

An Investigation into Multi-Task Learning for Point-of-Interest Category Classification and Next-POI Prediction

Vitor H. O. Silva, Ingrid F. Almeida, Tarik S. Paiva, Germano B. Santos, Fabrício A. Silva, Felipe T. Sousa
Laboratório de Inteligência em Sistemas Pervasivos e Distribuídos (NESPED-LAB), UFV, Florestal, Brazil
E-mail: {vitor.h.oliveira, ingred.almeida, tarik.paiva, germano.santos, fabricio.asilva, felipe.t.sousa}@ufv.br

Abstract—Multi-Task Learning (MTL) has shown significant promise across various domains by improving generalization and robustness through shared representations. This work explores MTL within Location-Based Social Networks (LBSNs), focusing on two complementary tasks: POI Category Classification and Next-POI Prediction. We propose a joint MTL architecture that shares lower-level embeddings and sequence encoders while maintaining task-specific heads. Through experiments on the Gowalla LBSN dataset, we compare our MTL model against established single-task baselines. While our model demonstrated strong performance in POI category classification, outperforming the Human Mobility Representation Model (HMRM), the results for next-POI prediction were more competitive against MHA+PE. Crucially, our findings indicate that the multi-task learning approach did not consistently yield substantial improvements over the single-task baselines across both tasks, with many performance differences being minor and often within statistical standard deviations. Furthermore, contrary to expectations, the MTL model required significantly more computational resources and wall time to converge compared to the cumulative effort of individual single-task models. Our contributions include the introduction of a unified MTL framework for these POI-related tasks, the design of novel task-specific heads, and a detailed analysis of performance and convergence characteristics, highlighting observed limitations of MTL in this specific application. This research provides insights into the practical challenges of applying MTL, emphasizing the need for careful architectural design and optimization in achieving its theoretical benefits.

Index Terms—Multi-Task Learning, Point-of-Interest (POI), Location-Based Recommendation, Sequential Modeling

I. INTRODUCTION

Multi-Task Learning (MTL) is a machine learning paradigm where multiple related tasks are learned jointly, sharing representations and inductive biases to improve generalization performance [1]. By leveraging shared information, MTL can potentially mitigate overfitting and yield more robust feature extractors compared to single-task approaches. Its success in domains like computer vision [2], natural language processing [3] and recommendation systems [4] motivates its application to Location-Based Social Networks (LBSNs), where understanding user interactions with Points-of-Interest (POIs) – specific physical locations that someone might find useful or interesting, such as restaurants, shops, or landmarks - is fundamental.

In this work, we investigate the application of MTL to two critical POI-related tasks:

- 1) **POI Category Classification**: Classify the semantic category (e.g., shopping, travel) of a POI based on its features and context within a user’s trajectory.
- 2) **Next-POI Prediction**: Predicting the category of the next POI a user will visit, given the sequence of their prior visits.

These tasks are ostensibly related, as both draw upon the same underlying user mobility data. Accurate POI category representations could inform next-location predictions, and conversely, sequential patterns could provide context for category classification. However, the tasks possess fundamentally different natures. POI Category Classification is primarily a **static classification task** that relies on the intrinsic, context-rich features of individual locations. In contrast, Next-POI Prediction is a **dynamic, sequential task** that depends on capturing temporal patterns and transitions.

This dichotomy raises a critical question: can a single, shared representation effectively serve two tasks with such distinct underlying characteristics? The assumption that MTL will be beneficial is not guaranteed. It is possible that forcing a shared encoder to learn features for both a static and a sequential task could result in **negative transfer**, where the joint training process actually hinders performance by learning a “compromise” representation that is not specialized enough for either task.

Therefore, the central hypothesis of this study is that a standard hard parameter-sharing MTL architecture will face significant limitations when applied to this specific task pairing, failing to achieve substantial and consistent improvements over well-tuned, specialized single-task models. Our work is thus framed as an empirical investigation into the boundary conditions of MTL’s effectiveness, exploring whether the presumed task relatedness is sufficient to overcome their inherent structural differences.

To test this hypothesis, we conducted experiments on the Gowalla LBSN dataset [5], [6], comparing our MTL model against strong single-task baselines (HMRM [7] and MHA+PE [8]). Our results lend weight to our hypothesis. While the proposed models perform well, particularly in POI category classification, the MTL framework does not deliver the consistent, significant gains often expected. The performance differences between the MTL and single-task models were frequently marginal and fell within standard deviations, suggesting their

statistical performance was largely comparable. This outcome indicates that, for this problem, the architectural constraints and task dissimilarities may have offset the potential benefits of joint learning.

The main contributions of this paper are as follows:

- We introduce an MTL framework to simultaneously address POI category classification and next-POI prediction.¹
- We design task-specific heads to capture category co-occurrence and sequential transition dynamics, respectively.
- We provide a critical analysis of the limitations of a hard parameter-sharing MTL approach for this task pairing. Our findings highlight that significant task dissimilarity can challenge the effectiveness of MTL, serving as a case study on the importance of task compatibility in model design.

The rest of this paper is organized as follows. In Section II, we review the theoretical foundations of MTL and prior work on POI modeling. Section III describes our joint architecture, data preprocessing, and training protocol. Section IV presents experimental results. Finally, Section V summarizes our findings and discusses future research directions.

II. THEORETICAL FOUNDATIONS AND RELATED WORK

A. POI Classification and Next-POI Prediction

POI Category Classification refers to the task of inferring the semantic category of a location based on contextual information such as geographic coordinates, visit frequency, or user behavior.

Next-POI Prediction, in contrast, aims to predict which specific location a user is likely to visit next, given their past movement history. It is a sequential task that requires modeling temporal and spatial patterns within user trajectories.

The prediction and classification of POIs are fundamental challenges in applications such as recommendation systems and urban planning. Xu et al. [9] propose the Tree-guided Multi-task Embedding (TME) model, which aims to improve the semantic annotation of POIs using a hierarchical structure of categories and mobility-based representation learning.

POI Category Classification has been extensively addressed in location-based recommendation systems, with various techniques applied to capture patterns and improve prediction accuracy. In this context, Lim et al. [10] propose the Hierarchical Multi-Task Graph Recurrent Network (HMT-GRN), which uses multi-task learning to simultaneously predict the next POI and its geographic region.

Although these approaches have made significant progress in the POI classification and next-location prediction fields, they tend to focus only on single-task objectives. None of the cited models propose a unified multi-task learning (MTL) framework that jointly optimizes both POI category classification and next-POI prediction tasks. This motivates the

development of solutions that leverage shared representations across related POI tasks.

B. Multi-Task Learning

MTL was formalized by Caruana [1] as an inductive transfer mechanism that enhances generalization by learning shared representations across related tasks. Early neural architectures employed *hard parameter sharing*, where a common set of hidden layers is used for all tasks, yielding strong regularization and reduced training time.

Recent surveys [11], [12] organize contemporary MTL research along five methodological dimensions: (i) *parameter sharing* (hard vs. soft); (ii) *relationship learning* (discovering task affinity or hierarchy); (iii) *feature routing* (e.g., cross-stitch, sluice networks, attention gating); (iv) *optimization* (conflict-aware gradient techniques); and (v) *pre-training and instruction tuning*. MTL has evolved from small, homogeneous task sets to dozens of heterogeneous objectives spanning computer vision, natural language processing, and recommender systems.

A fundamental architectural challenge in MTL is deciding *how much, where, and when* to share parameters across tasks. Three canonical sharing schemes have emerged:

Hard Parameter Sharing: A single encoder is shared by all tasks, followed by task-specific heads. This remains the simplest and most popular baseline, providing effective regularization [1].

Soft Parameter Sharing: Each task has its own encoder, but networks exchange information through learned cross-connections, such as Cross-Stitch units [13] or Sluice networks [14].

Mixture-of-Experts (MoE): Multi-Gate Mixture-of-Experts (MMoE) architectures maintain a pool of shared expert subnetworks; each task learns a gating function to combine experts, outperforming monolithic sharing in large-scale recommendation settings [15], [16].

To mitigate task interference and loss imbalance, several optimization techniques have been proposed: GradNorm equalizes gradient magnitudes across tasks [17]; MGDA finds Pareto-optimal descent directions [18]; PCGrad removes conflicting gradient components [19]; and Dynamic Weight Averaging (DWA) reweights losses based on their relative learning speeds [20].

Despite these successes, MTL still faces several key challenges:

- **Negative Transfer** — Unrelated or adversarial tasks can degrade shared representations, harming individual task performance [12], [14].
- **Gradient Conflict** — Simultaneous optimization may produce opposing gradient directions, slowing or destabilizing convergence [18], [19].
- **Data Heterogeneity** — Variations in modality, label granularity, and dataset size complicate sampling strategies and minibatch construction [21], [22].

¹The complete source code is publicly available at <https://github.com/VitorHugoOli/PoiMtlNet> for full reproducibility.

- **Scalability** — Routing complexity, memory footprint, and evaluation costs often grow super-linearly as the number of tasks increases [11], [12].

Recent work addresses these issues via task clustering [22], dynamic curricula, conflict-aware optimizers, and parameter-efficient adapters [11]. Nevertheless, principled criteria for task grouping, theoretical guarantees of Pareto efficiency, and energy-efficient training of very large MTL foundation models remain open research directions.

C. Multi-task learning applied in POI

MTL has been explored in problems related to Points of Interest (POIs), primarily due to its ability to improve generalization by leveraging shared information across correlated tasks. In POI modeling scenarios, such as user trajectory prediction and recommendation, MTL allows the learning of complementary objectives simultaneously, allowing for more accuracy.

Early MTL-based approaches such as MCARNN [23] employ recurrent neural networks with temporal attention mechanisms to jointly predict user activities and future visited locations. Similarly, the iMTL framework [24] uses an LSTM architecture to model next-activity prediction, incorporating temporal dynamics in user behavior modeling.

More recent works have begun incorporating contextual and semantic signals. TLR-M [25], for example, applies Transformer-based architectures to simultaneously predict the next POI and queue waiting time, demonstrating the value of attention-based models in capturing spatio-temporal dependencies. MTPR [26] combines LSTMs and adversarial learning to address uncertainty in check-ins and improve multi-task POI recommendation both location and temporal context with a generative component.

Despite these advances, existing MTL approaches often focus on joint modeling of next-POI prediction and temporal or behavioral signals, while the relationship between POI category classification and next-POI prediction remains underexplored. Some Models such as TME [9] address category annotation using graph-based encoders, but treat prediction and classification separately.

This creates a gap in the literature regarding the joint modeling of semantic and sequential POI tasks. In contrast, our proposed model introduces a unified MTL framework that jointly performs POI category prediction and next-POI classification using a hard parameter-sharing architecture. By modeling both tasks simultaneously, our approach promotes knowledge transfer across shared spatio-temporal representations, enabling more robust predictions even in data-sparse environments.

III. METHODOLOGY

This section details the proposed methodology for Point-of-Interest (POI) category classification and next-POI prediction using a MTL approach. We first describe the generation of POI embeddings and the preparation of data for both tasks. Subsequently, we present the architectures of the single-task

baselines, which are also used as task-specific heads within our MTL framework. Finally, we elaborate on the proposed MTL architecture, including the hard parameter-sharing strategy and the Nash-MTL optimizer employed for gradient aggregation.

A. POI Embeddings and Data Preparation

This work computes POI embeddings using graph attention convolution along with contrastive learning based on Deep Graph Infomax (DGI). These embeddings are the input for the MTL model discussed in Section III-B.

1) *POI Embeddings*: A point of interest is defined as a tuple $\langle id, lat, long, cat \rangle$, where id serves as a unique identifier, lat and $long$ represent its coordinates, and cat denotes its category, such as restaurant or pharmacy. A place suffers from complementarity effect [27], implying that a location is affected by its neighbors (e.g. leisure zones are often near residential districts). Hence, we propose a graph-based contrastive method to capture this spatial locality behavior, generating a POI embedding to serve as input to our MTL model.

To capture the spatial locality and capture neighborhood information, we model the weighted graph $G(V, E)$ as a Delaunay Triangulation, where each vertex $v \in V$ represents a POI and edge $e_{ij} \in E$ defines the connection between POI p_i and a POI p_j . Besides that, following [28] the weight of an edge e_{ij} is defined as $w_{ij} = \log((1 + D^{1.5}/1 + d_{ij}^{1.5}))$, where D is the diagonal length of bounding box that encloses the coordinates of POIs, and d_{ij} is the geodesic distance between p_i and p_j . The node feature matrix is based on category one-hot encoding of each POI, represented by $X \in \mathbb{R}^{n_c}$, where n_c is the number of possible categories.

Therefore, to encode the Delaunay graph we apply a graph attention convolution layer [29], yielding an embedding $h_i \in \mathbb{R}^d$, where d is the dimension of the output. Moreover, we utilize a Deep Graph Infomax (DGI) [30] as a contrastive learning method to maximize the mutual information between local representations (node embeddings) and the global graph embedding compared to a corrupted version of a graph, which is guided by the Equation 1.

$$\mathcal{L} = \frac{1}{2|V|} \left(\sum_{i=1}^{|V|} \mathbb{E}_{(X,G)} [\log \mathcal{D}(\vec{h}_i, \vec{s})] + \sum_{j=1}^{|V|} \mathbb{E}_{(\tilde{X},G)} [\log(1 - \mathcal{D}(\tilde{h}_j, \vec{s}))] \right) \quad (1)$$

where X and G denote the positive samples for the node feature matrix and graph adjacency, respectively, while \tilde{X} is the corrupted version of the feature matrix. To corrupt the features, we employ a random shuffling method. Additionally, \vec{s} is the global graph embedding derived by applying global mean pooling to the embedding matrix $H = [h_1, h_2, \dots, h_n]$ in our model. Moreover, \mathcal{D} is a discriminator that differentiates each positive example from its corrupted version, learning from their dissimilarities; in our POI Embedding, this discriminator is implemented as a linear model.

2) *Data Processing for POI Category Classification:* Each POI is represented by its 64-dimensional DGI embedding $e \in \mathbb{R}^{64}$. The dataset therefore consists of pairs (e, c) , where c is the POI’s ground-truth category to be predicted.

3) *Data Processing for Next-POI Category Prediction:* This task predicts the category of the next POI a user will visit from their recent check-in history.

- 1) **Trajectory building.** Check-ins (user, POI, timestamp) are ordered chronologically per user; users with fewer than five visits are discarded.
- 2) **Sequence extraction.** From each trajectory we draw non-overlapping windows of length $L_h = 9$: (p_1, \dots, p_9) as context and p_{target} as the next POI.
- 3) **Feature/label construction.**
 - The input is the concatenation of the 64-dimensional embeddings of p_1 – p_9 , yielding a $9 \times 64 = 576$ -dim vector. Histories shorter than nine are padded by a special POI whose embedding is $\mathbf{0}$.
 - The label is the category of p_{target} .

This format allows the model to infer the category of the next POI solely from the embeddings of the previous visits.

B. Multi-Task Learning Architecture

Our proposed Multi-Task Learning (MTL) architecture, depicted in Figure 1, is built upon a hard parameter-sharing scheme for the joint training of POI Category Classification and Next-POI Prediction. This design integrates Feature-wise Linear Modulation (FiLM) layers to condition the shared representations, a technique aimed at modulating task interactions and mitigating negative transfer [22], [31].

Architecture Overview: Input features for POI Category Classification ($\mathbf{x}^{(c)}$) and Next-POI Prediction ($\mathbf{x}^{(n)}$) are first processed by separate, task-specific encoders. These encoders, implemented as MLPs, map the raw input features into a shared latent space of dimension d_{shared} .

Feature-wise Linear Modulation (FiLM): To allow for task-specific adaptation within the shared pathway, we employ FiLM layers [31]. A unique, learnable task embedding \mathbf{e}_t for each task t is used to generate a scaling vector γ_t and a shifting vector β_t . These vectors modulate the encoded features $\mathbf{h}_{\text{enc}}^{(t)}$ from each task-specific encoder:

$$\text{FiLM}(\mathbf{h}_{\text{enc}}^{(t)} | \mathbf{e}_t) = \gamma_t \odot \mathbf{h}_{\text{enc}}^{(t)} + \beta_t$$

This lightweight operation conditions the features before they enter the fully shared layers of the network.

Shared Processing Layers: The FiLM-modulated representations are processed by a sequence of shared layers. This module begins with a linear transformation followed by a LeakyReLU activation, LayerNorm, and dropout, and continues through L_{shared} residual blocks. This hard parameter-sharing approach regularizes the model by forcing it to learn a common, robust feature representation for both tasks [32].

Task-Specific Heads: Finally, the processed shared representations are passed to dedicated, unshared task-specific heads, as detailed in Section IV-A. These heads generate

the final predictions for each task, allowing for specialized output generation and isolated gradient computation during backpropagation.

Rationale for Hard Parameter Sharing: The choice of hard parameter sharing is motivated by its efficiency and regularization benefits, which are critical for our target application. This approach is supported by several factors:

- **Computational Efficiency:** Unlike soft-sharing methods that often introduce task-specific parameters and increase computational overhead, hard sharing maintains a compact model. This is crucial for deployment on resource-constrained edge devices (e.g., NVIDIA Jetson platforms).
- **Implicit Regularization:** By constraining the hypothesis space, hard sharing acts as a regularizer, often leading to more generalizable models, especially when tasks are related [14]. Seminal work by Caruana demonstrated significant error reduction with this approach [1].
- **Empirical Performance:** In practice, hard parameter sharing frequently matches or exceeds the performance of more complex architectures on many benchmarks, while offering faster training and inference [22].

This architecture strategically balances model capacity with knowledge transfer, prioritizing efficiency for potential real-time recommendation scenarios.

C. Nash-MTL Optimizer

MTL necessitates a unified update direction that benefits all tasks, even when their gradients conflict. Nash-MTL formulates this as a cooperative K -player bargaining game [21], where each task acts as an “agent” seeking to maximize its utility (loss reduction). The optimizer selects a step that is (i) beneficial to all tasks, (ii) scale-invariant to arbitrary loss reweightings, and (iii) proportionally fair, ensuring no alternative direction increases one task’s utility without decreasing the product of all utilities [21]. The unique solution satisfying these axioms is the Nash Bargaining Solution (NBS). This approach allows Nash-MTL to “negotiate” at each iteration, yielding a compromise descent direction that improves every loss while maximizing the joint progress of the entire system.

Formal Formulation.: Given task-specific objectives $\{\mathcal{L}_k(\boldsymbol{\theta})\}_{k=1}^K$ and their gradients $\mathbf{g}_k = \nabla_{\boldsymbol{\theta}} \mathcal{L}_k$ at current parameters $\boldsymbol{\theta}$, Nash-MTL frames gradient aggregation as a bargaining game to determine the common update direction [21]. The utility for each task k is its signed improvement, $u_k(\Delta\boldsymbol{\theta}) = \mathbf{g}_k^T \Delta\boldsymbol{\theta}$, where $\Delta\boldsymbol{\theta}$ is the parameter update within an ε -ball [21]. The NBS is found by maximizing the weighted geometric mean of utilities:

$$\Delta\boldsymbol{\theta}^* = \arg \max_{\Delta\boldsymbol{\theta} \in \mathcal{B}_\varepsilon} \sum_{k=1}^K \log(u_k(\Delta\boldsymbol{\theta})) \quad \text{s.t. } u_k(\Delta\boldsymbol{\theta}) > 0, \forall k. \quad (2)$$

By setting $\Delta\boldsymbol{\theta} = \sum_k \alpha_k \mathbf{g}_k$ with positive weights α , the solution simplifies to solving the nonlinear system $(\mathbf{G}^T \mathbf{G}) \boldsymbol{\alpha} = \boldsymbol{\alpha}^{-1}$ [21], where $\mathbf{G} = [\mathbf{g}_1 \dots \mathbf{g}_K]$ [21]. The parameter update is then $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \mathbf{G} \boldsymbol{\alpha}$, with η chosen to

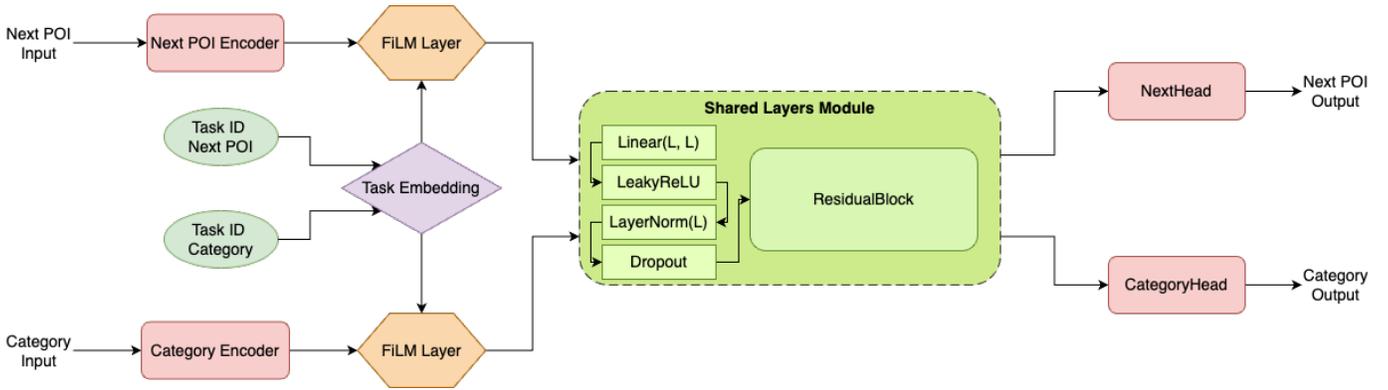


Fig. 1. The proposed Multi-Task Learning (MTL) architecture. Task-specific encoders process inputs for POI Category Classification and Next-POI Prediction. Feature-wise Linear Modulation (FiLM) layers adapt these features using learnable task embeddings. The modulated representations are then processed by a shared path of residual blocks. Finally, dedicated task-specific heads generate the respective outputs.

ensure monotonic loss decrease and convergence to a Pareto-stationary point [21].

Practical Aspects.: Nash-MTL is architecture-agnostic and requires only two matrix-vector products per iteration. Its reliance on gradient signs rather than scales inherently balances heterogeneous tasks without manual re-weighting. For efficiency, task weights can be updated less frequently, significantly reducing runtime while maintaining performance [21].

Parameter Partition.: In our model (Figure 1), we explicitly expose two disjoint parameter sets:

$$\Theta_{\text{shared}} = \{\text{task_emb, film, shared_layers}\},$$

$$\Theta_{\text{specific}} = \{\text{category_encoder, next_encoder, CategoryHead, NextHead}\}.$$

This partition enables independent learning-rate scheduling or regularization if domain shifts between tasks emerge during fine-tuning.

The Nash-MTL optimizer was employed for its ability to aggregate tasks gradients in a balanced and principled manner, formulating the optimization process as a bargaining problem and aiming for joint progress across all tasks. In our evaluation, Nash-MTL was compared with different strategies, including PCGrad [19] and an approach with no optimizer. Among these, it was found that Nash-MTL consistently yielded a better overall performance of the MTL model, with a lower combined multi-task loss, which supported our decision to adopt it as our choice of optimizer.

IV. EXPERIMENTAL SETUP AND RESULTS

A. Dataset and Evaluation Metrics

To evaluate our model, we use a subset of the public *Gowalla* check-in dataset, originally collected by Cho *et al.* [5]. We focus our analysis on the data from the U.S. state of Florida, which is one of the most active regions in the original dataset. This subset comprises a total of N_{users} users, N_{poi} unique Points-of-Interest (POIs), and N_{checkins} check-ins.

Following previous work, each POI is mapped to one of seven high-level categories: *Community*, *Entertainment*, *Food*, *Nightlife*, *Outdoors*, *Shopping*, and *Travel*. These categories will be used for the POI category prediction task.

Evaluation protocol: To ensure a robust evaluation of our models, all experiments were conducted using a 5-fold cross-validation methodology. For every category, we report precision, recall, and F_1 -score, and utilize the macro-average of these metrics to summarize overall performance. Class-wise metrics are crucial because category frequencies are highly skewed in real LBSN data.

B. Discussion and Comparison with Established Models

In this section, we discuss the performance of our proposed MTL model and its single-task (Single) counterpart. For a comprehensive evaluation, we selected two state-of-the-art approaches as baselines. The Human Mobility Representation Model (HMRM), introduced by Chen *et al.* (2020) [7], is designed for POI category classification. HMRM calculates POI embeddings by considering users’ temporal visiting patterns and the co-occurrence of locations within an individual’s historical trajectory. To quantify the relationship between locations and visiting times, it employs Point-wise Mutual Information (PMI). The model then utilizes matrix factorization techniques to generate latent representations (embeddings) from various inputs, and these embeddings are subsequently used to train a Support Vector Machine (SVM) for predicting POI categories.

For the task of next POI category prediction, we compare our models against MHA+PE, a solution proposed by Zeng *et al.* (2019) [8]. This model enhances a recurrent neural network with a Multi-Head Attention (MHA) mechanism, originally developed for machine translation by Vaswani *et al.* (2017) [33], and incorporates Positional Encoding (PE). The MHA mechanism allows the model to extract correlations from different parts of a sequence, effectively capturing the relevance of various elements within a user’s trajectory, while positional encoding helps to propagate the order of records in a sequence.

1) *POI Category Classification*: As shown in Table I, both our MTL and Single models significantly outperform HMRM [7] across all POI categories in terms of F1-score, precision, and recall. For instance, in the ‘Shopping’ category, our

MTL model achieves an F1-score of 62.51 ± 0.94 compared to HMRM’s 46.69 ± 0.81 . Similarly, for ‘Food’, the MTL model scores 57.43 ± 1.46 against HMRM’s 28.44 ± 0.42 . While both MTL and Single models are competitive, the Single model shows marginally better F1-scores in several categories like ‘Community’ (53.11 ± 0.58) and ‘Outdoors’ (47.75 ± 0.89), whereas the MTL model excels in ‘Food’ and ‘Shopping’. Overall, our approaches demonstrate a substantial improvement over the HMRM baseline for this task.

TABLE I

TABLE COMPARING THE RESULTS OF POI CLASSIFICATION FOR FLORIDA

		MTL	Single	HMRM
F1	Community	52.13 ± 0.90	53.11 ± 0.58	20.25±0.52
	Entertainment	41.44 ± 1.29	41.73 ± 1.65	13.07±1.14
	Food	57.43 ± 1.46	56.70 ± 0.84	28.44±0.42
	Nightlife	29.94 ± 2.27	34.22 ± 2.08	25.54±1.73
	Outdoors	47.19 ± 1.01	47.75 ± 0.89	16.11±0.68
	Shopping	62.51 ± 0.94	61.88 ± 1.01	46.69±0.81
	Travel	43.16 ± 1.43	43.20 ± 1.94	15.25±1.19
Precision	Community	51.39 ± 1.77	49.54 ± 1.70	21.13 ± 0.75
	Entertainment	44.08 ± 1.46	44.47 ± 3.66	14.52 ± 1.34
	Food	58.05 ± 1.23	59.04 ± 1.12	48.94 ± 0.88
	Nightlife	36.07 ± 2.05	32.52 ± 1.83	18.34 ± 1.34
	Outdoors	46.39 ± 1.12	47.74 ± 1.09	13.00 ± 0.48
	Shopping	60.24 ± 1.21	61.59 ± 1.44	39.63 ± 0.82
	Travel	46.48 ± 1.12	44.69 ± 4.89	24.58 ± 1.91
Recall	Community	52.93 ± 1.35	57.37 ± 2.46	19.48 ± 0.99
	Entertainment	39.14 ± 1.69	39.88 ± 4.93	11.90 ± 1.08
	Food	56.99 ± 3.71	54.60 ± 2.01	20.05 ± 0.32
	Nightlife	25.80 ± 3.54	36.77 ± 5.86	42.13 ± 3.28
	Outdoors	48.09 ± 2.11	47.79 ± 1.51	21.27 ± 1.78
	Shopping	65.04 ± 2.69	62.28 ± 2.83	56.82 ± 0.98
	Travel	40.39 ± 2.73	43.08 ± 7.15	11.07 ± 0.95

2) *Next POI Category Prediction*: The results for next POI category prediction, detailed in Table II, indicate a competitive performance landscape. The MHA+PE model [8] shows strong F1-scores in several categories, notably ‘Food’ (43.47 ± 0.50) and ‘Shopping’ (43.53 ± 1.66), outperforming our MTL and Single models in these instances. However, our MTL model achieves the best F1-score in ‘Travel’ (64.61 ± 1.11) and ‘Nightlife’ (22.07 ± 0.52), significantly surpassing MHA+PE in these categories. The Single model also shows competitive results, leading in ‘Entertainment’ (26.06 ± 1.01) and ‘Outdoors’ (22.45 ± 0.81). This suggests that while MHA+PE is highly effective for certain POI categories, our MTL and Single models offer superior or comparable performance in others, particularly where different sequential patterns might be leveraged. Also, it is important to notice that since we have an unbalanced result for the MTL and single, this could lead to the worse of other results.

Overall, while our proposed models show strong performance in POI category classification against HMRM [7], the next POI category prediction task presents a more competitive landscape when compared against a sophisticated sequential model like MHA+PE [8]. Comparing the MTL and Single models directly reveals that the multi-task learning approach did not yield the anticipated substantial improvements over the single-task baseline across both tasks. Crucially, many of these observed differences in F1-scores are minor and often fall within the reported standard deviations, suggesting

TABLE II
TABLE COMPARING THE RESULTS OF NEXT POI PREDICTION FOR FLORIDA

		MTL	Single	MHA+PE
F1	Community	34.79 ± 1.05	34.90 ± 0.61	35.83±1.70
	Entertainment	24.21 ± 1.75	26.06 ± 1.01	25.48±0.86
	Food	28.06 ± 3.55	26.35 ± 2.97	43.47±0.50
	Nightlife	22.07 ± 0.52	21.79 ± 0.30	1.55±1.50
	Outdoors	21.61 ± 0.50	22.45 ± 0.81	12.56±1.94
	Shopping	42.58 ± 2.06	42.18 ± 1.13	43.53±1.66
	Travel	64.61 ± 1.11	64.32 ± 0.88	26.91±0.97
Precision	Community	30.22 ± 1.36	30.04 ± 1.08	42.59±1.48
	Entertainment	32.64 ± 2.55	28.63 ± 2.85	38.72±1.56
	Food	39.05 ± 0.70	39.87 ± 1.48	35.91±0.81
	Nightlife	17.71 ± 2.40	15.27 ± 0.67	39.67±23.94
	Outdoors	19.63 ± 2.93	19.50 ± 2.08	46.83±1.82
	Shopping	42.57 ± 0.78	43.47 ± 1.08	39.04±1.06
	Travel	60.64 ± 1.87	63.21 ± 0.67	36.64±1.74
Recall	Community	41.54 ± 5.27	41.92 ± 3.49	31.16±3.20
	Entertainment	19.52 ± 3.18	24.16 ± 2.16	19.05±1.25
	Food	22.19 ± 4.32	19.94 ± 3.75	55.12±1.46
	Nightlife	32.13 ± 9.47	38.35 ± 3.11	0.80±0.79
	Outdoors	25.06 ± 4.26	27.07 ± 3.40	7.30±1.32
	Shopping	42.87 ± 4.56	41.05 ± 2.33	49.37±3.74
	Travel	69.36 ± 4.15	65.52 ± 2.25	21.30±1.03

that, statistically, the performance of the MTL and Single models is largely comparable, without a clear, consistent, and significant advantage for the multi-task learning setup in these experiments.

C. Convergence Comparison

To further evaluate the practical implications of the MTL approach compared to single-task models, we conducted a convergence experiment. We measured the wall time, number of epochs, and Mega Floating Point Operations (MFLOPs) required for the MTL model and the individual single-task models (SingleClass and SinglePred) to reach predefined target average F1-scores. The target F1-score for the *Category* prediction task was set to 47, and for the *Next POI* category classification task, it was set to 32.2. These specific F1 values were chosen because they are close to the best-achieved results for each respective task, thereby representing a comparable and significant level of predictive performance for the models to attain. To ensure robust measurements for this experiment, all models were evaluated through a 5-fold cross-validation process. The specific metrics are detailed in Table III.

TABLE III
CONVERGENCE METRICS TO REACH TARGET F1-SCORE

Model	Time(s)	Epochs	MFLOPs
Category	16.26	3.8	2.315
Next	18.71	3.2	0.012
MTL	80.88	3.2	0.234

The results from this experiment were contrary to the potential expectation that an MTL framework might offer training efficiencies. Our findings indicated that the MTL model took substantially longer to converge to the target F1-scores. Specifically, the MTL approach required almost four times more wall time compared to the cumulative time of the

individual single-task models. Furthermore, the computational cost, when considering MFLOPs, was roughly double for the MTL setup. This suggests that while attempting to learn multiple tasks simultaneously, the MTL model, in this configuration, incurred a significantly higher overhead in both time and computational resources. Given these observations, for deployment scenarios where convergence speed and computational efficiency are critical factors, employing separate, optimized single-task models would likely be a more convenient and resource-efficient strategy.²

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a Multi-Task Learning (MTL) framework employing hard parameter-sharing, task-specific encoders, and FiLM layers to jointly address POI Category Classification and Next-POI Prediction using DGI-based POI embeddings.

Essentially, the MTL approach did not consistently demonstrate superior performance over its single-task counterparts and exhibited higher computational demands in terms of convergence time and MFLOPs. These findings underscore that the anticipated benefits of MTL are not universally guaranteed and are highly dependent on task compatibility and architectural design.

The lack of significant improvement from the MTL model prompts an analysis of the potential underlying causes, especially since optimizers like Nash-MTL mitigated overt gradient conflicts. We hypothesize three primary factors contributed to this outcome:

- **Subtle Negative Transfer due to Task Dissimilarity:** Although related, the two tasks possess fundamentally different natures. POI Category Classification is a static task that relies on the intrinsic, context-rich features of a single POI embedding. In contrast, Next-POI Prediction is a dynamic, sequential task that depends on capturing temporal patterns and transitions within a user’s trajectory. The shared encoder may have been forced to learn a “compromise” representation that was not specialized enough for either task, thus failing to outperform the focused single-task models.
- **Task Difficulty and Representation Mismatch:** The performance gap between the two tasks (with category classification achieving higher F1-scores) suggests an imbalance in difficulty. The representation learned by the shared layers might have become biased towards the features required for the simpler, static classification task, inadvertently hindering its effectiveness for the more complex sequential prediction task. The shared representation may not have been rich enough to simultaneously encode both semantic properties and sequential dynamics effectively.

²Experiments conducted on Apple M2 Pro (10-core CPU, 16-core GPU) with 32GB unified memory running macOS 15.5. Key software versions: Python 3.9.6, PyTorch 2.6.0.dev20241014+cpu, NumPy 1.26.4, scikit-learn 1.5.2, CVXPY 1.5.2. Training utilized PyTorch MPS backend for Apple Silicon acceleration.

- **Architectural Restrictiveness:** Our choice of a hard parameter-sharing architecture, while efficient and strongly regularizing, may have been too restrictive for the tasks. The distinct nature of the tasks might require more flexible computational pathways. A single shared block may be insufficient to learn a representation that benefits both, suggesting that soft-sharing or expert-based models could be more appropriate.

These insights contribute to the broader understanding of MTL’s practical challenges. Future research will proceed along several directions motivated by these hypotheses. We plan to explore alternative parameter-sharing mechanisms, such as **soft sharing (e.g., Cross-Stitch Networks) or Mixture-of-Experts (MoE) models**, to test the hypothesis that the hard-sharing architecture was overly restrictive. Furthermore, investigating **advanced multi-task optimizers and loss-balancing schemes** beyond gradient conflict mitigation could address the task difficulty imbalance. Finally, a deeper analysis of **task relatedness**, perhaps using techniques to measure feature representation overlap, could help quantify the degree of negative transfer and guide future architectural choices.

ACKNOWLEDGMENT

The authors would like to acknowledge the support of the Brazilian Ministry of Science, Technology and Innovations (MCTI), the Manna Team, Fundação Araucária, Softex, CNPq (Grant Number 421548/2022-3), and the Research Support Foundation of the State of Minas Gerais (FAPEMIG), and CAPES.

REFERENCES

- [1] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [2] I. Kokkinos, “Ubernet: Training a ‘universal’ convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory,” *arXiv preprint arXiv:1609.02132*, 2016.
- [3] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, “Finetuned language models are zero-shot learners,” in *Proceedings of the 10th International Conference on Learning Representations (ICLR)*, 2022. [Online]. Available: <https://openreview.net/forum?id=gEzrGCozdqR>
- [4] L. Zhang, Z. Sun, J. Zhang, Y. Lei, C. Li, Z. Wu, H. Kloeden, and F. Klanner, “An interactive multi-task learning framework for next poi recommendation with uncertain check-ins,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, 2020, pp. 3551–3557.
- [5] E. Cho, S. A. Myers, and J. Leskovec, “Friendship and mobility: User movement in location-based social networks,” in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 2011, pp. 1082–1090. [Online]. Available: <https://cs.stanford.edu/people/jure/pubs/gowalla-kdd11.pdf>
- [6] J. Leskovec and A. Krevl, “Snap datasets: Stanford large network dataset collection,” <https://snap.stanford.edu/data/loc-gowalla.html>, 2014, accessed: 2024-05-21.
- [7] M. Chen, Y. Zhao, Y. Liu, X. Yu, and K. Zheng, “Modeling spatial trajectories with attribute representation learning,” in *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 10. IEEE, 2020, pp. 4829–4841.
- [8] J. Zeng, X. He, H. Tang, and J. Wen, “A next location predicting approach based on a recurrent neural network and self-attention,” in *International Conference on Collaborative Computing: Networking, Applications and Worksharing*. Springer, 2019, pp. 309–322.

- [9] R. Xu, M. Chen, Y. Gong, Y. Liu, X. Yu, and L. Nie, "Tme: Tree-guided multi-task embedding learning towards semantic venue annotation," *ACM Trans. Inf. Syst.*, vol. 41, no. 4, p. 112, 2023.
- [10] N. Lim, B. Hooi, S.-K. Ng, Y. L. Goh, R. Weng, and R. Tan, "Hierarchical multi-task graph recurrent network for next poi recommendation," in *SIGIR '22: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2022, pp. 1133–1143. [Online]. Available: <https://doi.org/10.1145/3477495.3531989>
- [11] J. Yu, Y. Dai, X. Liu, J. Huang, Y. Shen, K. Zhang, R. Zhou, E. Adhikarla, W. Ye, Y. Liu, Z. Kong, K. Zhang, Y. Yin, V. Nambodiri, B. D. Davison, J. H. Moore, and Y. Chen, "Unleashing the power of multi-task learning: A comprehensive survey spanning traditional, deep, and pretrained foundation model eras," *arXiv preprint arXiv:2404.18961*, 2024. [Online]. Available: <https://arxiv.org/abs/2404.18961>
- [12] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 12, pp. 5586–5609, 2022.
- [13] I. Misra, A. Shrivastava, A. Gupta, and M. H'ebert, "Cross-stitch networks for multi-task learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3994–4003.
- [14] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Sluice networks: Learning what to share between loosely related tasks," *arXiv preprint arXiv:1705.08142*, 2017.
- [15] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 1930–1939.
- [16] Z. Yu, Y. Zhang, and W. Yu, "Multi-gate mixture-of-experts for multi-task learning in recommendations," *arXiv preprint arXiv:1904.01038*, 2019.
- [17] Z. Chen, V. Badrinarayanan, C. Lee, and A. Rabinovich, "Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks," in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018, pp. 794–803.
- [18] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018, pp. 527–538.
- [19] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn, "Gradient surgery for multi-task learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6836–6848.
- [20] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, introduces Dynamic Weight Averaging (DWA).
- [21] A. Navon, A. Shamsian, I. Achituve, H. Maron, K. Kawaguchi, G. Chechik, and E. Fetaya, "Multi-task learning as a bargaining game," *arXiv preprint arXiv:2202.01017*, 2022.
- [22] T. Standley, A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, "Which tasks should be learned together in multi-task learning?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3713–3724.
- [23] D. Liao, W. Liu, Y. Zhong, J. Li, and G. Wang, "Predicting activity and location with multi-task context aware recurrent neural network," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 3435–3441. [Online]. Available: <https://doi.org/10.24963/ijcai.2018/477>
- [24] L. Zhang, Z. Sun, J. Zhang, Y. Lei, C. Li, Z. Wu, H. Kloeden, and F. Klanner, "An interactive multi-task learning framework for next poi recommendation with uncertain check-ins," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 3551–3557, main track. [Online]. Available: <https://doi.org/10.24963/ijcai.2020/491>
- [25] S. Halder, K. H. Lim, J. Chan, and X. Zhang, "Transformer-based multi-task learning for queuing time aware next poi recommendation," in *Advances in Knowledge Discovery and Data Mining*. Cham: Springer International Publishing, 2021, pp. 510–523.
- [26] B. Xia, Y. Bai, J. Yin, Q. Li, and L. Xu, "Mtp: A multi-task learning based poi recommendation considering temporal check-ins and geographical locations," *Applied Sciences*, vol. 10, no. 19, 2020. [Online]. Available: <https://www.mdpi.com/2076-3417/10/19/6664>
- [27] J. Du, Y. Zhang, P. Wang, J. Leopold, and Y. Fu, "Beyond geo-first law: Learning spatial representations via integrated autocorrelations and complementarity," in *2019 IEEE international conference on data mining (ICDM)*. IEEE, 2019, pp. 160–169.
- [28] W. Huang, L. Cui, M. Chen, D. Zhang, and Y. Yao, "Estimating urban functional distributions with semantics preserved poi embedding," *International Journal of Geographical Information Science*, vol. 36, no. 10, pp. 1905–1930, 2022.
- [29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.
- [30] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," *arXiv preprint arXiv:1809.10341*, 2018.
- [31] E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. C. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artificial Intelligence*, 2018, pp. 3942–3951.
- [32] J. Baxter, "A model of inductive bias learning," *Journal of Artificial Intelligence Research*, vol. 12, pp. 149–198, 2000.
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.