メタ学習に基づく短期間イベントデータからの点過程モデリング

瀧本　祥章† 　田中　佑典†† 　岩田　具治†† 　大川　真耶†††
金 　秀明† 　戸田　浩之†††† 　倉島 　健†

† NTT 人間情報研究所　〒 239–0847 神奈川県横須賀市光の丘 1-1
†† NTT コミュニケーション科学基礎研究所　〒 619–0237 京都府相楽郡精華町光台 2-4 NTT 京阪奈ビル
††† NTT Research, Inc. PHI 研究所 HARVARD UNIVERSITY 52 OXFORD ST, CAMBRIDGE, MA 02138
†††† 横浜市立大学データサイエンス学部　〒 236–0027 神奈川県横浜市金沢区瀬戸 22-2
E-mail: †{yoshiaki.takimoto.ar,hideaki.kin.cn,takeshi.kurashima.uf}@hco.ntt.co.jp,
††{yusuke.tanaka.rh,tomoharu.iwata.gy}@hco.ntt.co.jp, †††maya.okawa@ntt-research.com,
††††hirotoda@acm.org

あらまし　タクシーの乗車，バイクシェアの貸出，犯罪の発生，病気の感染症など，人の行動は事象の発生時刻の系列からなるデータ（イベントデータ）で表せる．このようなイベントデータをモデル化し，未来の事象（イベント）を予測することは，交通制御や取り締まり，疫病対策など多くのアプリケーションで有用である．このようなイベントデータのモデル化と予測において点過程は有用な技術である．近年では，点過程モデルに深層学習を組み込みことにより，予測精度の向上を目指す取り組みが盛んになっている．しかし，それらの研究の多くは，学習のために長期間にわたって収集されたイベントデータを必要とし，現実のシナリオでは適用が困難なことが多い．この問題を解決するため，限られたデータ，すなわち，短期間のイベントデータから将来のイベントを予測するための新しいメタ学習手法を提案する．提案手法はまず，Recurrent Neural Network を介して，短期間のイベントデータから潜在表現（タスク表現）を得る．次にタスク表現を入力とする Monotonic Neural Network（MNN）によって，点過程の累積強度関数をモデル化する．更に，土地の利用法などの地理的特徴や周期パターンなど人の活動に影響を与える外的要因を考慮するため，MNN の拡張を行った．提案手法は関連するタスクから学習した事前知識を有効に活用することにより，短期間イベントデータでも高精度な予測ができる．複数の実データを用いた実験により，提案手法はいくつかの既存手法よりも高い予測性能を持つことを確認した．

キーワード　点過程，イベントデータ，メタ学習，深層学習

1 はじめに

タクシーの乗車 [1]，バイクシェアにおける貸出 [2]，犯罪の発生 [3]，病気の感染 [4] など，人の行動はイベントデータで表せる．このようなイベントデータは，いつ，どのようなイベントが発生したかを表す．イベントデータをモデル化し，将来のイベントの発生を予測することは，交通管理や取り締まり，疫病対策など多くのアプリケーションに有用である．例えば，バイクシェアの貸出がいつ，どこで発生するかを正確に予測できれば，運営会社はバイクを適切に配置でき，供給不足による機会損失を防げる．[5].

イベントデータ，すなわち，連続時間上に離散的に存在するイベント系列のモデル化には点過程がよく用いられる [6, 7]．点過程はイベントの発生確率を強度関数によって決定する．近年，この強度関数を深層学習によってモデル化することにより，表現力や精度を向上させる試みがある．例えば，Du らは点過程の一種である Hawkes 過程の強度関数をモデル化するのに Recurrent Neural Network （RNN）を使用した [8]．また，Omi らは [8] を Monotonic Neural Network（MNN）と統合することにより，更に表現力を向上させた [9]．これらの手法はイベント間の複雑な依存関係を学習することにより，高精度なイベント予測を実現している．

これらの深層学習ベースの点過程手法はイベント予測に優れた性能を示すが，学習のために長期間のイベントデータが必要である．しかし，実世界のアプリケーションでは，そのような長期間のイベントデータを利用できない場合がある．例えば，新たな地域へのビジネスの展開時や，バイクシェアにおける新しいバイクステーションを新設する場合である．

このようなデータ不足に対応するための手法としてメタ学習がある．時系列データのためのメタ学習手法としては，[10–16] が提案されている．これらの手法は長期の時系列データを持つ関連するタスクから事前知識の学習を行い，短期間しかデータのないタスクの予測に活用する．また，イベントデータを対象としたメタ学習手法としては，Xie らの Hawkes 過程を対象とした手法がある [17]．Xie らの手法は上記の手法と同様に短期間のイベントデータの集合から学習を行う．しかし，そのモデルの強度関数は予め定められたパラメトリックな仮定に依存しているため，表現力が制限される．また，この手法は人の行動に影響を与える外的要因を考慮することができない．外的な要

因の例としては，時間帯や曜日などの時間的な周期パターンや，土地の利用方法，犯罪発生数，通りの清潔さ，平均通勤時間などの地理的特徴が挙げられる．これらの特徴を考慮することにより，異なる場所間で共通する周期的なパターンなどの意味的な類似性を考慮でき，精度向上が期待できる．

そこで本研究では，既存の長期間のイベントデータを用いて学習を行い，未知の短期間のイベントデータから，将来のイベントを予測するための新しいメタ学習手法を提案する．提案手法では，まず RNN を用いて短期間のイベントデータ（サポートセット）からタスクの表現を得る．次に，タスク表現に基づいて強度関数を決定し，その強度関数を用いてイベント予測を行う．RNN を用いた定式化により，短期間のイベントの系列から得られる時間情報を十分に活用できる．また，強度関数は，予測時刻とタスク表現を入力とする MNN を用いて設計する．この設計により，学習は容易でありつつ，柔軟な強度関数の表現を学習できる．更に，強度関数に用いる MNN を地理的特徴および時間的な周期を考慮できるように拡張する．この拡張により，土地の利用方法などの地域の意味的な類似性を考慮しつつ，イベントデータにおける日ごと，週ごとの周期的パターンを自動的に学習可能になる．

短期間のイベントデータからのイベント予測の精度検証のため，2 つのデータを用いてで実験を行った．実験の結果，提案手法は既存の 3 手法よりも高い予測性能を持つことが確認された．また，実験結果から，提案手法は周期的な変化を捉えられ，地理的特徴を効果的に取り組むことが示された．

本稿の主な貢献は以下のとおりである．

- 短期間のイベントデータから予測可能な点過程モデルのためのメタ点過程学習フレームワークを提案
- 周期性を考慮するように MNN を拡張
- 実データを用いて，提案手法が既存手法よりもイベント予測が正確に行えることを確認

2 関 連 研 究

2.1 点 過 程

点過程は，タクシーの乗車 [1]，バイクシェアにおける貸出 [2]，ソフトウェアの故障 [18]，感染症の感染 [4]，金融取引 [19] など様々な分野でイベントデータのモデル化に利用されている [6, 7]．点過程ではイベントが発生する確率を強度関数で表現し，強度関数は最尤推定によって学習される．強度関数には，定常関数（定常ポアソン過程），時間依存で変化する関数（非定常ポアソン過程）[20]，イベントの履歴に応じて変化する関数（Hawkes 過程 [21] など）がよく用いられる．周期性を非定常ポアソン過程に組み込むため，強度関数を設計するといった取り組みが行われている [22–24]．しかし，人手で設計した強度関数はそのパラメトリックな仮定により表現力が制限されるといった課題がある．そこで強度関数の表現力を向上させるため，深層学習ベースの手法が提案されている [8, 9, 25–27]．例えば，[27] は非定常ポアソン過程と深層学習を組み合わせ，バスの停留所の利用状況を予測する．また，[8, 9, 25, 26] では Hawkes 過程を RNN や Attention 機構と組み合わせている．しかし，これらの手法はイベントデータから次のイベント時刻を予測することに着目しており，長期間のイベントデータが学習に利用できることを前提としている．そのため，短期間のイベントデータしかない場合は考慮されておらず，実世界のアプリケーションでは十分に学習ができず，高精度な予測を行えない可能性がある．

2.2 メ タ 学 習

メタ学習は，顔認識 [28]，物体検出 [29]，ロボットの模倣学習 [30]，密度推定 [31]，生成モデル [32, 33] などの様々な few-shot learning の問題設定において成果がでている．時系列データに対するメタ学習手法は [10–16] などがあるが，連続時間上のイベントデータに対して用いることはできない．また，短期間のイベントデータからイベント予測するためのメタ学習手法は非常に少ない．Xie らは Model-Agnostic Meta-Learning (MAML)[34] と Hawkes 過程 [21] を組み合わせたメタ学習手法を提案したが，この強度関数はパラメトリックな仮定に依存し，表現力に制約がある．MAML は二階微分を用いるため，Xie の手法は強度関数をニューラルネットなどの大規模モデルに置き換えることが難しい．また，人の活動に影響を与える周期性や地理的特徴などの外的な要因を取り組むことができない．本稿では， Black-box adaptation[35] に基づくメタ学習手法を提案する．本手法では，短期間の系列を RNN を介して埋め込み表現に変換し，メタ学習器の入力とする．既存手法と異なり，提案手法は二階微分を必要とせず，周期性や地理的特徴などの外的要因も対応可能である．

3 準 備

提案手法はイベントデータ，すなわち，連続時間上で離散的に発生するイベントの系列をモデル化や予測するのに広く用いられている点過程に基づいている [20]．本章では，点過程のフレームワークを簡単に説明する．

$t \in \mathbb{R}_{\geqq 0}$ をイベントの発生時刻とする．点過程ではイベントの発生が強度関数 $\lambda(t) \geqq 0$ によって決定すると仮定する．強度関数は

$$\lambda(t) = \lim_{\Delta \to 0} \frac{P(\text{one event occurs in } [t, t+\Delta])}{\Delta}, \qquad (1)$$

のように定義される．ここで Δ は微小時間である．強度関数 $\lambda(t)$ は時刻 t におけるイベント発生の瞬間的な確率であり，これを設計することによりイベント発生確率の時間変化をモデル化できる．

イベントデータ $X_T = \{t_1, \ldots, t_N\} (0 \leqq t_1 \leqq \cdots \leqq t_N \leqq T)$ が得られたときを考える．なお，T は観測期間を表し，N は期間内に観測されたイベント数を表す．X_T の尤度は

$$p(X_T) = \prod_{t_i \in X_T} \lambda(t_i) \times \exp\left[- \int_0^T \lambda(t) dt \right], \qquad (2)$$

のように書き下される．尤度 $p(X_T)$ の対数を最大化することによって，強度関数を推定可能である．また，推定された強度

関数を用いてイベントの発生を予測可能である．

4 提案手法

4.1 問題設定

まず初めに，**タスク**を定義する．タスクは観測されたイベントデータから強度関数を推定するという 1 つの問題を表す．なお，3 章で述べたフレームワークはイベントデータが 1 つであるため，**単一タスク**に相当する．一方で，本稿では**複数タスク**を取り扱う．例えば，バイクシェアシステムにおいて，複数のバイクステーションで発生する自転車の貸出をモデル化することである．本稿の目的は，複数のタスクから得られる大量のデータを利用して，**未知タスク**の短いイベントデータから将来のイベントを予測をすることである．この問題は様々なアプリケーションで重要であり，例えば**新設**のバイクステーションの利用状況をその初期の短いイベントデータから予測し，自転車の再配置を行うのに有効である．

上記の問題設定を数学的に表記すると以下のようになる．M をタスクの集合，$m \in M$ をタスク，$g \in \mathbb{R}^{K_g}$ を土地の利用方法などの地理的特徴を表す K_g 次元のベクトルであるとする．学習フェーズにて，複数のタスクのイベントデータと地理的特徴のペア $\mathcal{D} = \{(X^{(m)}_{T^{\mathrm{e}}}, g^{(m)})\}_{m \in M}$ を用いてモデルの学習を行う．その後予測フェーズにて，未知タスク $m^* \notin M$ の現在の時刻 T^{c} までの短期間のイベントデータ $X^{(m^*)}_{T^{\mathrm{c}}}$ と地理的特徴 $g^{(m^*)}$ が与えられたときに，将来の時間区間 $[T^{\mathrm{c}}, T^{\mathrm{e}}]$ における強度関数 $\lambda^{(m^*)}(t)$ を予測する．ここで，$T^{\mathrm{c}} < T^{\mathrm{e}}$ である．なお，本稿では単純化のため，$T^{\mathrm{c}}, T^{\mathrm{e}}$ を予め定められた値としているが，タスクごとに異なる値をとるなどの拡張は容易である．

4.2 モデル概要

未知のタスクについて，短期間のイベントデータから将来発生するイベントを予測する問題のために，我々は点過程のための新しいメタ学習手法を提案する．提案するフレームワークの全体像は図 1 のとおりである．学習フェーズでは，MAML[34] を参考に，予測フェーズをシミュレーションするエピソード学習フレームワークに従う．そのため，まず各タスクの学習データ $X^{(m)}_{T^{\mathrm{e}}}$ を 2 つの集合，**サポートセット** $\mathcal{S}^{(m)} = \{t^{(m)}_i \mid 0 \leqq t^{(m)}_i \leqq T^{\mathrm{c}}\}$ および**クエリセット** $\mathcal{Q}^{(m)} = \{t^{(m)}_i \mid T^{\mathrm{c}} < t^{(m)}_i \leqq T^{\mathrm{e}}\}$ に分割する．図 1 に示すように，サポートセット $\{\mathcal{S}^{(m)}\}$ と地理的特徴 $\{g^{(m)}\}$ から強度関数が推定され，サポートセットおよびクエリセットから損失関数が計算される．この手順により，未知タスクのイベントデータが短い場合であっても，そのタスクの将来のイベントを予測できるようにモデルを学習可能となる．本フレームワークは**タスク表現エンコーダ**と**強度予測器**の 2 つの重要な構成要素を持つ．タスク表現エンコーダは各タスクのイベントデータと地理的特徴をタスク表現と呼ぶ潜在空間に埋め込み，タスク間の関係性を推定できるようにする．強度予測器は，周期的パターンを明示的にモデル化した深層学習ベースの強度関数である．この定式化により，人の行動における周期性を考慮しつつ，柔軟な強度関数をモデル化することが可能になる．これらの詳細はそれぞれ 4.3 節と 4.4 節 に，メタ学習のアルゴリズムは 4.5 節に後述する．

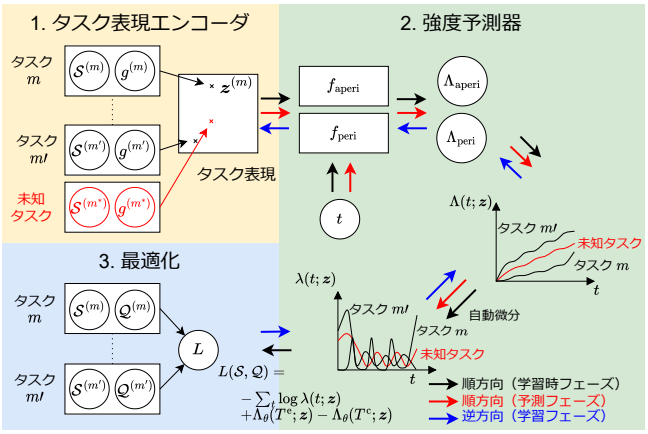


図 1: 提案モデル． **学習フェーズ:** まず，タスク表現 $\boldsymbol{z}$ が各サポートセット $\mathcal{S}$ 及び地理的特徴 g から推定される．次に各タスク専用の強度関数 $\lambda(t;\boldsymbol{z})$ の各時刻 t の値をタスク表現から推定する．$\lambda(t;\boldsymbol{z})$ は周期的強度関数 $\lambda_{\mathrm{peri}}(t;\boldsymbol{z})$ と非周期的強度関数 $\lambda_{\mathrm{aperi}}(t;\boldsymbol{z})$ の和であり，それぞれの積分が MNN（$f_{\mathrm{peri}}(t,\boldsymbol{z})$, $f_{\mathrm{aperi}}(t,\boldsymbol{z})$）によってモデル化される．その後，損失 L を各タスク専用の強度関数から計算し，誤差逆伝播法で学習を行う．**予測フェーズ:** まず，タスク表現 z をサポートセット $\mathcal{S}^{(m^*)}$ から得る．その後，学習フェーズと同様に強度関数を推定する．

4.3 タスク表現エンコーダ

我々はタスク間の関係性をモデル化するために，サポートセット $\mathcal{S}^{(m)}$ と地理的特徴 $g^{(m)}$ をニューラルネットワークで埋め込んで得られる $K_{\boldsymbol{z}}$ 次元のベクトルであるタスク表現 $\boldsymbol{z} \in \mathbb{R}^{K_{\boldsymbol{z}}}$ を導入する．まず，サポートセットを潜在空間に埋め込んで $K_{\mathcal{S}}$ 次元のベクトルであるサポート表現 $\boldsymbol{z}_{\mathcal{S}} \in \mathbb{R}^{K_{\mathcal{S}}}$ を得る．その後，サポート表現に地理的特徴を取り込み，タスク表現 $\boldsymbol{z}$ を得る．

サポート表現．例えば，人気のあるスポットには多くの人が集まるなど，イベントは通常互いに関連している．このようなイベント間の相互の関係性を捉えるために，双方向 RNN をエンコーダとして使用する．タスク m に対するサポート表現は

$$\boldsymbol{z}^{(m)}_{\mathcal{S}} = \frac{1}{|\mathcal{S}^{(m)}|} \sum_{n=1}^{|\mathcal{S}^{(m)}|} \mathrm{RNN}\left(t^{(m)}_n, t^{(m)}_n - t^{(m)}_{n-1}\right), \qquad (3)$$

とした．RNN の入力は，$t^{(m)}_n$ は n 番目のイベントと，その前のイベントとの時間差 $t^{(m)}_n - t^{(m)}_{n-1}$ である．ここで，$|\cdot|$ は集合に含まれる要素数を表し，$t_0 = 0$ とする．なお，RNN に入力されるイベントはサポートセットに含まれるイベントのみであり，予測対象であるクエリセットは入力されないため，予測時でも双方向 RNN を用いることが可能である．

タスク表現．例えば，住宅地では通勤による朝夕の移動が多く，商業地では買い物客による日中の移動が多いなど，地理的特徴もイベントの発生に関連がある．このような関係を把握するため，地理的特徴をサポート表現に取り込む．タスク m に対するタスク表現は

$$\boldsymbol{z}^{(m)} = \mathrm{FNN}\Big(\boldsymbol{z}_{\mathcal{S}}^{(m)}, g^{(m)}\Big), \tag{4}$$

とした．ここで，$\mathrm{FNN} : \mathbb{R}^{K_{\mathcal{S}}+K_{\mathrm{g}}} \to \mathbb{R}^{K_{\mathrm{z}}}$ は全結合層である．

なお，(3) 中の RNN および (4) 中の FNN は全てのタスクで共通である．RNN と地理的特徴により，それぞれのイベントデータの初期パターンを適切に捉え，タスク表現に反映できる．そのため，提案手法は短期間の系列の初期パターンの違いを識別することにより，将来の強度関数を精度良く予測できる．

4.4 強度予測器

本節ではタスク m に対する強度関数 $\lambda(t; \boldsymbol{z}^{(m)})$ について述べる．なお，強度関数は (4) で定義されるタスク表現に依存する．人の活動をモデル化する場合強度関数が周期的な変化を示すことが多い．そのため，強度関数を

$$\lambda\Big(t; \boldsymbol{z}^{(m)}\Big) = \lambda_{\mathrm{peri}}\Big(t; \boldsymbol{z}^{(m)}\Big) + \lambda_{\mathrm{aperi}}\Big(t; \boldsymbol{z}^{(m)}\Big), \tag{5}$$

のように**周期的強度関数** $\lambda_{\mathrm{peri}}(t; \boldsymbol{z}^{(m)})$ と**非周期的強度関数** $\lambda_{\mathrm{aperi}}(t; \boldsymbol{z}^{(m)})$ の和としてモデル化する．柔軟な強度関数のモデル化にはニューラルネットワークを用いることが有望であるものの，$\lambda_{\mathrm{peri}}(t; \boldsymbol{z}^{(m)})$ と $\lambda_{\mathrm{aperi}}(t; \boldsymbol{z}^{(m)})$ を直接ニューラルネットワークでモデル化すると，尤度 (2) に含まれる強度関数の積分計算が困難である．また，一般的な数値近似を用いることは精度低下や計算コストの課題が生じる．この課題を回避するため [9] と同様に区間 $[0, t]$ の強度関数の積分をニューラルネットワークを用いてモデル化する．すなわち，(5) の積分は

$$\begin{aligned}\Lambda\Big(t; \boldsymbol{z}^{(m)}\Big) &:= \int_0^t \lambda\Big(u; \boldsymbol{z}^{(m)}\Big) du \\ &= \Lambda_{\mathrm{peri}}\Big(t; \boldsymbol{z}^{(m)}\Big) + \Lambda_{\mathrm{aperi}}\Big(t; \boldsymbol{z}^{(m)}\Big),\end{aligned} \tag{6}$$

のように定義される．なお，$\Lambda_{\mathrm{peri}}(t; \boldsymbol{z}^{(m)})$ および $\Lambda_{\mathrm{aperi}}(t; \boldsymbol{z}^{(m)})$ はそれぞれ $\lambda_{\mathrm{peri}}(t; \boldsymbol{z}^{(m)})$ と $\lambda_{\mathrm{aperi}}(t; \boldsymbol{z}^{(m)})$ の累積関数である．累積関数はニューラルネットワークでモデル化されているため，累積関数を t に関して自動微分することにより，対応する強度関数を得ることができる．

周期的強度関数． 周期的な制約，すなわち，$\lambda_{\mathrm{peri}}(t; \boldsymbol{z}^{(m)}) = \lambda_{\mathrm{peri}}(t+\tau; \boldsymbol{z}^{(m)})$ を満たすように周期的強度関数を設計する．ここで，τ は事前知識によって決まる 1 週間や 1 日などの人の活動の周期である．強度関数は定義上非負であるため，その累積関数は t について単調増加関数である．そのため，周期的強度関数 $\Lambda_{\mathrm{peri}}(t; \boldsymbol{z}^{(m)})$ を MNN を用いて定義する．なお，MNN は $f_{\mathrm{peri}}(t, \boldsymbol{z}^{(m)}) : \mathbb{R}^{K_{\boldsymbol{z}}+1} \to \mathbb{R}$ のように表される．MNN は重みを非負の値に制限し，活性化関数を tanh や softplus などの単調増加関数に設定した全結合層であり，MNN の出力は入力に対して単調増加する．以上の条件から，$\Lambda_{\mathrm{peri}}(t; \boldsymbol{z}^{(m)})$ を

$$\begin{aligned}\Lambda_{\mathrm{peri}}\Big(t; \boldsymbol{z}^{(m)}\Big) &= s\Big(f_{\mathrm{peri}}\Big(t', \boldsymbol{z}^{(m)}\Big) - f_{\mathrm{peri}}\Big(0, \boldsymbol{z}^{(m)}\Big)\Big) \\ &\quad + s\left\lfloor \frac{t}{\tau} \right\rfloor \Big(f_{\mathrm{peri}}\Big(\tau, \boldsymbol{z}^{(m)}\Big) - f_{\mathrm{peri}}\Big(0, \boldsymbol{z}^{(m)}\Big)\Big),\end{aligned} \tag{7}$$

のように定義する．ここで，$t' = t - \tau\lfloor \frac{t}{\tau} \rfloor$ は周期的強度関数の位相であり，$\lfloor \cdot \rfloor$ は床関数である．s はスケールパラメータであり，学習データ $\mathcal{D}$ の統計量から定められる値であり，正規化のように値域を調整するために用いる．本稿では，学習データのクエリセットに含まれるイベント数の最大値 $\max_m \left|\mathcal{Q}^{(m)}\right|$ とした．なお，$f_{\mathrm{peri}}(0, \boldsymbol{z}^{(m)})$ を引くのは，$\Lambda_{\mathrm{peri}}(0; \boldsymbol{z}^{(m)}) = 0$ とするためである．

定理 4.1. $\lambda_{\mathrm{peri}}(t, \boldsymbol{z}^{(m)})$ は周期 τ の周期関数である．

証明. 定義および (7) より

$$\begin{aligned}&\lambda_{\mathrm{peri}}\Big(t, \boldsymbol{z}^{(m)}\Big) \\ &\quad= \frac{\partial \Lambda_{\mathrm{peri}}\Big(t; \boldsymbol{z}^{(m)}\Big)}{\partial t} \\ &\quad= \frac{\partial}{\partial t} s\Big(f_{\mathrm{peri}}\Big(t', \boldsymbol{z}^{(m)}\Big) - f_{\mathrm{peri}}\Big(0, \boldsymbol{z}^{(m)}\Big)\Big) \\ &\qquad+ \frac{\partial}{\partial t} s\left\lfloor \frac{t}{\tau} \right\rfloor \Big(f_{\mathrm{peri}}\Big(\tau, \boldsymbol{z}^{(m)}\Big) - f_{\mathrm{peri}}\Big(0, \boldsymbol{z}^{(m)}\Big)\Big), \\ &\quad= \frac{\partial}{\partial t} s f_{\mathrm{peri}}\Big(t', \boldsymbol{z}^{(m)}\Big) + \frac{\partial}{\partial t} s\left\lfloor \frac{t}{\tau} \right\rfloor f_{\mathrm{peri}}\Big(\tau, \boldsymbol{z}^{(m)}\Big),\end{aligned} \tag{8}$$

$$\begin{aligned}&\lambda_{\mathrm{peri}}\Big(t+\tau, \boldsymbol{z}^{(m)}\Big) \\ &\quad= \frac{\partial \Lambda_{\mathrm{peri}}\Big(t+\tau; \boldsymbol{z}^{(m)}\Big)}{\partial t} \\ &\quad= \frac{\partial}{\partial t} s f_{\mathrm{peri}}\left(t + \tau - \tau\left\lfloor \frac{t+\tau}{\tau} \right\rfloor, \boldsymbol{z}^{(m)}\right) \\ &\qquad+ \frac{\partial}{\partial t} s\left\lfloor \frac{t+\tau}{\tau} \right\rfloor f_{\mathrm{peri}}\Big(\tau, \boldsymbol{z}^{(m)}\Big) \\ &\quad= \frac{\partial}{\partial t} s f_{\mathrm{peri}}\left(t - \tau\left\lfloor \frac{t}{\tau} \right\rfloor, \boldsymbol{z}^{(m)}\right) \\ &\qquad+ \frac{\partial}{\partial t} s\left(\left\lfloor \frac{t}{\tau} \right\rfloor + 1\right) f_{\mathrm{peri}}\Big(\tau, \boldsymbol{z}^{(m)}\Big)\end{aligned} \tag{9}$$

(8) と (9) より

$$\begin{aligned}&\lambda_{\mathrm{peri}}\Big(t+\tau, \boldsymbol{z}^{(m)}\Big) - \lambda_{\mathrm{peri}}\Big(t, \boldsymbol{z}^{(m)}\Big) \\ &\quad= \frac{\partial}{\partial t} s f_{\mathrm{peri}}\Big(\tau, \boldsymbol{z}^{(m)}\Big) \\ &\quad= 0.\end{aligned} \tag{10}$$

したがって，$\lambda_{\mathrm{peri}}(t, \boldsymbol{z}^{(m)}) = \lambda_{\mathrm{peri}}(t+\tau, \boldsymbol{z}^{(m)})$ である．よって $\lambda_{\mathrm{peri}}(t, \boldsymbol{z}^{(m)})$ は周期 τ の周期関数．□

非周期的強度関数． この関数はニューラルネットワークで定義された強度関数である．周期的強度関数と同様に MNN $f_{\mathrm{aperi}}(t, \boldsymbol{z}^{(m)}) : \mathbb{R}^{K_{\boldsymbol{z}}+1} \to \mathbb{R}$ を用いて

$$\Lambda_{\mathrm{aperi}}(t; \boldsymbol{z}^{(m)}) = s\Big(f_{\mathrm{aperi}}\Big(t, \boldsymbol{z}^{(m)}\Big) - f_{\mathrm{aperi}}\Big(0, \boldsymbol{z}^{(m)}\Big)\Big), \tag{11}$$

のようにモデル化する．

4.5 メタ学習

ニューラルネットワークのパラメータを θ とし，$\Lambda_\theta(t; \boldsymbol{z}^{(m)})$ のように下付き文字で表記する．本章では，エピソード学習 [34] に基づいて最適なパラメータ θ の推定を行う．解くべき最

Algorithm 1: 提案手法の学習アルゴリズム

Input: データセット $\mathcal{D}$, 時間区間 $[T^{\mathrm{c}}, T^{\mathrm{e}}]$
Output: パラメータ θ

```
s を D から決定
while not done do
    M からタスク m をランダム抽出
    X^(m) をサポートセット S とクエリセット Q に分割
    (4) を用いて z を計算
    目的関数 (13) とその勾配を計算
    パラメータ θ を目的関数と勾配から更新
```

適化問題は

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{m\sim M}\Big[\mathbb{E}_{(\mathcal{S},\mathcal{Q})\sim X^{(m)}}[L_\theta(\mathcal{S},\mathcal{Q})]\Big], \qquad (12)$$

となる．なお，$\mathbb{E}[\cdot]$ は期待値を表し，添え字の m は省略されている．目的関数 $L_\theta(\mathcal{S},\mathcal{Q})$ は

$$L_\theta(\mathcal{S},\mathcal{Q}) = -\sum_{t\in\mathcal{Q}} \log \frac{\partial \Lambda_\theta(t;\boldsymbol{z})}{\partial t} + \Lambda_\theta(T^{\mathrm{e}};\boldsymbol{z}) - \Lambda_\theta(T^{\mathrm{c}};\boldsymbol{z}), \qquad (13)$$

である．第 2, 3 項は (6) から得られ，第 1 項は自動微分で得られる．学習アルゴリズムは Algorithm 1 のとおりである．なお，最適化の対象にはサポートセットも含める．すなわち，目的関数を $L_\theta(\mathcal{S},\mathcal{S}\cup\mathcal{Q})$ とする．これにより，T^{c} 付近の推定を安定化できる．

5 評価実験

5.1 データセット

提案手法の検証に用いた 2 つの実世界のデータ（Bikeshare, Taxi）及び地理的特徴について述べる．

Bikeshare. Bikeshare は NYC Citi Bike[1]によって収集された自転車の移動の記録である．このデータはニューヨーク市の 1,491 のバイクステーションで収集された 8 年間（2013 年 1 月 1 日–2021 年 1 月 31 日）のデータで，114,193,076 の移動が含まれている．全ての移動には自転車の貸出時刻とバイクステーションの id が記録されている．2020 年 3 月までに開設された 539 のステーションを学習データに，2020 年 4 月から 7 月に開設された 52 のステーションを検証データに，残りの 2020 年 8 月移行に開設された 76 のステーションを評価に用いた．観測開始時刻 t_0 はそのステーションが初めて利用された翌日の午前 0 時とし，予測時間区間は $[T^{\mathrm{c}}, T^{\mathrm{e}}] = [12\,\mathrm{hours}, 7\,\mathrm{days}]$ とした．各データ期間の最終週に開設されたステーションは期間の重複をさけるため除外した．RNN の入力のためにサポートセット内のイベントが 5 つ以下のステーションを除外した．

Taxi[36]**.** Taxi はニューヨーク市における 2014 年の 1 年間の 77,399,896 の移動から構成されるデータである．各移動はタクシーの乗車時刻と地域（GeoHash[2]）に紐づけられている．

表 1: データセットの統計量

データセット		$\lvert D\rvert$	$\lvert\mathcal{S}\rvert$ (M±SD)	$\lvert\mathcal{Q}\rvert$ (M±SD)
Bikeshare	train	539	10.5 ± 13.5	263.5 ± 295.1
	val.	52	6.3 ± 5.4	256.2 ± 253.7
	test	76	8.0 ± 10.2	240.9 ± 305.6
Taxi	train	970	33.6 ± 37.4	346.9 ± 346.2
	val.	276	35.1 ± 38.4	347.8 ± 343.7
	test	328	31.6 ± 30.8	364.2 ± 350.8

地域と期間に基づいて学習用，検証用，評価用にデータを分割した．具体的には地域を Geohash（レベル 7）に基づいて 970 個を学習用，276 個を検証用，328 個評価用に分割した．また，期間については学習用では 1 月から 5 月まで，検証用では 6 月，7 月以降をテスト用に割り当てた．各地域の該当期間から連続した 3 日間（午前 5 時から 3 日後の午前 5 時）のイベントデータを抽出し，データセットとした．予測時間区間は $[T^{\mathrm{c}}, T^{\mathrm{e}}] = [7\,\mathrm{hours}, 3\,\mathrm{days}]$ とした．また，サポートセットのイベント数が 5 つ以下の地域は除外した．

更に，各バイクステーションと GeoHash の地域に紐づける地理的特徴を Community District Profiles[3]から収集した．使用した特徴は，土地の利用方法，公共施設の数，犯罪数, 道の清潔さ, 平均通勤時間である．バイクステーションの緯度経度, GeoHash の地域の中心の緯度経度を基準点として対応付けを行った．

各データセットの統計量は表 1 のとおりである．なお，表中の M は平均，SD は標準偏差を表す．

5.2 比較手法

比較手法は以下のとおりである．

Proposed: 本稿で説明した提案手法である．RNN として 1 層 128 ユニットの双方向 LSTM を用いた．FNN は 1 層 128 ユニットで活性化関数を tanh とした．各表現の次元数は $K_{\boldsymbol{z}} = K_{\mathcal{S}} = 128$ とした．MNN は [9] と同様に，全ての重みを非負とし，隠れ層の活性化関数に tanh，出力層に softplus を用いた．MNN の重みの初期値は Glorot の一様分布 [37] に従って決定した．

HPP (Homogeneous Poisson Process) [20]: HPP は強度関数が一定の点過程モデルである．各タスクのサポートセットに基づいて，強度関数を $\lambda(t) = \left|\mathcal{S}^{(m^*)}\right|/T^{\mathrm{c}}$ のように決定した．

NNIPP (Neural Network Inhomogeneous Poisson process): NNIPP はニューラルネットワークベースの非定常ポアソン過程 [20] である．非定常ポアソン過程は時間に依存した強度関数であり，ネットワークは提案手法の非周期的成分と同一である（ただし，入力は時刻 t のみ）．全てのタスクを同一のタスクとみなして学習，予測を行う．強度関数および損失は (11) と (13) から $\boldsymbol{z}$ を取り除いたものである．

NM (NNIPP + MAML): NM は NNIPP にメタ学習のデ

1：https://ride.citibikenyc.com/system-data
2：http://geohash.org/site/tips.html
3：https://communityprofiles.planning.nyc.gov/

Algorithm 2: NM の学習アルゴリズム

Input: データセット $\mathcal{D}$, 時間区間 $[T^{\mathrm{c}}, T^{\mathrm{e}}]$, 内側ループ回数 l
Output: パラメータ θ

s から $\mathcal{D}$ を決定
while *not done* **do**
  M から m を抽出
  $X^{(m)}$ をサポートセット $\mathcal{S}$ とクエリセット $\mathcal{Q}$ に分割
  θ をタスク専用パラメータ θ' に代入
  repeat
    目的関数 (14) と (15) とその勾配を計算
    パラメータ θ' を目的関数と勾配から更新
  until l *times*
  目的関数 (16) と (17) とその二階微分を計算
  パラメータ θ を目的関数と二階微分から更新

ファクトスタンダード手法である MAML[34] を適用した手法である．すなわち，モデルは NNIPP と同一である．MAML ではメタ学習を実現するために，Algorithm 2 のように 2 重ループによる学習を行う．内側のループでは，未知のタスクに適応するため目的関数を

$$\hat{\theta}' = \arg\min_{\theta} E_{\mathcal{S}\sim\mathcal{D}}[L_\theta(\mathcal{S})] \tag{14}$$

$$L_\theta(\mathcal{S}) = -\sum_{t\in\mathcal{S}} \log\lambda_\theta(t) + \Lambda_{\hat{\theta}'}(T^{\mathrm{c}}) - \Lambda_{\hat{\theta}'}(0), \tag{15}$$

としてパラメータの更新を行う．本稿では，内側のループ回数を l と表記し，$l = \{1, 2, 3, 4\}$ とした．また，外側の目的関数は

$$\hat{\theta} = \arg\min_{\theta} E_{(\mathcal{S},\mathcal{Q})\sim\mathcal{D}}[L_{\hat{\theta}'}(\mathcal{Q})] \tag{16}$$

$$L_{\hat{\theta}'}(\mathcal{Q}) = -\sum_{t\in\mathcal{Q}} \log\lambda_{\hat{\theta}'}(t) + \Lambda_{\hat{\theta}'}(T^{\mathrm{e}}) - \Lambda_{\hat{\theta}'}(T^{\mathrm{c}}), \tag{17}$$

である．

なお，点過程にメタ学習を組み込んだ既存手法である HARMLESS[17] はタスク間の関連を示すネットワークが必要であり，適用できないため比較を行っていない．

5.3 実験設定

実装と環境． アルゴリズムの実装には Python 3.9.5 と Pytorch 1.10.2[38] を用いた．各実験は Xeon Platinum 8176 (2.10GHz) と NVIDIA TITAN V で実行した．また，MAML の実装には higher 0.2.1[39] を用いた．

最適化． 各手法の最適化には Adam [40] ($\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$) を用いた．なお，HPP は勾配法を必要としないため Adam を使用していない．

評価指標． 本稿では，精度比較のために 2 つの指標を用いた．1 つ目の評価指標はテストデータに対する負の対数尤度（test nll，(13)）である．2 つ目の評価指標は平均二乗誤差（MSE）である．誤差を計算するため，時間区間 $[T^{\mathrm{c}}, T^{\mathrm{e}}]$ を均一に J 分割（本稿では $J = 100$）し，各区間 $\{[T_j^{\mathrm{a}}, T_j^{\mathrm{b}}]\}_{j=1}^{J}$ において予測イベント数と実際のイベント数の比較を

表 2: ハイパーパラメータの探索範囲

ハイパーパラメータ	範囲
バッチサイズ	{4, 8, 16, 32}
MNN のユニット数（周期的強度関数あり）	{128, 256, 512}
MNN のユニット数（周期的強度関数なし）	{256, 512, 1024}
FNN の層の数	{1, 2}

表 3: 各手法の精度 (低い方が高精度).

	Bikeshare		Taxi	
	test nll	MSE	test nll	MSE
HPP	111.15	24.30	-347.0	23.12
NNIPP	137.86	24.23	-288.3	25.15
NM($l = 1$)	84.22	21.33	-332.3	20.99
NM($l = 2$)	54.29	18.39	-351.0	19.79
NM($l = 3$)	61.36	19.61	-344.5	19.33
NM($l = 4$)	94.29	21.69	-352.0	19.27
Proposed	**-9.83**	**14.22**	**-400.0**	**15.34**

$$\mathrm{MSE} = \frac{1}{J}\sum_{j=1}^{J}\left[\left|X_{T_j^{\mathrm{b}}}^{(m^*)}\right| - \left|X_{T_j^{\mathrm{a}}}^{(m^*)}\right| - \int_{T_j^{\mathrm{a}}}^{T_j^{\mathrm{b}}} \hat{\lambda}(u)du\right]^2, \tag{18}$$

のように行った．各区間の長さは Bikeshare では 1.56 hours, Taxi では 39 minutes となった．

ハイパーパラメータ． 全ての手法についてハイパーパラメータをグリッドサーチによって選択した．具体的には，100 エポックの学習を行い，全てのハイパーパラメータの組み合わせと全てのエポックにおける検証用データセットに対する尤度に基づいてモデルを選択した．探索した範囲は表 2 のとおりである．なお，周期的強度関数の有無で探索範囲が異なるのは，周期的強度関数がある場合 MNN が 2 つになり，周期的強度関数がない場合と比較したとき MNN のパラメータ数が 2 倍になるためである．

5.4 定量評価

各データセットに対する予測精度は表 3 のようになった．提案手法の精度が NM に勝ることがわかる．また，Bikeshare で $l = 4$ のときに $l = 3$ よりも低い精度を示していることがわかる．これは，l を増加させても必ずしも精度が向上しないことを示す．NNIPP は予測対象タスクのサポートセットを用いず，全てのタスクに対して同じ予測出力を行うため，最も精度が悪かった．HPP は単純であるものの，予測対象タスクのサポートセットを用いて強度関数の大きさを調整する．そのため，HPP の精度は NNIPP より優れている．しかし，HPP はイベント数しか考慮しないため，NM や提案手法よりは劣る結果となった．表 4 は提案手法と NM による学習および予測時間である．提案手法は学習に NM($l = 2$) と同程度の時間がかかり，予測に 2 倍弱の時間がかかることがわかる．しかし，入出力が 1 日単位や時間単位であることから，実行速度は許容範囲であると考えられる．

表 4: エポック当たりの学習時間と予測時間.

データセット	手法	学習 (s)	予測 (s)
Bikeshare	NM($l=1$)	25.69	4.08
	NM($l=2$)	36.42	4.59
	NM($l=3$)	49.69	5.95
	NM($l=4$)	59.12	5.97
	Proposed	34.30	8.58
Taxi	NM($l=1$)	53.40	16.21
	NM($l=2$)	86.75	20.77
	NM($l=3$)	102.86	22.88
	NM($l=4$)	127.71	26.21
	Proposed	88.67	37.49

表 5: 外的要因による予測性能の変化

	Bikeshare		Taxi	
	test nll	MSE	test nll	MSE
外的要因なし	67.14	19.50	-381.0	16.16
地理的特徴のみ	21.04	16.59	-359.4	19.45
周期性のみ	49.2	19.02	-367.3	18.17
外的要因あり	**-9.83**	**14.22**	**-400.0**	**15.34**

表 6: 地理的特徴別の予測性能比較

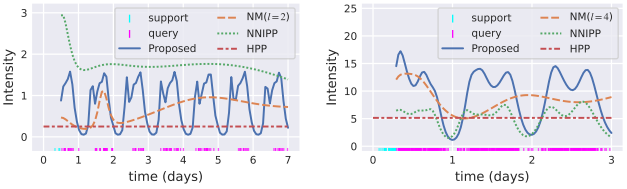
入力地域特徴	Bikeshare		Taxi	
	test nll	MSE	test nll	MSE
なし	49.2	19.02	-367.3	18.17
土地の利用方法	24.26	16.35	-377.4	17.05
公共施設の数	27.18	16.85	-379.5	16.41
その他 [4]	22.48	19.14	-399.4	15.11
すべて	**-9.83**	**14.22**	**-400.0**	**15.34**

5.5 アブレーション研究

各外的要因（周期性，地理的特徴）の効果を検証するため，提案手法の外的要因の有無を変化させて評価した結果は表 5 のとおりである．Bikeshare では，各要因を除外することにより，精度が低下し，両方の要因を低下すると更に精度が低下することがわかる．これは，外的要因を取り入れた提案手法の有効性を示唆している．一方，Taxi では外的要因なしの精度が地域特徴のみや周期性のみを考慮した手法よりも精度が高いことがわかる．しかし，双方を考慮した手法の精度が最も良い．そのため，これらの外的要因を組み合わせて考慮することが有効であることを示唆している．

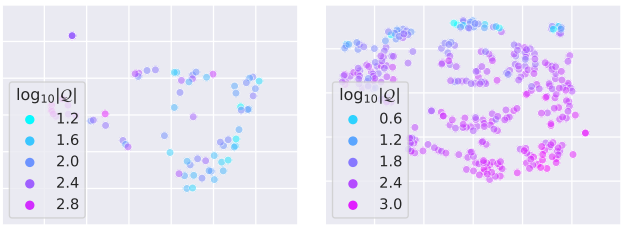
表 6 は提案手法について各地理的特徴量のみを用いた場合の精度である．表 6 からどの地理的特徴量についても用いたほうが精度が向上することがわかる．また，全ての地理的特徴量を組み合わせた場合に更に精度が向上することがわかる．このことから，全ての地理的特徴量がイベント予測に有用であることが確認された．

4：犯罪数，道の清潔さ，平均通勤時間



(a) Bikeshare (b) Taxi

図 2: 強度関数の予測結果



(a) Bikeshare (b) Taxi

図 3: t-SNE によるタスク表現 $\boldsymbol{z}$ の可視化 [5]

5.6 定性評価

図 2 は各手法によって予測された強度関数である．なお，図中下部のシアンのスパイクはサポートセット中のイベントを表し，マゼンタのスパイクはクエリセット中のイベントを表している．図 2a において，提案手法は周期的なパターンを捉えて予測できていることがわかる．一方，MN は 4 日目以降，1 日の中で強度関数が大きく変化せず，周期的なパターンを捉えていないことがわかる．図 2b から，提案手法が朝と夕方の 1 日 2 回のピーク（通勤ラッシュ）を持つ周期的なパターンを捉えていることがわかる．一方，NM は周期的なパターンを考慮しないため，ピーク時に大きな予測誤差が発生する．

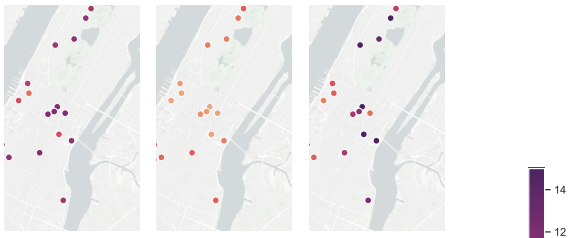
タスク表現エンコーダの有効性を確認するため，タスク表現 $\boldsymbol{z}$ を t-SNE[41] により可視化した結果が図 3 である．図 3 からタスク表現はクエリセットのイベント数を反映していることがわかる．この結果は，タスク表現エンコーダがイベントデータの初期（サポートセット）と将来の長期間（クエリセット）の関係を捉えられていることを示している．

図 4 はニューヨークのマンハッタンにあるバイクステーションを対象に 2 つの時間帯における貸出数の予測値と実際の値を示したものである．提案手法は NM と異なり，ピーク時とオフピーク時の両方の時間帯の需要の変化を正確に予測できていることがわかる．

6 まとめ

本稿では，短期間のイベントデータから将来のイベントを予測する点過程のための新しいメタ学習手法を提案した．提案手法のタスク表現エンコーダは RNN により短期間のイベントデータをタスク表現に埋め込む．MNN はタスク表現を入力として受け取り，各タスク用にあわせて強度関数をモデル化する．

5：点の色はクエリセットのイベント数を示す．



Proposed NM Ground truth

(a) ピーク時 (t_0 から 6 日目の午後 5 時台)

Proposed NM Ground truth

(b) オフピーク時 (t_0 から 5 日目の午後 9 時台)

図 4: マンハッタンでの各ステーションの貸出数

更に，人の活動に大きく影響を与える要因である地理的特徴と時間的な周期をそれぞれタスク表現，MNN に組み込むための拡張を行った．複数の実データを用いた実験により，提案手法は複数の既存手法よりも高い予測性能を持つことが確認された．今後の課題は，提案手法により，画像などのより複雑な特徴を考慮できるようにすることである．

文　献

[1] Junbiao Pang, Jing Huang, Xue Yang, Zuyun Wang, Haitao Yu, Qingming Huang, and Baocai Yin. Discovering fine-grained spatial pattern from taxi trips: Where point process meets matrix decomposition and factorization. *IEEE trans Intell Transp Syst*, 2017.

[2] Daniel Gervini and Manoj Khanal. Exploring patterns of demand in bike sharing systems via replicated point process models. *J R Stat Soc Ser C Appl Stat*, 2019.

[3] George O Mohler, Martin B Short, P Jeffrey Brantingham, Frederic Paik Schoenberg, and George E Tita. Self-exciting point process modeling of crime. *J Am Stat Assoc*, 2011.

[4] Edward Choi, Nan Du, Robert Chen, Le Song, and Jimeng Sun. Constructing disease network and temporal progression model via context-sensitive hawkes process. In *ICDM*, 2015.

[5] Cyrille Médard De Chardon, Geoffrey Caruso, and Isabelle Thomas. Bike-share rebalancing strategies, patterns, and purpose. *J Transp Geogr*, 2016.

[6] Junchi Yan. Recent advance in temporal point process: from machine learning perspective. *SJTU Technical Report*, 2019.

[7] Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural temporal point processes: A review. *arXiv*, 2021.

[8] 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 *KDD*, 2016.

[9] Takahiro Omi, Naonori Ueda, and Kazuyuki Aihara. Fully neural network based model for general temporal point processes. *arXiv*, 2019.

[10] Tomoharu Iwata and Atsutoshi Kumagai. Few-shot learning for time-series forecasting. *arXiv*, 2020.

[11] Mauro Ribeiro, Katarina Grolinger, Hany F ElYamany, Wilson A Higashino, and Miriam AM Capretz. Transfer learning with seasonal and trend adjustment for cross-building energy forecasting. *Energy and Buildings*, 2018.

[12] Thiyanga S Talagala, Rob J Hyndman, George Athanasopoulos, et al. Meta-learning how to forecast time series. *Monash Econometrics and Business Statistics Working Papers*, 2018.

[13] Ricardo BC Prudêncio and Teresa B Ludermir. Meta-learning approaches to selecting time series models. *Neurocomputing*, 2004.

[14] Christiane Lemke and Bogdan Gabrys. Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 2010.

[15] Ali Hooshmand and Ratnesh Sharma. Energy predictive models with limited data using transfer learning. In *ACM e-Energy*, 2019.

[16] Abbas Raza Ali, Bogdan Gabrys, and Marcin Budka. Cross-domain meta-learning for time-series forecasting. *Procedia Comput. Sci.*, 2018.

[17] Yujia Xie, Haoming Jiang, Feng Liu, Tuo Zhao, and Hongyuan Zha. Meta learning with relational information for short sequences. *NeurIPS*, 2019.

[18] Chin-Yu Huang, Michael R. Lyu, and Sy-Yen Kuo. A unified scheme of some nonhomogenous poisson process models for software reliability estimation. *IEEE Trans. Softw. Eng.*, 2003.

[19] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *MML*, 2015.

[20] J.F.C. Kingman. *Poisson Processes*. John Wiley & Sons, Ltd, 2005.

[21] Alan G Hawkes. Point spectra of some mutually exciting point processes. *J. R. Stat. Soc., Ser. B, Methodol.*, 1971.

[22] Benedek Kovács. Weighted Fair Resource Sharing Without Queuing Delay. In *ICNS*, 2011.

[23] Alexei Pozdnoukhov and Fergal Walsh. Exploratory novelty identification in human activity data streams. In *IWGS*, 2010.

[24] R Dean Malmgren, Jake M Hofman, Luis AN Amaral, and Duncan J Watts. Characterizing individual communication patterns. In *KDD*, 2009.

[25] Hongyuan Mei and Jason Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *arXiv*, 2016.

[26] Yulong Gu. Attentive neural point processes for event forecasting. In *AAAI*, 2021.

[27] Aditya Krishna Menon and Young Lee. Predicting short-term public transport demand via inhomogeneous poisson processes. In *CIKM*, 2017.

[28] Jianzhu Guo, Xiangyu Zhu, Chenxu Zhao, Dong Cao, Zhen Lei, and Stan Z Li. Learning meta face recognition in unseen domains. In *CVPR*, 2020.

[29] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *ICCV*, 2019.

[30] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *CoRL*, 2017.

[31] Scott Reed, Yutian Chen, Thomas Paine, Aäron van den Oord, SM Eslami, Danilo Rezende, Oriol Vinyals, and Nando de Freitas. Few-shot autoregressive density estimation: Towards learning to learn distributions. *arXiv*, 2017.

[32] Danilo Rezende, Ivo Danihelka, Karol Gregor, Daan Wierstra, et al. One-shot generalization in deep generative models. In *ICML*, 2016.

[33] Jörg Bornschein, Andriy Mnih, Daniel Zoran, and Danilo J Rezende. Variational memory addressing in generative models. *arXiv*, 2017.

[34] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[35] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv*, 2018.

[36] NYC OpenData. 2014 yellow taxi trip data. `https://data.cityofnewyork.us/Transportation/2014-Yellow-Taxi-Trip-Data/gkne-dk5s`, 2015. Accessed: 2022-08-14.

[37] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019.

[39] Edward Grefenstette, Brandon Amos, Denis Yarats, Phu Mon Htut, Artem Molchanov, Franziska Meier, Douwe Kiela, Kyunghyun Cho, and Soumith Chintala. Generalized inner loop meta-learning. *arXiv*, 2019.

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[41] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *JMLR*, 2014.