A ADDITIONAL METHODOLOGY DETAILS

A.1 DERIVATION OF EQUATION 4

We denote $\tau := t - t'$, $\nu := s - s'$, the variables $t' \in [0, T]$, $\tau \in [0, \tau_{\max}]$, $s' \in \mathcal{S}$ and $\nu \in B(0, a_{\max})$, where the sets $\mathcal{S}, B(0, a_{\max}) \subset \mathbb{R}^2$. Viewing the spatial and temporal variables, i.e., (t', τ) and (s', ν), as left and right mode variables, respectively, the kernel function SVD (Mollenhauer et al., 2020; Mercer, 1909) of k gives that

$$k(t', \tau, s', \nu) = \sum_{k=1}^{\infty} \sigma_k g_k(t', \tau) h_k(s', \nu). \tag{A.1}$$

We assume that the SVD can be truncated at $k \le K$ with a residual of ε for some small $\varepsilon > 0$, and this holds as long as the singular values σ_k decay sufficiently fast. To fulfill the approximate finite-rank representation, it suffices to have the scalars σ_k and the functions g_k and h_k so that the expansion approximates the kernel k, even if they are not SVD of the kernel. This leads to the following assumption:

Assumption A.1. *There exist coefficients σ_k, and functions $g_k(t', \tau)$, $h_k(s', \nu)$ s.t.*

$$k(t', \tau, s', \nu) = \sum_{k=1}^{K} \sigma_k g_k(t', \tau) h_k(s', \nu) + O(\varepsilon). \tag{A.2}$$

To proceed, one can apply kernel SVD again to g_k and h_k respectively, and obtain left and right singular functions that potentially differ for different k. Here, we impose that *across $k = 1, \cdots, K$, the singular functions of g_k are the same* (as shown below, being approximately same suffices) set of basis functions, that is,

$$g_k(t', \tau) = \sum_{l=1}^{\infty} \beta_{k,l} \psi_l(t') \varphi_l(\tau).$$

As we will truncate l to be up to a finite rank again (up to an $O(\varepsilon)$ residual) we require the (approximately) shared singular modes only up to L. Similarly as above, technically it suffices to have a finite-rank expansion to achieve the $O(\varepsilon)$ error without requiring them to be SVD, which leads to the following assumption where we assume the same condition for h_k:

Assumption A.2. *For the g_k and h_k in equation A.2, up to an $O(\varepsilon)$ error,*

(i) The K temporal kernel functions $g_k(t', \tau)$ can be approximated under a same set of left and right basis functions, i.e., there exist coefficients β_{kl}, and functions $\psi_l(t')$, $\varphi_l(\tau)$ for $l = 1, \cdots, L$, s.t.

$$g_k(t', \tau) = \sum_{l=1}^{L} \beta_{kl} \psi_l(t') \varphi_l(\tau) + O(\varepsilon), \quad k = 1, \cdots, K. \tag{A.3}$$

(ii) The K spatial kernel functions $h_k(s', \nu)$ can be approximated under a same set of left and right basis functions, i.e., there exist coefficients γ_{kr}, and functions $u_r(s')$, $v_r(\nu)$ for $r = 1, \cdots, R$, s.t.

$$h_k(s', \nu) = \sum_{r=1}^{R} \gamma_{kr} u_r(t') v_r(\nu) + O(\varepsilon), \quad r = 1, \cdots, R. \tag{A.4}$$

Inserting equation A.3 and equation A.4 into equation A.2 gives the rank-truncated representation of the kernel function. Since K, L, R are fixed numbers, assuming boundedness of all the coefficients and functions, we have the representation with the final residual as $O(\varepsilon)$, namely,

$$k(t', \tau, s', \nu) = \sum_{l=1}^{L} \sum_{r=1}^{R} \sum_{k=1}^{K} \sigma_k \beta_{kl} \gamma_{kr} \psi_l(t') \varphi_l(\tau) u_r(t') v_r(\nu) + O(\varepsilon).$$

Defining $\sum_{k=1}^{K} \sigma_k \beta_{kl} \gamma_{kr}$ as α_{lr} leads to equation 4.

A.2 ALGORITHMS

Algorithm 1 Model parameter estimation

```
Input: Training set $X$, batch size $M$, epoch number $E$, learning rate $\gamma$, constant $a > 1$ to update $s$ in equation 6.
Initialization: model parameter $\theta_0$, first epoch $e = 0$, $s = s_0$.
while $e < E$ do
  for each batch with size $M$ do
    1. For 1D temporal point process, compute $\ell(\theta)$, $\{\lambda(t_{c_t})\}_{c_t=1,\dots,|\mathcal{U}_{\text{bar},t}|}$. For spatio-temporal point process, compute $\ell(\theta)$, $\{\lambda(t_{c_t}, s_{c_s})\}_{c_t=1,\dots,|\mathcal{U}_{\text{bar},t}|, c_s=1,\dots,|\mathcal{U}_{\text{bar},s}|}$.
    2. Set $b = \min\{\lambda(t_{c_t})\}_{c_t=1,\dots,|\mathcal{U}_{\text{bar},t}|} - \epsilon$ (or $\min\{\{\lambda(t_{c_t}, s_{c_s})\}_{c_t=1,\dots,|\mathcal{U}_{\text{bar},t}|, c_s=1,\dots,|\mathcal{U}_{\text{bar},s}|} - \epsilon$), where $\epsilon$ is a small value to guarantee logarithm feasibility.
    3. Compute $\mathcal{L}(\theta) = -\ell(\theta) + \frac{1}{w} p(\theta, b)$.
    4. Update $\theta_{e+1} \leftarrow \theta_e - \gamma \frac{\partial \mathcal{L}}{\partial \theta_e}$.
    5. $e \leftarrow e + 1, w \leftarrow w \cdot a$
  end for
end while
```

Algorithm 2 Synthetic data generation

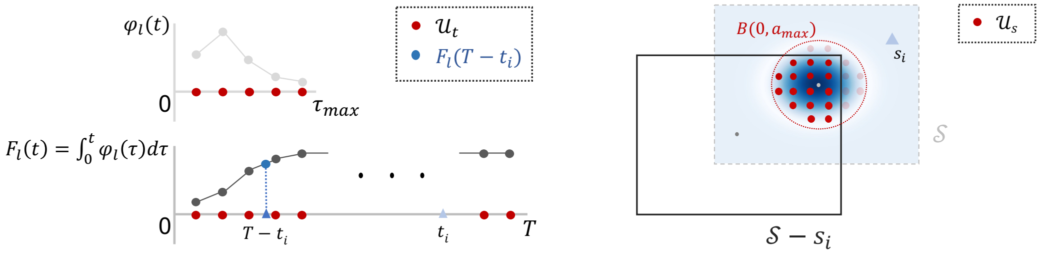
```
Input: Model $\lambda(\cdot), T, \mathcal{S}$, Upper bound of conditional intensity $\bar{\lambda}$.
Initialization: $\mathcal{H}_T = \emptyset, t = 0, n = 0$
while $t < T$ do
  1. Sample $u \sim \text{Unif}(0, 1)$.
  2. $t \leftarrow t - \ln u / \bar{\lambda}$.
  3. Sample $s \sim \text{Unif}(\mathcal{S}), D \sim \text{Unif}(0, 1)$.
  4. $\lambda = \lambda(t, s | \mathcal{H}_T)$.
  if $D\bar{\lambda} \le \lambda$ then
    $n \leftarrow n + 1; t_n = t, s_n = s$.
    $\mathcal{H}_T \leftarrow \mathcal{H}_T \cup \{(t_n, s_n)\}$.
  end if
end while
if $t_n >= T$ then
  return $\mathcal{H}_T - \{(t_n, s_n)\}$
else
  return $\mathcal{H}_T$
end if
```

A.3 GRID-BASED MODEL COMPUTATION

In this section, we elaborate on the details of the grid-based efficient model computation.

In Figure A.1, we visualize the procedure of computing the integrals of $\int_0^{T-t_i} \varphi_l(t)dt$ and $\int_{\mathcal{S}} v_r(s - s_i)ds$ in equation 8, respectively. Panel (a) illustrates the calculation of $\int_0^{T-t_i} \varphi_l(t)dt$. As explained in Section 4.2, the evaluations of φ_l only happens on the grid $\mathcal{U}_t$ over $[0, \tau_{\text{max}}]$ (since $\varphi_l(t) = 0$ when $t > \tau_{\text{max}}$). The value of $F(t) = \int_0^t \varphi_l(\tau)d\tau$ on the grid can be obtained through numerical integration. Then given t_i, the value of $F(T - t_i) = \int_0^{T-t_i} \varphi_l(t)dt$ is calculated using linear interpolation of F on two adjacent grid points of $T - t_i$. Panel (b) shows the computation of $\int_{\mathcal{S}} v_r(s - s_i)ds$. Given s_i, $\int_{\mathcal{S}} v_r(s - s_i)ds = \int_{B(0, a_{\text{max}}) \cap \{\mathcal{S} - s_i\}} v_r(s)ds$ since $v_r(s) = 0$ when $s > a_{\text{max}}$. Then $B(0, a_{\text{max}})$ is discretized into the grid $\mathcal{U}_s$, and $\int_{\mathcal{S}} v_r(s - s_i)ds$ can be calculated based on the value of v_r on the grid points in $\mathcal{U}_s \cap \mathcal{S} - s_i$ (the deep red dots in Figure A.1(b)) using numerical integration.

To evaluate the sensitivity of our model to the chosen grids, we compare the performance of `DNSK+Barrier` on 3D Data set 2 using grids with different resolutions. The quantitative results of testing log-likelihood and intensity prediction error are reported in Table A.1. We use $|\mathcal{U}_t| = 50, |\mathcal{U}_s| = 1500$ for the experiments in the main paper. As we can see, the model shows similar performances when a higher grid resolution is used and works slightly less accurately but still



(a) (b)

Figure A.1: (a) Computation of $\int_0^{T-t_i} \varphi_l(t)dt$ computation based on grid $\mathcal{U}_t$. Red dots represent grid points. Line segments connecting two light or dark grey dots represent the linear interpolation of φ_l and F_l. Here t_i is the time of the historical event which is fixed. (b) Computation of $\int_{\mathcal{S}} v_r(s - s_i)ds$ based on grid $\mathcal{U}_s$. The background heatmap represents the evaluation of v_r over $\mathcal{S}$. Here the fixed s_i is the location of the historical event. The integral is calculated based on the values of v_r on grid points with dark red color.

Table A.1: Comparison of `DNSK+Barrier` performance on 3D Data set 2 with different grid resolutions. Testing log-likelihood per event and intensity MRE are reported. The highlighted ones are the results in the main paper.

	Spatial resolution: $\lvert\mathcal{U}_s\rvert$		
Temporal resolution: $\lvert\mathcal{U}_t\rvert$	1000	1500	3000
30	$-2.272_{(0.005)}/0.102$	$-2.252_{(0.002)}/0.088$	$-2.250_{(0.002)}/0.081$
50	$-2.257_{(0.002)}/0.095$	$-2.251_{(0.001)}/0.082$	$-2.249_{(0.001)}/0.078$
100	$-2.255_{(0.001)}/0.091$	$-2.252_{(0.001)}/0.081$	$-2.250_{(0.001)}/0.078$

better than other baselines with less number of grid points. It reveals that our choice of grid resolution is accurate enough to capture the complex dynamics of event occurrences for this non-stationary data, and the model performance is robust to different grid resolutions.

In practice, the grids can be flexibly chosen to reach the balance of model accuracy and computational efficiency. For instance, the number of uniformly distributed grid points along one dimension can be chosen around $\mathcal{O}(n_0)$, where n_0 is the average number of events in one observed sequence. Note that $|\mathcal{U}_t|$ or $|\mathcal{U}_s|$ would be far less than the total number of observed events because we use thousands of sequences (2000 in our synthetic experiments) for model learning. And the grid size can be even smaller when it comes to non-Lebesgue-measured space.

A.4 DETAILS OF COMPUTATIONAL COMPLEXITY

We provide the detailed analysis of the $\mathcal{O}(n)$ computation complexity of $\mathcal{L}(\theta)$ in Section 4.3 as following:

- Computation of log-summation. The evaluation of $\{u_r\}_{r=1}^R$ and $\{\psi_l\}_{l=1}^L$ over n events costs $\mathcal{O}((R + L)n)$ complexity. The evaluation of $\{\varphi_l\}_{l=1}^L$ is of $\mathcal{O}(L|\mathcal{U}_t|)$ complexity since it relies on the grid $\mathcal{U}_t$. With the assumption that the conditional intensity is bounded by a constant C in a finite time horizon (Lewis and Shedler, 1979; Daley et al., 2003; Zhu et al., 2022), for each fixed j, the cardinality of set $\{(i, j) \mid t_j < t_i \leq t_j + \tau_{\text{max}}\}$ is less than $C\tau_{\text{max}}$, which leads to a $\mathcal{O}(RC\tau_{\text{max}}n)$ complexity of $\{v_r\}_{r=1}^R$ evaluation.
- Computation of integral. The integration of $\{\varphi_l\}_{l=1}^L$ only relies on numerical operations of $\{\varphi_l\}_{l=1}^L$ on grids $\mathcal{U}_t$ without extra evaluations of neural networks. The integration of $\{v_r\}_{r=1}^R$ depends on the evaluation on grid $\mathcal{U}_s$ of $\mathcal{O}(R|\mathcal{U}_s|)$ complexity.
- Computation of barrier. $\{\varphi_l\}_{l=1}^L$ on grid $\mathcal{U}_{\text{bar},t}$ is estimated by numerical interpolation of previously computed $\{\varphi_l\}_{l=1}^L$ on grid $\mathcal{U}_t$. Additional neural network evaluations of $\{v_r\}_{r=1}^R$ cost no more than $\mathcal{O}(RC\tau_{\text{max}}n)$ complexity.

B DEEP NON-STATIONARY KERNEL FOR MARKED STPPS

In marked STPPs (Reinhart, 2018), each observed event is associated with additional information describing event attribute, denoted as $m \in \mathcal{M} \subset \mathbb{R}^{d_{\mathcal{M}}}$. Let $\mathcal{H} = \{(t_i, s_i, m_i)\}_{i=1}^{n}$ denote the event sequence. Given the observed history $\mathcal{H}_t = \{(t_i, s_i, m_i) \in \mathcal{H} | t_i < t\}$, the conditional intensity function of a marked STPPs is similarly defined as:

$$\lambda\left(t, s, m\right) = \lim_{\Delta t \downarrow 0, \Delta s \downarrow 0, \Delta m \downarrow 0} \frac{\mathbb{E}\left[\mathbb{N}([t, t+\Delta t] \times B(s, \Delta s) \times B(m, \Delta m)) \mid \mathcal{H}_t\right]}{|B(s, \Delta s)||B(m, \Delta m)|\Delta t},$$

where $B(m, \Delta m)$ is a ball centered at $m \in \mathbb{R}^{d_{\mathcal{M}}}$ with radius Δm. The log-likelihood of observing $\mathcal{H}$ on $[0, T] \times \mathcal{S} \times \mathcal{M}$ is given by

$$\ell(\mathcal{H}) = \sum_{i=1}^{n} \log \lambda\left(t_i, s_i, m_i\right) - \int_0^T \int_{\mathcal{S}} \int_{\mathcal{M}} \lambda(t, s, m) dm ds dt.$$

B.1 KERNEL INCORPORATING MARKS

One of the salient features of our spatio-temporal kernel framework is that it can be conveniently adopted in modeling marked STPPs with additional sets of mark basis functions $\{g_q, h_q\}_{q=1}^{Q}$. We modify the influence kernel function k accordingly as following:

$$k(t', t-t', s', s-s', m', m) = \sum_{q=1}^{Q} \sum_{r=1}^{R} \sum_{l=1}^{L} \alpha_{lrq} \psi_l(t') \varphi_l(t-t') u_r(s') v_r(s-s') g_q(m') h_q(m).$$

Here $m', m \in \mathcal{M} \subset \mathbb{R}^{d_{\mathcal{M}}}$ and $\{g_q, h_q : \mathcal{M} \to \mathbb{R}, q = 1, \dots, Q\}$ represented by independent neural networks model the influence of historical mark m' and current mark m, respectively. Since the mark space $\mathcal{M}$ is always categorical and the difference between m' and m is of little practical meaning, we use g_q and h_q to model m' and m separately instead of modeling $m - m'$.

B.2 LOG-BARRIER AND MODEL COMPUTATION

The conditional intensity for marked spatio-temporal point processes at (t, s, m) can be written as:

$$\lambda(t, s, m) = \mu + \sum_{l,r,q} \alpha_{lrq} \sum_{(t_i, s_i, m_i) \in \mathcal{H}_t} \psi_l(t_i) \varphi(t-t_i) u_r(s_i) v_r(s-s_i) g_q(m_i) h_q(m).$$

We need to guarantee the non-negativity of λ over the space of $[0, T] \times \mathcal{S} \times \mathcal{M}$. When the total number of unique categorical mark in $\mathcal{M}$ is small, the log-barrier can be conveniently computed as the summation of λ on grids $\mathcal{U}_{\text{bar},t} \times \mathcal{U}_{\text{bar},s} \times \mathcal{M}$. In the following we focus on the case that $\mathcal{M}$ is high-dimensional with $\mathcal{O}(n)$ number of unique marks.

For model simplicity we use non-negative g_q and h_q in this case (which can be done by adding a non-negative activation function to the linear output layer in neural networks). We re-write $\lambda(t, s, m)$ and denote as following:

$$\lambda(t, s, m) = \mu + \underbrace{\sum_{q} \left(\sum_{l,r} \alpha_{lrq} \sum_{(t_i, s_i, m_i) \in \mathcal{H}_t} \psi_l(t_i) \varphi(t-t_i) u_r(s_i) v_r(s-s_i) g_q(m_i) \right)}_{\hat{F}_q(t,s)} h_q(m).$$

Note that the function in the brackets are only with regard to t, s. We denote it as $\hat{F}_q(t, s)$ (since it is in the rth rank of mark). Since $h_q(m) \geq 0$, the non-negativity of λ can be guaranteed by the non-negativity of $\hat{F}_q(t, s)$. Thus we apply log-barrier method on $\hat{F}_q(t, s)$. The log-barrier term becomes:

$$p(\theta, b) := -\frac{1}{Q|\mathcal{U}_{\text{bar},t} \times \mathcal{U}_{\text{bar},s}|} \sum_{c_t=1}^{|\mathcal{U}_{\text{bar},t}|} \sum_{c_s=1}^{|\mathcal{U}_{\text{bar},s}|} \sum_{q=1}^{Q} \log(\hat{F}_q(t_{c_t}, s_{c_s}) - b),$$

Since our model is low-rank, the value of Q will not be large.

For the model computation, the additional evaluations for $\{g_q\}_{q=1}^Q$ on events is of $\mathcal{O}(Qn)$ complexity and the evaluations for $\{h_q\}_{q=1}^Q$ only depends on the unique number of marks which at most of $\mathcal{O}(n)$. The log-barrier method does not introduce extra evaluation in mark space. Thus the overall computation complexity for `DNSK` in marked STPPs is still $\mathcal{O}(n)$.

C ADDITIONAL EXPERIMENTAL RESULTS

In this section we provide details of data sets and experimental setup, together with additional experimental results.

Synthetic data sets. To show the robustness of our model, we generate three temporal data sets and three spatio-temporal data sets using the following kernels:

(i) 1D Data set 1 with exponential kernel: $k(t', t) = 0.8e^{-(t-t')}$.

(ii) 1D Data set 2 with non-stationary kernel: $k(t', t) = 0.3(0.5 + 0.5\cos(0.2t'))e^{-2(t-t')}$.

(iii) 1D Data set 3 with infinite rank kernel:

$$k(t', t) = 0.3\sum_{j=1}^{\infty} 2^{-j}\left(0.3 + \cos(2 + (\frac{t'}{5})^{0.7}1.3(j+1)\pi)\right)e^{-\frac{8(t-t')^2}{25}j^2}$$

(iv) 2D Data set 1 with exponential kernel: $k(t', t, s', s) = 0.5e^{-1.5(t-t')}e^{-0.8s'}$.

(v) 3D Data set 1 with non-stationary inhibition kernel:

$$k(t', t, s', s) = 0.3(1 - 0.01t)e^{-2(t-t')}\frac{1}{2\pi\sigma_{s'}^2}e^{-\frac{\|s'\|^2}{2\sigma_{s'}^2}}\frac{\cos(10\|s - s'\|)}{2\pi\sigma_s^2(1 + e^{10(\|s-s'\|-0.5)})}e^{-\frac{\|s-s'\|^2}{2\sigma_s^2}}$$

, where $\sigma_{s'} = 0.5, \sigma_s = 0.15$.

(vi) 3D Data set 2 with non-stationary mixture kernel:

$$k(t', t, s', s) = \sum_{r=1}^{2}\sum_{l=1}^{2}\alpha_{rl}u_r(s')v_r(s - s')\psi_l(t')\varphi_l(t - t')$$

, where $u_1(s') = 1 - a_s(s'_2 + 1), u_2(s') = 1 - b_s(s'_2 + 1), v_1(s - s') = \frac{1}{2\pi\sigma_1^2}e^{-\frac{\|s-s'\|^2}{2\sigma_1^2}}, v_2(s - s') = \frac{1}{2\pi\sigma_2^2}e^{-\frac{\|s-s'-0.8\|^2}{2\sigma_2^2}}, \psi_1(t') = 1 - a_t t', \psi_2(t') = 1 - b_t t', \varphi_1(t - t') = e^{-\beta(t-t')}, \varphi_2(t - t') = (t - t' - 1) \cdot I(t - t' < 3)$, and $a_s = 0.3, b_s = 0.4, a_t = 0.02, b_t = 0.02, \sigma_1 = 0.2, \sigma_2 = 0.3, \beta = 2, (\alpha_{11}, \alpha_{12}, \alpha_{21}, \alpha_{22}) = (0.6, 0.15, 0.225, 0.525)$.

Note that kernel (iii) is the one we illustrated in Figure 1, which is of infinite rank according to the formulas. In Figure 1, the value matrix of $k(t', t)$ and $k(t', t - t')$ are the kernel evaluations on a same 300×300 uniform grid. As we can see, the rank of the value matrix of the same kernel k is reduced from 298 to 7 after changing to the displacement-based kernel parameterization.

Details of Experimental setup. For `RMTPP` and `NH` we test embedding size of $\{32, 64, 128\}$ and choose 64 for experiments. For `THP` we take the default experiment setting recommended by Zuo et al. (2020). For `NSMPP` we use the same model setting in Zhu et al. (2022) with rank 5. Each experiment is implemented by the following procedure: Given the data set, we split 90% of the sequences as training set and 10% as testing set.

We use independent fully-connected neural networks with two-hidden layers for each basis function. Each layer contains 64 hidden nodes. The temporal rank of `DNSK+Barrier` is set to be 1 for synthetic data (i), (ii), (iv), (v), 2 for (vi), and 3 for (iii). The spatial rank is 1 for synthetic data (iv), (v) and 2 for (vi). The temporal and spatial rank for real data are both set to be 2 through cross

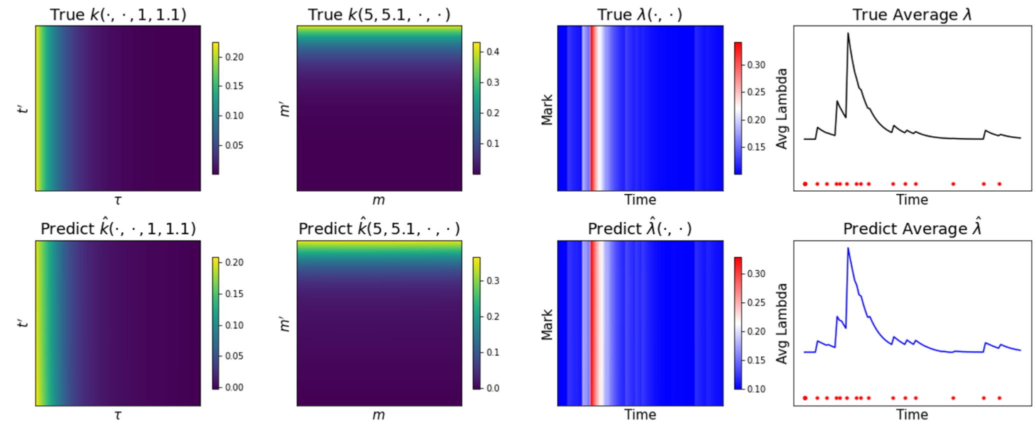


Figure A.2: Kernel recovery results of 2D exponential kernel. The first two columns show the true kernel and kernel learned by `DNSK+Barrier`. The last two columns shows the true and predicted conditional intensity functions of a test sequence. The line charts visualize the conditional intensity average over the 1D mark space at any given time for the ease of presentation. The red dots indicate the time of observed events.

validation. For each real data set, the $\tau_{\max}$ is chosen to be around $T/4$ and $s_{\max}$ is 1 for each data set since the location space is normalized before training. The hyper-parameter of `DNSK+Softplus` are the same as `DNSK+Barrier`. For `RMTPP`, `NH`, and `THP` the batch size is 32 and the learning rate is 10^{-3}. For others, the batch size is 64 and the learning rate is 10^{-1}. The quantitative results are collected by running each experiment for 5 independent times. All experiments are implemented on Google Colaboratory (Pro version) with 25GB RAM and a Tesla T4 GPU.

C.1 SYNTHETIC RESULTS WITH 2D & 3D KERNEL

In this section we present additional experiment results for the synthetic data sets with 2D exponential and 3D non-stationary mixture kernel. Our proposed model successfully recovers the kernel and event conditional intensity in both case. Note that the recovery of 3D mixture kernel demonstrates the capability of our model to handle complex event dependency with mixture patterns by conveniently setting time and mark rank to be more than 1.

C.2 ATLANTA TEXTUAL CRIME DATA WITH HIGH-DIMENSIONAL MARKS

Figure A.4 visualizes the fitting and prediction results of `DNSK+Barrier`. Our model presents an decaying pattern in temporal effect and captures two different patterns of spatial influence for incidents in the northeast. Besides, the in-sample and out-of-sample intensity predictions demonstrate the ability of `DNSK` to characterize the event occurrences by showing different conditional intensities.

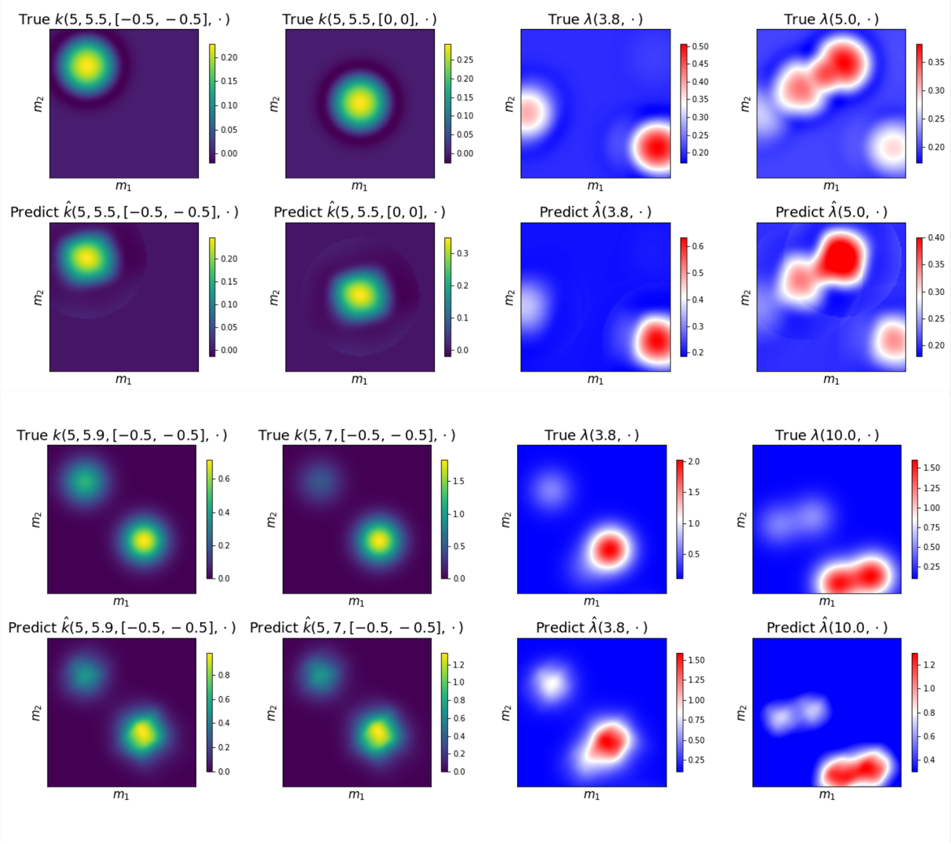


Figure A.3: Kernel recovery results of 3D non-stationary mixture kernel. The first two columns show snapshots of the true kernel and kernel learned by `DNSK+Barrier`. The last two columns shows snapshots of the true and predicted conditional intensity functions of a test sequence.

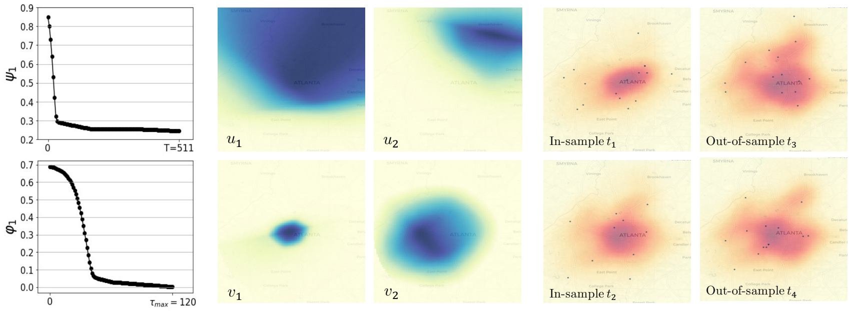


Figure A.4: Model fitting and prediction for high-dimensional real data. First column shows the learned temporal functions. Four panels in the middle shows the learned spatial functions, Deeper color depth indicates higher function value. Last four panels show the predicted conditional intensity over space at two in-sample times and two out-of-sample times, respectively. The dots represent event occurrences at that day.