A Score Matching Estimator

We provide the proof of the score matching estimator for temporal point processes and spatial point processes below, respectively. Generally speaking, the derivation for temporal point processes is more complex than spatial point processes because as we can see later there exists some complications arising from different limits of integration due to the order constraint on timestamps.

A.1 Temporal Point Processes

Given an observation window $[0, T]$, a sequence from a temporal point process is composed of a random number of timestamps arranged in a sequential order $S = \{t_n\}_{n=1}^N$ where $t_1 < t_2 < \ldots < t_N$ and $t_n \in [0, T]$ is the n-th event timestamp. We assume the ground-truth process generating the data has a density $p(S)$ and design a parameterized model with density $p_\theta(S)$ where θ is the model parameter to estimate. Following Sahani et al. [25], we define a Fisher-divergence objective:

$$F(\theta) = \mathbb{E}_{p(S)} \frac{1}{2} \sum_{n=1}^{N} \left(\frac{\partial \log p(S)}{\partial t_n} - \frac{\partial \log p_\theta(S)}{\partial t_n} \right)^2 . \tag{13}$$

The above loss can be understood as matching the variational derivatives of log-density w.r.t. the counting process $N(t)$ given that the counting process is non-decreasing and piece-wise constant with unit steps.

However, the above loss cannot be minimized directly as it depends on the gradient of the ground-truth data distribution which is unknown. Following the derivation of Hyvärinen [12], this dependence can be eliminated by using a trick of integration by parts. Let us expand Eq. (13), discard $(\frac{\partial \log p(S)}{\partial t_n})^2$ which does not depend on parameter θ, and examine the cross-term:

$$\begin{aligned}
&\mathbb{E}_{p(S)} \left[\frac{\partial \log p(S)}{\partial t_n} \frac{\partial \log p_\theta(S)}{\partial t_n} \right] \\
=& \int p(S) \frac{\partial \log p(S)}{\partial t_n} \frac{\partial \log p_\theta(S)}{\partial t_n} dS \\
=& \int_{S_{t_n^-}} \int_{t_n} \frac{\partial p(S)}{\partial t_n} \frac{\partial \log p_\theta(S)}{\partial t_n} dt_n dS_{t_n^-} \\
=& \int_{S_{t_n^-}} p(S) \frac{\partial \log p_\theta(S)}{\partial t_n} \Big|_{t_n = t_{n-1}}^{t_n = t_{n+1}} - \int_{t_n} p(S) \frac{\partial^2 \log p_\theta(S)}{\partial t_n^2} dt_n dS_{t_n^-} \\
=& \mathbb{E}_{p(S)} \left[\frac{\partial \log p_\theta(S)}{\partial t_n} (\delta(t_n - t_{n+1}) - \delta(t_n - t_{n-1})) - \frac{\partial^2 \log p_\theta(S)}{\partial t_n^2} \right] .
\end{aligned} \tag{14}$$

where $S_{t_n^-}$ represents the sequence excluding t_n, the fourth line uses integration by parts, the fifth line uses delta function to evaluate the limits of t_n given the order constraint of timestamps. Therefore, the loss in Eq. (13) can be rewritten as:

$$\begin{aligned}
F(\theta) = \mathbb{E}_{p(S)} &\left[\sum_{n=1}^{N} \frac{1}{2} \left(\frac{\partial \log p_\theta(S)}{\partial t_n} \right)^2 \right. \\
&\left. - \frac{\partial \log p_\theta(S)}{\partial t_n} (\delta(t_n - t_{n+1}) - \delta(t_n - t_{n-1})) + \frac{\partial^2 \log p_\theta(S)}{\partial t_n^2} \right] + C_1 .
\end{aligned} \tag{15}$$

where the constant C_1 does not depend on θ and can be discarded. To construct the final empirical loss, we replace the expectation by the empirical average, which eliminates the delta functions as any two timestamps cannot overlap with each other, and obtain the final version:

$$\hat{F}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{N_m} \frac{1}{2} \left(\frac{\partial \log p_\theta(S_m)}{\partial t_{m,n}} \right)^2 + \frac{\partial^2 \log p_\theta(S_m)}{\partial t_{m,n}^2} + C_1, \tag{16}$$

where we take M sequences $\{S_m\}_{m=1}^M$ from $p(S)$, $t_{m,n}$ is the n-th timestamp on the m-th sequence.

It is worth noting that Sahani et al. [25] assumed the parametric density satisfies the smoothness property: $\partial_{t_n} \log p_\theta(S)|_{t_n=t_{n+1}} = \partial_{t_{n+1}} \log p_\theta(S)|_{t_{n+1}=t_n}$ to cancel most delta functions. Here, we emphasize that this smoothness assumption is not necessary. In our derivation, we do not utilize this smoothness property, but just take advantage of the non-overlapping of timestamps to eliminate all delta functions and obtain the same objective.

A.2 Spatial Point Processes

Let us consider a planar point process for example. Given a 2-D observation region $\mathcal{X} \subseteq \mathcal{R}^2$, a realization from a 2-D spatial point process is composed of a random number of points $S = \{\mathbf{x}_n\}_{n=1}^N$ where $\mathbf{x}_n \in \mathcal{X}$ is the n-th event location (2-D coordinate). It is worth noting that these points are in no order. Similarly, we define a Fisher-divergence objective:

$$F(\theta) = \mathbb{E}_{p(S)} \frac{1}{2} \sum_{n=1}^{2N} \left(\frac{\partial \log p(S)}{\partial x_n} - \frac{\partial \log p_\theta(S)}{\partial x_n} \right)^2, \tag{17}$$

where x_n is any entry in vector $\mathbf{x}_n$, i.e., $x_n \in \mathbf{x}_n$.

Similarly, we use the trick of integration by parts to eliminate the dependence of the loss on the gradient of the unknown ground-truth data distribution. Let us expand Eq. (17), discard $(\frac{\partial \log p(S)}{\partial x_n})^2$ which does not depend on parameter θ, and examine the cross-term:

$$\begin{aligned}
&\mathbb{E}_{p(S)} \left[\frac{\partial \log p(S)}{\partial x_n} \frac{\partial \log p_\theta(S)}{\partial x_n} \right] \\
=& \int p(S) \frac{\partial \log p(S)}{\partial x_n} \frac{\partial \log p_\theta(S)}{\partial x_n} dS \\
=& \int_{S_{x_n^-}} \int_{x_n} \frac{\partial p(S)}{\partial x_n} \frac{\partial \log p_\theta(S)}{\partial x_n} dx_n dS_{x_n^-} \\
=& \int_{S_{x_n^-}} p(S) \frac{\partial \log p_\theta(S)}{\partial x_n} \bigg|_{x_n=-\infty}^{x_n=+\infty} - \int_{x_n} p(S) \frac{\partial^2 \log p_\theta(S)}{\partial x_n^2} dx_n dS_{x_n^-} \\
=& \mathbb{E}_{p(S)} \left[- \frac{\partial^2 \log p_\theta(S)}{\partial x_n^2} \right].
\end{aligned} \tag{18}$$

where $S_{x_n^-}$ represents the realization excluding x_n, the fourth line uses integration by parts, the fifth line assumes a weak regularity condition: $p(S)\partial_{x_n} \log p_\theta(S)$ goes to zero for any θ when $|x_n| \to \infty$. It is worth noting that in spatial point processes the limits of x_n are no longer constrained because there is no order for the points. Therefore, the loss in Eq. (17) can be rewritten as:

$$F(\theta) = \mathbb{E}_{p(S)} \left[\sum_{n=1}^{2N} \frac{1}{2} \left(\frac{\partial \log p_\theta(S)}{\partial x_n} \right)^2 + \frac{\partial^2 \log p_\theta(S)}{\partial x_n^2} \right] + C_1. \tag{19}$$

where the constant C_1 does not depend on θ and can be discarded. Replacing the expectation by the empirical average, we obtain the final empirical loss:

$$\hat{F}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{2N_m} \frac{1}{2} \left(\frac{\partial \log p_\theta(S_m)}{\partial x_{m,n}} \right)^2 + \frac{\partial^2 \log p_\theta(S_m)}{\partial x_{m,n}^2} + C_1, \tag{20}$$

where we take M realizations $\{S_m\}_{m=1}^M$ from $p(S)$, $x_{m,n}$ is the n-th variable on the m-th realization.

A.3 Spatio-temporal Point Processes

It is interesting to see that both score matching estimators for temporal point processes and spatial point processes have the same empirical loss (see Eq. (16) and Eq. (20)) regardless of whether the points are sequential or not. Therefore, it is easy to draw the conclusion that for spatio-temporal point processes the empirical score matching estimator is:

$$\hat{F}(\theta) = \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{\tilde{N}_m} \frac{1}{2} \left(\frac{\partial \log p_\theta(S_m)}{\partial s_{m,n}} \right)^2 + \frac{\partial^2 \log p_\theta(S_m)}{\partial s_{m,n}^2} + C_1, \tag{21}$$

where $s_{m,n}$ is the n-th variable on the m-th sequence, $\tilde{N}_m$ is the number of equivalent variables on the m-th sequence, $\tilde{N}_m = N_m$ for 1-D point process, e.g., temporal point process; $\tilde{N}_m = 2N_m$ for 2-D point process, e.g., spatial point process; and $\tilde{N}_m = 3N_m$ for 3-D point process, e.g., spatio-temporal point process, etc.

B Denoising Score Matching Estimator

In Appendix A, we provide an estimator trying to match the gradient of the log-density of the point process model to the log-density of the point process data. However, the estimator requires the second derivatives, which is computationally expensive. To avoid this issue, following the derivation of Vincent [31], we derive a denoising score matching estimator.

Differently, the denoising score matching estimator tries to match the gradient of the log-density of the model to the log-density of the noisy point process data. We add a small noise to the sequence S to obtain a noisy sequence $\tilde{S}$ (we add noise to each variable $\tilde{s}_n = s_n + \epsilon$), which is distributed as $p(\tilde{S}) = \int p(\tilde{S} \mid S)p(S)dS$. Therefore, the Fisher divergence between the noisy data distribution $p(\tilde{S})$ and model distribution $p_\theta(\tilde{S})$ is:

$$F_{\text{DSM}}(\theta) = \mathbb{E}_{p(\tilde{S})} \frac{1}{2} \sum_{n=1}^{\tilde{N}} \left(\frac{\partial \log p(\tilde{S})}{\partial \tilde{s}_n} - \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} \right)^2, \tag{22}$$

where $\tilde{s}_n$ is any entry in the noisy vector $\tilde{\mathbf{s}}_n$, i.e., $\tilde{s}_n \in \tilde{\mathbf{s}}_n = (\tilde{t}_n, \tilde{\mathbf{x}}_n)$, $\tilde{N}$ is the number of equivalent variables. Let us expand Eq. (22), discard $(\frac{\partial \log p(\tilde{S})}{\partial \tilde{s}_n})^2$ which does not depend on parameter θ, and examine the cross-term:

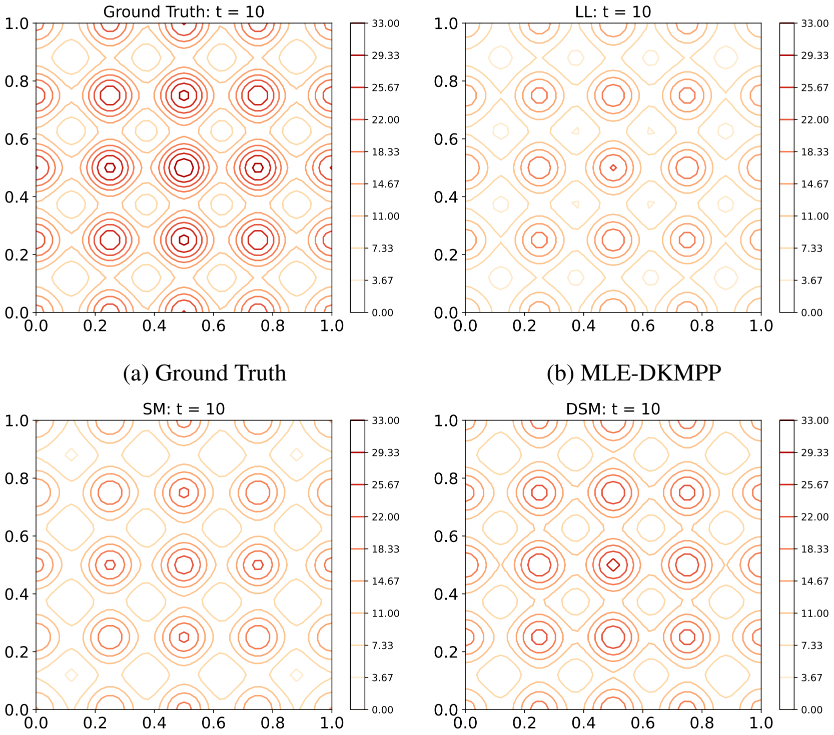
$$\begin{aligned}
&\mathbb{E}_{p(\tilde{S})} \left[\frac{\partial \log p(\tilde{S})}{\partial \tilde{s}_n} \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} \right] \\
=& \int p(\tilde{S}) \frac{\partial \log p(\tilde{S})}{\partial \tilde{s}_n} \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} d\tilde{S} \\
=& \int \frac{\partial}{\partial \tilde{s}_n} \int p(\tilde{S} \mid S)p(S)dS \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} d\tilde{S} \\
=& \iint p(S) \frac{\partial p(\tilde{S} \mid S)}{\partial \tilde{s}_n} \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} dS d\tilde{S} \\
=& \iint p(S)p(\tilde{S} \mid S) \frac{\partial \log p(\tilde{S} \mid S)}{\partial \tilde{s}_n} \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} dS d\tilde{S} \\
=& \mathbb{E}_{p(S,\tilde{S})} \left[\frac{\partial \log p(\tilde{S} \mid S)}{\partial \tilde{s}_n} \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} \right].
\end{aligned} \tag{23}$$

Therefore, the loss in Eq. (22) can be rewritten as:

$$\begin{aligned}
&F_{\text{DSM}}(\theta) \\
=& \sum_{n=1}^{\tilde{N}} \mathbb{E}_{p(\tilde{S})} \frac{1}{2} \left(\frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} \right)^2 - \mathbb{E}_{p(S,\tilde{S})} \left[\frac{\partial \log p(\tilde{S} \mid S)}{\partial \tilde{s}_n} \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} \right] + C \\
=& \mathbb{E}_{p(S,\tilde{S})} \sum_{n=1}^{\tilde{N}} \frac{1}{2} \left(\frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} \right)^2 - \frac{\partial \log p(\tilde{S} \mid S)}{\partial \tilde{s}_n} \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} + C \\
=& \mathbb{E}_{p(S,\tilde{S})} \frac{1}{2} \sum_{n=1}^{\tilde{N}} \left(\frac{\partial \log p(\tilde{S} \mid S)}{\partial \tilde{s}_n} - \frac{\partial \log p_\theta(\tilde{S})}{\partial \tilde{s}_n} \right)^2 + C_2,
\end{aligned} \tag{24}$$

where the constants C and C_2 do not depend on θ and can be discarded. Replacing the expectation by the empirical average, we obtain the final empirical loss:

$$\hat{F}_{\text{DSM}}(\theta) = \frac{1}{2M} \sum_{m=1}^{M} \sum_{n=1}^{\tilde{N}_m} \left(\frac{\partial \log p(\tilde{S}_m \mid S_m)}{\partial \tilde{s}_{m,n}} - \frac{\partial \log p_\theta(\tilde{S}_m)}{\partial \tilde{s}_{m,n}} \right)^2 + C_2, \tag{25}$$



(a) Ground Truth

(b) MLE-DKMPP

(c) Score-DKMPP

(d) Score-DKMPP+

Figure 2: The intensity function at $t = 10$ estimated by three estimators (MLE-DKMPP, Score-DKMPP, Score-DKMPP+) with 1,000 Monte Carlo (MC) samples. Three estimators exhibit similar performance. Score-DKMPP and Score-DKMPP+ do not require Monte Carlo integration, and thus their estimation remain consistent regardless of MC samples. In contrast, MLE-DKMPP heavily relies on MC sampling and therefore its performance depends on the number of MC samples.

where we take M clean and noisy sequences $\{S_m, \tilde{S}_m\}_{m=1}^{M}$ from $p(S, \tilde{S})$, $\tilde{s}_{m,n}$ is the n-th variable on the m-th noisy sequence, $\tilde{N}_m$ is the number of equivalent variables on the m-th noisy sequence.

C Experimental Details

C.1 Synthetic Data

Data Simulation We generate a 3-D spatio-temporal point process synthetic dataset. The spatial observation $\mathbf{x}$ spans the area of $[0, 1] \times [0, 1]$, while the temporal observation window covers the time interval of $[0, 10]$. We assume a 1-D covariate function $Z(\mathbf{u} = (\tau, \mathbf{r})) = (\mathcal{N}(r_1 \mid 0.5, 0.5) + \mathcal{N}(r_2 \mid 0.5, 0.5))$ on the domain. We set $f_{w_1}(\mathbf{u}_j, Z(\mathbf{u}_j)) = 20Z(\mathbf{u}_j) + 0.1$, $k_{\phi,w_2}(\mathbf{s}, \mathbf{u}_j) = k_\phi(g_{w_2}(\mathbf{s}), g_{w_2}(\mathbf{u}_j))$ where k_ϕ is the RBF kernel $k_\phi(\mathbf{x}, \mathbf{x}') = \exp\left(-\phi\|\mathbf{x} - \mathbf{x}'\|^2\right)$ with $\phi = 100$ and g_{w_2} is a linear transformation $g_{w_2}(\mathbf{s}) = \mathbf{s} + 0.1$. We fix the representative points on a regular grid: 5 representative points evenly spaced on each axis, so there are $5^3 = 125$ representative points in total. We use the thinning algorithm to generate 5,000 sequences according to the intensity function specified above. The statistics of the synthetic data are shown in Table 3.

Training Details We fit a DKMPP model to the synthetic data with the ground-truth representative points and an RBF base kernel. Both the kernel mixture weight network f and the non-linear transformation g in the deep kernel are implemented using MLPs with ReLU activation functions. Therefore, the learnable parameters are w_1, w_2, ϕ. The intensity functions at $t = 10$ estimated by three different estimators are shown in Fig. 2.

Table 3: The statistics of synthetic and real-world datasets.

Dataset	Covariate Dimension	# of sequences	average # of events per sequence
Synthetic	1	5,000	17
Crimes in Vancouver	1	1,096	87
NYC Vehicle Collisions	768	61	327
NYC Complaint Data	768	301	63

C.2 Real-world Data

Data Preprocessing The preprocessing details of three real-world datasets are shown below. We listed the statistics of three real-world datasets after preprocessing in Table 3.

Crimes in Vancouver This dataset is composed of more than 530 thousand crime records, including all categories of crimes committed in Vancouver from 2003 to 2017. Each crime record contains the time and location (latitude and longitude) of the crime. We split the data into multiple sequences by year, month and day. Then, we select events from 2013 to 2016, drop the NaN value, and scale the time and space into a volume of $[0, 1] \times [0, 1] \times [0, 10]$. We select the categorical feature 'Crime Type' as the descriptive feature for each event. We convert the categorical feature into the corresponding numerical feature as the covariate. Finally, the dimension of covariate $\mathbf{Z}$ is 1.

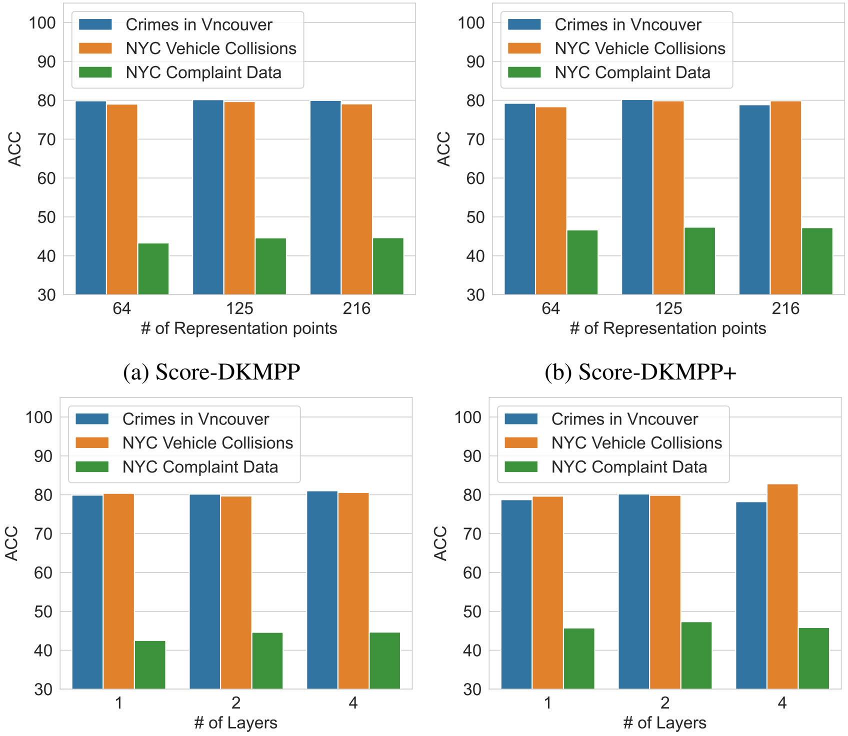
NYC Vehicle Collisions The New York City vehicle collision dataset contains about 1.05 million vehicle collision records. Each collision record includes the time and location (latitude and longitude). We split the data into multiple sequences by day. We select the records from 01/01/2019 to 02/03/2019, drop the NaN value and scale the time and space into a volume of $[0, 1] \times [0, 1] \times [0, 10]$. We select 'BOROUGH', 'CONTRIBUTING FACTOR VEHICLE 1', 'CONTRIBUTING FACTOR VEHICLE 2' and 'VEHICLE TYPE CODE 1' as the descriptive features for each event. We concatenate several textual features and use a pre-trained DistilBERT [26] to extract the textual features, and concatenate the textual features with other numerical/categorical features as the covariate. Finally, the dimension of covariate $\mathbf{Z}$ is 768.

NYC Complaint Data This dataset contains over 228 thousand complaint records in New York City. Each record includes the date, time, and location (latitude and longitude) of the complaint. We split the data into multiple sequences by hour. We select the records from 01/11/2022 to 13/11/2022, drop the NaN value and scale the time and space into a volume of $[0, 1] \times [0, 1] \times [0, 10]$. We select 'OFNS_DESC', 'JURIS_DESC', 'LAW_CAT_CD', 'PD_DESC', 'VIC_RACE', 'VIC_SEX' and 'PREM_TYP_DESC' as the descriptive features for each event. We concatenate several textual features and use a pre-trained DistilBERT [26] to extract the textual features, and concatenate the textual features with other numerical/categorical features as the covariate. Finally, the dimension of covariate $\mathbf{Z}$ is 768.

Training Details Each dataset is divided into training, validation and test data using a $50\%/40\%/10\%$ split ratio based on time. For the real-world data, we fix the representative points on a regular grid: 5 representative points evenly spaced on each axis, so there are $5^3 = 125$ representative points in total. We use three different kernel functions for comparisons: the RBF kernel $k_\phi(\mathbf{x}, \mathbf{x}') = \exp\left(-\phi\|\mathbf{x} - \mathbf{x}'\|^2\right)$, the rational quadratic (RQ) kernel $k_\phi(\mathbf{x}, \mathbf{x}') = (1+\phi\|\mathbf{x}-\mathbf{x}'\|^2)^{-\frac{1}{2}}$, and the Ornstein-Uhlenbeck (OU) kernel $k_\phi(\mathbf{x}, \mathbf{x}') = \exp\left(-\phi\|\mathbf{x} - \mathbf{x}'\|\right)$. For the three real-world datasets, the kernel mixture weight network f and the non-linear transformation g in the deep kernel are implemented using MLPs with ReLU activation functions and we fixed the number of layers in f and g as 2.

Hyperparameters We tested the performance of Score-DKMPP and Score-DKMPP+ with different hyperparameters on three real-world datasets. We tested the number of representation points and network structures. For the representation points, we tested three different values, 64, 125, and 216, respectively. For the network structure, we tested the number of layers of 1, 2, and 4. When we tested the effect of representation points, we fix the network with 2 hidden layers and when we tested the effect of network structure, we fix the number of representation points as 125.

For the representation points, the accuracy of Score-DKMPP and Score-DKMPP+ overall have better performance when the number of representation points is 125. The accuracy tends to increase when



(a) Score-DKMPP

(b) Score-DKMPP+

(c) Score-DKMPP

(d) Score-DKMPP+

Figure 3: (a) The ACC performance of Score-DKMPP with the number of representation points of 64, 125 and 216; (b) the ACC performance of Score-DKMPP+ with the number of representation points of 64, 125 and 216; (c) the ACC performance of Score-DKMPP with the number of layers of 1, 2 and 4; (d) the ACC performance of Score-DKMPP+ with the number of layers of 1, 2 and 4.

the number of representation points increases from 64 to 125, however, except for the performance of Score-DKMPP on the complaint dataset, for other real-world datasets, the accuracy starts to drop.

For the network structure, the accuracy using Score-DKMPP on the Crimes in Vancouver and NYC Complaint data tends to increase as the number of layers increases. However, for Score-DKMPP+, we only capture a similar trend on the NYC Vehicle Collisions data. Other than those mentioned above, we do not observe a significant pattern when increasing the number of layers.