale. Cumulative intensity and derivative are exact closed forms; the loss invokes zero quadrature. This is the default for multi-corpus pretraining.

**SurF-CSB — closed-form.** Between-event intensity is monotone-increasing (delayed excitation, saturating build-up); the shifted-softplus basis keeps both $\Lambda_\theta$ and $\lambda_\theta$ in closed form.

**SurF-GLQ with fixed-cost per-interval quadrature.** The intensity is an unconstrained positive MLP; $\Lambda_\theta$ uses $Q = 8$ Gauss–Legendre nodes *per inter-event interval* $[0, \tau_i]$, not over the full window $[0, T]$.

### G.4 Shared Canonical Target

**Proposition** (Shared canonical target)**.** *Let $\mathcal{P}_1, \mathcal{P}_2$ be point processes arising from potentially different datasets with intensities $\lambda_1^*, \lambda_2^*$, each satisfying the hypotheses of Theorem 2, and let $(\phi_\theta, \Lambda_\theta)$ be trained jointly on the mixture $\pi_1\mathcal{P}_1 + \pi_2\mathcal{P}_2$ with $\pi_d > 0$ by minimizing the population SurF loss. Assume realizability: some $\theta^\star$ satisfies $\Lambda_{\theta^\star}(\tau \mid \phi_{\theta^\star}(\mathcal{H})) = \Lambda_d^*(\tau \mid \mathcal{H})$ for $\mathcal{P}_d$-almost every history $\mathcal{H}$ and both $d$. Then every population minimizer agrees with $\Lambda_d^*$ $\mathcal{P}_d$-almost everywhere for both $d$, and the rescaled gaps $\Lambda_\theta(\tau_i \mid \phi_\theta(\mathcal{H}_{t_i}))$ are i.i.d. $\mathrm{Exp}(1)$ on each dataset. The two datasets therefore share one target, and the loss couples the encoder through it even though its $(t_i, k_i)$ inputs remain dataset-specific.*

*Proof.* By Appendix B, $\mathcal{L}_{\text{SurF}}(\theta)$ evaluated on a sequence drawn from $\mathcal{P}_d$ is the negative log-likelihood of that sequence under the model, so writing $p_d^*$ for the true inter-arrival law of $\mathcal{P}_d$ and $p_\theta$ for the model's, the population objective is

$$R(\theta) = \sum_{d=1,2} \pi_d \, \mathbb{E}_{p_d^*}\big[-\log p_\theta\big] = \sum_{d=1,2} \pi_d \, \mathrm{KL}\big(p_d^* \,\|\, p_\theta\big) + c, \tag{32}$$

where $c$ collects the entropies of $p_1^*$ and $p_2^*$ and does not depend on $\theta$. Each summand is minimized exactly when $\mathrm{KL}(p_d^* \| p_\theta) = 0$, i.e. $p_\theta = p_d^*$ $\mathcal{P}_d$-almost everywhere, and realizability makes the two minima simultaneously attainable; since $\pi_d > 0$, $R$ is therefore minimized precisely at those $\theta$ that match both. A point process law determines its cumulative intensity and conversely, and $\lambda_\theta^\dagger = \partial\Lambda_\theta / \partial(\Delta t)$ by equation 2, so this is equivalent to $\Lambda_\theta(\cdot \mid \phi_\theta(\cdot)) = \Lambda_d^*$ $\mathcal{P}_d$-almost everywhere. Applying Corollary 1 to each $\mathcal{P}_d$ gives the $\mathrm{Exp}(1)$ conclusion. □

*Remark* 2 (Scope). The proposition is standard maximum-likelihood consistency applied to a mixture; we state it only to make explicit that, under realizability, pooling corpora does not trade one corpus's $\mathrm{Exp}(1)$ target against another's. It concerns corpora in the training mixture; transfer to unseen corpora is evaluated empirically (Section 4.3, Appendix J.1).

## H Datasets

We evaluate SurF on six real-world datasets spanning e-commerce, transportation, social media, geoscience, and online collaboration:

- **Taobao** (Xue et al., 2022): user click sequences from a large-scale online recommendation platform, capturing temporal patterns of user engagement with product categories.
- **Taxi** (Whong, 2014): taxi pick-up events across New York City neighborhoods, providing insights into urban mobility patterns.
- **Retweet** (Zhou et al., 2013): social cascades of retweet behaviors and user identities over time.
- **StackOverflow** (Leskovec & Krevl, 2014): activity logs from a question-and-answer platform with events representing posting and voting interactions.

Table 9: Pre-training corpus composition. The corpus spans 6 datasets with diverse temporal scales, totaling $\sim$1.3M events across $\sim$32K sequences.

| Dataset | Sequences | Events | Event Types | Avg. Length | Native units / span |
|---|---|---|---|---|---|
| Taxi | 2,000 | 74,078 | 10 | 37.04 | minutes–hours |
| Earthquake | 4,300 | 70,723 | 7 | 16.46 | hours–days |
| StackOverflow | 2,203 | 142,777 | 22 | 64.81 | hours–days |
| Amazon | 9,227 | 413,420 | 16 | 44.81 | hours–days |
| Retweet | 12,055 | 493,708 | 3 | 40.95 | seconds–hours |
| Taobao | 2,000 | 115,397 | 17 | 57.70 | hours–days |
| **Total** | **31,785** | **1,310,103** | 75 | **41.22** | seconds–days |

- **Amazon** (Ni et al., 2019): product review sequences where each event is a user reviewing a product in a specific category.
- **Earthquake** (Xue et al., 2024): seismic event sequences with magnitude and location codes, covering multi-day inter-event gaps from regional geological catalogs.

# I Model Architecture and Hyperparameters

The model has two components: a history encoder $\phi_\theta : \mathbb{R}^+ \times \mathcal{H} \to \mathbb{R}^d$ producing the encoding $\mathbf{h}_{i-1}$, and a cumulative-intensity head that maps $\mathbf{h}_{i-1}$ to $\Lambda_\theta(\cdot \mid \mathbf{h}_{i-1})$ (Appendix I.2).

## I.1 History Encoding

A multi-layer Transformer with time-aware embeddings:

$$\phi_\theta(t, \mathcal{H}_t) = \text{TransformerEncoder}_\theta(\text{Conformer}_\xi(\mathbf{E}_{\text{combined}}, \mathbf{M})), \tag{33}$$

$$\mathbf{E}_{\text{combined}} = [\mathbf{E}_{\text{type}}; \mathbf{E}_{\text{time}}] \in \mathbb{R}^{N \times 2d_{\text{hidden}}}. \tag{34}$$

Here $\mathbf{M}$ is the attention mask and $\text{Conformer}_\xi$ is an optional convolution-augmented block applied before the Transformer layers (toggled in the ablation of Section 4.4). Type embeddings: $\mathbf{E}^{(i)}_{\text{type}} = \text{Embedding}(k_i + 1) \in \mathbb{R}^{d_{\text{hidden}}}$ (the offset ensures valid indices).

## I.2 Cumulative-Intensity Head

For the amortized variants, the Transformer output $\mathbf{h}_{i-1}$ feeds a cumulative-intensity head that produces the parameters of $\Lambda_\theta(\cdot \mid \mathbf{h}_{i-1})$.

**SurF-CSB head.** A single linear layer maps $\mathbf{h}_{i-1}$ to $3M$ raw parameters $(\hat{\alpha}_m, \hat{\beta}_m, \hat{\delta}_m)_{m=1}^M$, with $\alpha_m = \text{softplus}(\hat{\alpha}_m)$, $\beta_m = \text{softplus}(\hat{\beta}_m) + \epsilon$. The $\delta_m$ are unconstrained, initialized uniformly in $[-3, 3]$ to span early-to-late activation.

**SurF-GLQ head.** A two-layer MLP takes $(\Delta t, \mathbf{h}_{i-1})$ and outputs a scalar, $f_\theta(\Delta t, \mathbf{h}) = \text{MLP}([\Delta t/T; \mathbf{h}])$. Intensity: $\tilde{\lambda}_\theta = \text{softplus}(f_\theta)$; cumulative via the Gauss–Legendre quadrature of equation 22. The MLP is lightweight (two layers, $d_{\text{hidden}}/2$ width), and $Q$ passes are batched.

**SurF-MoE head.** A single linear layer maps $\mathbf{h}_{i-1}$ to $2J$ raw parameters $(\hat{w}_j, \hat{\gamma}_j)_{j=1}^J$.

## I.3 Type Prediction Head

The output from the history encoding is passed through a Type Refinement Tower consisting of 3-4 Transformer encoder layers to produce a type context. This context is passed through an MLP before being projected to the type embedding space, where logits over the 75 global type IDs are masked to the current corpus's types (Appendix G.1).

$$\text{TypeNet}_\theta(\mathbf{h}) = \text{Proj}(\text{MLP}_{\text{type}}(\text{TypeRefinementTower}(\phi_\theta(t, \mathcal{H}_t)))) \tag{35}$$

### I.4 Hyperparameter Search

We optimize the time RMSE on the validation split (the test split is used only for the final reported numbers),

$$\mathcal{L}_{\text{hyperopt}} = \text{RMSE}_{\text{time}} \tag{36}$$

since we found that type accuracy is not heavily impacted by small changes in hyperparameters. We use Optuna with TPE sampling and median pruning for early termination of unpromising trials (patience 10 epochs). Checkpoints are likewise selected by validation RMSE.

**Searched:** $d_{\text{hidden}} \in \{32, 64, 128, 256, 512\}$, layers $L \in \{2, 3, 4\}$, batch size $B \in \{128, 256, 512\}$, learning rate $\eta \in [10^{-5}, 10^{-3}]$ (log-uniform), basis components $M \in \{8, 12, 16\}$ (CSB) or quadrature points $Q \in \{4, 8, 12\}$ (GLQ).

**Fixed:** $H = 4$ attention heads, dropout 0.1, weight decay $10^{-5}$, cosine annealing with warm restarts, gradient accumulation 1. We require $d_{\text{hidden}} \equiv 0 \pmod{H}$.

## J Additional Empirical Analysis

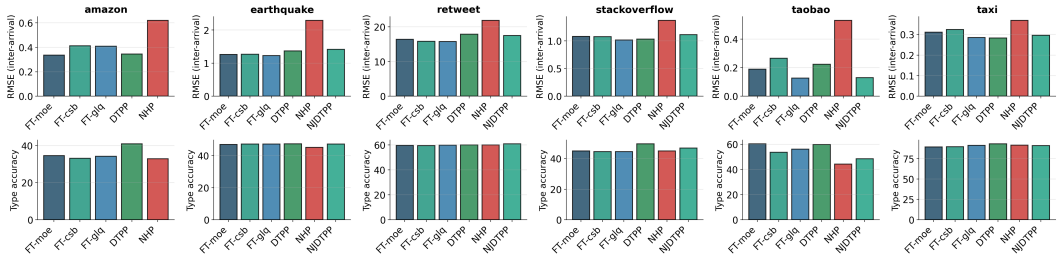


Figure 6: Next-event inter-arrival RMSE (top) and type accuracy (bottom) for finetuned SurF variants and three baseline models (DTPP, NHP, NJDTPP). NJDTPP has no Amazon result (Table 2) and is omitted there.

### J.1 Leave-One-Out Transfer Against In-Corpus Baselines

We compare leave-one-out checkpoints mainly in absolute time RMSE (Table 3). A ratio to a single in-corpus RMSE depends on which in-corpus model serves as the reference (per-dataset, joint or finetuned) and rewards a weak reference: AttNHP's joint RMSE on Amazon is 1.508, against 0.336 for SurF-MoE, which by itself makes its leave-one-out ratio small. The ratio reported in Section 4.3 therefore uses each model's best in-corpus RMSE as the reference, so it cannot be lowered by choosing a weak in-corpus reference.

In absolute terms, the best held-out SurF checkpoint beats every leave-one-out baseline on $5/6$ datasets and is within 0.032 of DTPP on StackOverflow, while baselines trained on the same five corpora can collapse on the held-out one (on Amazon, DTPP reaches 1.135 and AttNHP 1.975, against 0.330 for SurF-CSB). On Amazon (0.330 vs. 0.332), Earthquake (1.229 vs. 1.382) and Taobao (0.198 vs. 0.223), the held-out SurF checkpoint also matches or beats every AttNHP, THP and DTPP model in Table 3 that was trained on the target corpus; on Retweet, Taxi and StackOverflow it does not. This does not rely on choosing the variant per dataset: SurF-GLQ alone matches or beats them on all three (0.332, 1.241, 0.200).

On Amazon (0.330 vs. 0.402), Taobao (0.243 vs. 0.361) and Earthquake (1.229 vs. 1.380), SurF-CSB is *better* with the target removed from training than with it included. We attribute this to mixture composition: the six-corpus mixture is unbalanced (Retweet and Amazon supply 69% of events, Table 9), so removing one corpus changes the effective weights of the other five, and on Taobao the five-corpus mixture is the better-matched one. Loss-weighted or temperature-balanced mixing is a direct way to further improve the joint checkpoint.

**Leave-one-out likelihood.** RMSE scores only a point prediction. To test whether transfer also holds for the full predictive distribution, Table 10 reports the held-out time NLL of the leave-one-out SurF-GLQ checkpoint and of DTPP, the strongest baseline in our comparison, on the same folds and

the same events. SurF-GLQ attains lower NLL on all six held-out corpora, including StackOverflow, where its leave-one-out RMSE trails DTPP's by $0.032$.

Table 10: **Leave-one-out time NLL** (nats per event, lower is better) on the test split of the held-out corpus, which neither checkpoint saw during training. The score is $-\log p(\tau_i \mid \mathcal{H}_{t_i})$ of each observed inter-arrival time, without the end-of-window survival term, averaged over the same events for both models. Time units differ across corpora, so compare within a column only.

| **Model (leave-one-out)** | **Amazon** | **Retweet** | **Taxi** | **Earthquake** | **StackOverflow** | **Taobao** |
|---|---|---|---|---|---|---|
| DTPP | $0.515$ | $7.938$ | $0.687$ | $0.955$ | $1.257$ | $-0.875$ |
| SURF-GLQ | $\mathbf{0.243}$ | $\mathbf{7.622}$ | $\mathbf{-0.013}$ | $\mathbf{0.230}$ | $\mathbf{0.994}$ | $\mathbf{-1.519}$ |
| Events scored | $82{,}197$ | $59{,}634$ | $14{,}420$ | $13{,}852$ | $26{,}117$ | $27{,}955$ |