Learning District Visitation Sequences for Hierarchical Last-Mile Delivery

Farzam Salimi[1][0009−0007−4928−5288], Ana Rita Simões do Paço e Vaz[1,2][0009−0006−0769−7308], and António Galrão Ramos[1,2][0000−0002−0885−7792]

[1] INESC TEC - Institute for Systems and Computer Engineering, Technology and Science, Porto, Portugal
farzam.salimi@inesctec.pt
ana.r.vaz@inesctec.pt
[2] ISEP, Instituto Superior de Engenharia do Porto, Portugal
agr@isep.ipp.pt

Abstract. District-based planning is commonly used to make city-scale last-mile delivery more tractable, but the intermediate task of predicting district visitation order is still difficult to model in a way that is both data-driven and operationally simple. This paper studies hierarchical district sequencing as a supervised recurrent prediction problem rather than as an end-to-end routing optimisation problem. From 1,268,337 historical delivery records in Porto, delivery stops are assigned to a three-level polygon hierarchy and converted into route-level and parent-conditioned district sequences. We train gated recurrent unit (GRU) predictors for the three hierarchy levels, using parent-district information at the child levels, and evaluate them with route-disjoint train, validation, and test partitions. The models are compared with three lightweight operational baselines: a global frequency rule, a first-order transition rule, and a geometry-based permanent visitation order. The GRU provides the largest benefit at the macro-district level, improving Level 1 accuracy from 0.291 to 0.520 and top-5 accuracy from 0.712 to 0.877. At finer resolutions, the recurrent model remains competitive, but the smaller feasible child sets make transition-frequency baselines harder to surpass. These results provide a compact GRU-based recurrent baseline for hierarchical district-sequence prediction and clarify where recurrent memory appears most useful in district-based delivery planning.

Keywords: last-mile delivery · recurrent neural networks · district sequencing · hierarchical planning · vehicle routing

1 Introduction

Urban delivery operators repeatedly serve the same metropolitan territory, but the order in which service regions are visited can vary from route to route. These realised district orders are not arbitrary: they reflect geography, depot departure patterns, driver routines, demand distribution, and operational constraints that

are only partially captured by static distance-based rules. This makes historical route traces a useful source of information for learning district visitation behaviour.

This paper focuses on one specific decision inside hierarchical last-mile delivery planning: predicting the next district in a visitation sequence. The objective is not to solve the complete vehicle routing problem, nor to replace detailed stop-level optimisation. Instead, we isolate the sequencing layer and ask whether a compact recurrent model can reproduce district-order patterns more accurately than simple operational rules. This framing is useful because district order can be predicted before detailed routing and can later be used as a warm start, ordering prior, or constraint for downstream optimisation.

The study is carried out on a three-level polygon hierarchy for the city of Porto. At the coarsest level, the model predicts transitions between macro-districts along full routes. At the two finer levels, predictions are made within the currently active parent district, so that child-level decisions remain consistent with the hierarchy. This produces three related but different prediction tasks: route-level Level 1 sequencing, Level 2 sequencing conditioned on Level 1 parents, and Level 3 sequencing conditioned on Level 2 parents.

The central research question is therefore: at which spatial resolution does recurrent sequence memory add value beyond frequency-based and geometry-based baselines? This question is important because the benefit of machine learning is not expected to be uniform across the hierarchy. Coarse districts may require longer contextual memory, while fine-grained child districts may be largely governed by local transition regularities.

Contributions. This paper makes three contributions. First, it provides a GRU-based recurrent study of hierarchical district-sequence prediction in last-mile delivery, using the same compact architecture across three nested spatial resolutions. Second, it introduces a simple parent-conditioned recurrent formulation for Level 2 and Level 3 prediction, allowing child-district sequences to be learned within their active parent districts without requiring a joint multi-level decoder. Third, it provides an empirical scale-level analysis showing that recurrent modelling is most beneficial for coarse district sequencing, while fine-grained sequencing is often already well approximated by first-order transition statistics.

Paper organisation. The remainder of this paper is organised as follows. Section 2 reviews district-based decomposition, cluster ordering, and recurrent sequence models for operational prediction. Section 3 describes the data preparation pipeline and GRU-based prediction models. Section 4 reports the experimental protocol and quantitative results. Section 5 discusses the scale-dependent behaviour of recurrent modelling, and Section 6 concludes the paper.

2 Related Work

2.1 District-based decomposition in last-mile delivery

Large urban delivery instances are often too large to treat as a single monolithic routing problem. Decomposition approaches therefore partition the service territory into smaller geographic units before detailed routes are constructed. In last-mile delivery, this is particularly useful because operational decisions are shaped not only by distance, but also by service density, time pressure, driver familiarity, and repeated daily execution patterns [1]. The integrated Porto framework in [2] follows this logic by separating the planning process into districting, district sequencing, and detailed routing.

The present paper uses this hierarchical view but studies only the sequencing layer. This distinction is important: we do not claim to produce complete vehicle routes. Instead, we learn district-order information that can be used before, or together with, downstream route construction.

2.2 Cluster ordering as an intermediate routing decision

When customers are grouped into districts or clusters, the order of these groups becomes a separate planning decision. This connects the present problem to clustered routing models, including the Clustered Traveling Salesman Problem and clustered variants of the Vehicle Routing Problem [3, 4]. In such models, cluster order influences the structure of the detailed route because customers in the same cluster are usually served contiguously or under related constraints.

Classical approaches often define this order using geometric summaries, aggregated travel costs, or decomposition procedures [5]. These methods are attractive because they are interpretable and computationally efficient. However, they may ignore recurring patterns in executed routes, such as driver-specific habits, depot departure preferences, or systematic district-order motifs. This motivates learning district sequences directly from historical operations.

2.3 Recurrent sequence models for operational prediction

District visitation can be represented as a symbolic sequence: each token is a district identifier, and the learning task is to predict the next token given the previous visits. Recurrent neural networks are a natural baseline for this setting because they process the prefix sequentially and maintain a hidden state summarising previous decisions. Among recurrent architectures, gated recurrent units (GRUs) provide a compact alternative to more complex sequence models, with fewer parameters than many attention-based architectures and a direct autoregressive interpretation.

In routing and logistics, learning-based methods have been used to approximate decisions, capture hidden preferences, and support large-scale optimisation workflows [6, 7]. The contribution of the present paper is narrower: it evaluates whether GRU models are useful for district-level sequence prediction inside a

hierarchical delivery pipeline. This makes the GRU not an end-to-end optimiser, but a data-driven sequencing component.

2.4 Positioning of this paper

The paper is positioned as a recurrent benchmark for hierarchical district sequencing. It differs from classical clustered-routing algorithms because it predicts historically observed district orders rather than solving a full clustered routing instance. It also differs from higher-capacity neural routing approaches because it deliberately uses a compact GRU architecture and evaluates where such a model is sufficient. The resulting analysis is intended to identify the hierarchy levels at which recurrent memory provides practical value over simple frequency, transition, and static-order baselines.

3 Methodology

This work follows the hierarchical structure of the reference framework (districting–sequencing–routing) and focuses on the *sequencing* task. The service region has already been partitioned into a three-level hierarchy of polygonal districts (Level 1, Level 2, and Level 3). Using historical delivery traces, we learn sequence models that predict plausible district visitation orders at each level, with the goal of providing a data-driven sequencing component that can be integrated into the downstream routing stage.

3.1 Data preparation and hierarchical assignment

Each delivery record includes geographic coordinates (longitude, latitude) and operational descriptors that identify the route (in our data, a service date and a circuit identifier), as well as a visit time used to order stops. All polygon layers and delivery points are transformed to a common coordinate reference system prior to matching. Each stop is assigned to polygon identifiers using a spatial join with the predicate *within*. Data pre-processing stage is similar to the reference [2]. After that, polygon identifiers are normalised to string tokens (e.g., converting float-encoded IDs to integer-like strings) to ensure consistent representation during vocabulary construction.

3.2 Route construction and sequence extraction

Stops are grouped into routes using the operational route key (date and circuit), and are ordered using the recorded visit time. From each route, we extract district visitation sequences at multiple resolutions. The Level 1 sequence is obtained by recording the ordered list of visited Level 1 districts. To build conditional sequences at finer levels, we segment each route according to the hierarchy: Level 2 sequences are constructed within contiguous Level 1 segments, and Level 3 sequences are constructed within contiguous Level 2 segments. To avoid

over-representing repeated stops within the same district, we apply consecutive-collapsing (enabled in our experiments), which removes only *consecutive* duplicates in the extracted sequences. Districts that never appear in the historical data are naturally excluded from the learning problem.

3.3 Learning setup and data splitting

Sequencing is formulated as an autoregressive next-token prediction task. For each hierarchy level, we build a vocabulary consisting of all observed district tokens and four special tokens for padding, sequence start, sequence end, and unknown identifiers. Training examples are constructed by shifting each sequence by one position, so that the input contains the start token followed by the sequence, and the target contains the sequence followed by the end token. To ensure a fair evaluation and prevent leakage across partitions, we split the dataset by *route*: all sequences (and all segments at all levels) derived from the same route are assigned to the same split. We use an 80%/10%/10% split for training/validation/test.

3.4 Recurrent prediction model

The GRU architecture is used deliberately as a compact recurrent baseline rather than as a claim of architectural novelty. This choice allows us to test whether sequential memory improves district prediction beyond frequency-based and transition-based rules, while keeping the model simple enough to compare across hierarchy levels. More expressive architectures, such as attention-based decoders, may improve performance at coarse levels, but they also introduce additional modelling choices. We therefore treat the GRU as a controlled recurrent reference model for evaluating the usefulness of sequence memory in this setting.

For each hierarchy level, we train an independent GRU predictor. Let x_t denote the district token observed at position t in a sequence and let $E(\cdot)$ be the corresponding token-embedding matrix. At Level 1, the input at time t is the district embedding

$$z_t = E(x_t).$$

The hidden state is updated recurrently as

$$h_t = \mathrm{GRU}(z_t, h_{t-1}),$$

and the next-token distribution is obtained through a linear output layer followed by a softmax:

$$p(x_{t+1} \mid x_{\leq t}) = \mathrm{softmax}(W_o h_t + b_o).$$

For Level 2 and Level 3, the prediction is conditioned on the active parent district. If p denotes the parent token and $E_p(\cdot)$ its embedding matrix, the recurrent input becomes

$$z_t = [E(x_t) \,||\, E_p(p)],$$

where || denotes concatenation. The same parent embedding is provided at every time step of the segment, so the model learns child-level transitions under a fixed parent context. This design keeps the model simple while preventing child-level prediction from being treated as a flat vocabulary problem.

The three models are trained separately rather than jointly. This choice supports a diagnostic comparison across hierarchy levels: each model is evaluated only on the prediction task associated with its own spatial resolution.

3.5 Training procedure and evaluation

Models are trained with teacher forcing using a padding-masked next-token classification loss. Optimisation uses Adam, with gradient clipping to stabilise training and early stopping based on validation loss. We report token-level accuracy and top-5 accuracy computed over non-padding positions. All reported metrics are computed in a teacher-forced evaluation setting on the held-out test routes.

3.6 Implementation details

All experiments are implemented in PyTorch with a fixed random seed of 42. We use an embedding dimension of 32 and a GRU hidden size of 64 across all hierarchy levels, with a single recurrent layer and no dropout. Models are trained for up to 200 epochs with batch size 64 and an initial learning rate of 10^{-2}. We apply gradient clipping with maximum norm 1.0. Early stopping uses a patience of 8 epochs with a minimum validation improvement threshold of 10^{-4}. When enabled suggesting no further improvements, a Reduce-on-Plateau scheduler reduces the learning rate by a factor of 0.5 after 2 validation plateaus. Training runs on a GPU.

4 Experimental Results

This section reports the predictive performance of the proposed GRU-based sequencing models. All results are computed on the held-out test split using the route-level partitioning described in Section 3. Metrics are evaluated under teacher forcing and exclude padding positions.

Dataset. The experiments use the same operational dataset described in [2], collected from last-mile delivery activities in the city of Porto. The raw data comprise 1,268,337 delivery records, and 9,559 routes where each record corresponds to a single destination (stop) and includes a unique identifier, delivery timestamp, package size attributes, route identifiers (e.g., circuit/route ID), service date, and performance-related variables. Following the preprocessing pipeline in Section 3, each stop is mapped to a three-level polygon hierarchy and then aggregated into route-level district sequences used for training and evaluation.

4.1 Evaluation protocol

We evaluate next-district prediction at three spatial resolutions (Level 1, Level 2, Level 3) using token-level accuracy and top-5 accuracy. Accuracy measures whether the most probable predicted token matches the ground truth next token, while top-5 accuracy measures whether the ground truth token is among the five most probable predictions. To avoid data leakage, all sequences derived from the same operational route are assigned to the same data split, as described in Section 3.

4.2 Baselines

To contextualise the learned models, we compare against three lightweight baselines computed from the training split.

Most-frequent predicts the same candidates at every position: the five most common next-district tokens observed in the training targets for the corresponding hierarchy level.

Transition-frequency is a first-order transition baseline built from empirical next-token counts. At Level 1 it uses the previously visited district to select the most frequent successors, while at Levels 2 and 3 it is additionally conditioned on the parent district (i.e., transitions are learned separately within each parent). For each state, it returns the top-5 most frequent next districts; if a state was not observed in training, it falls back to the global most-frequent list.

Permanent sequence is a static-policy baseline that constructs a fixed visitation order using centroid-based Euclidean distances in the projected coordinate system and a greedy nearest-neighbour tour. For Level 1, a single order over Level 1 districts is computed starting from the depot location; for Levels 2 and 3, a separate static order is computed per parent district (Level 1→Level 2 and Level 2→Level 3). During evaluation, the predicted next district is taken as the successor in the corresponding static order (cyclic), and the top-5 list is defined by the next five successors.

These baselines reflect common choices in scalable hierarchical planning pipelines, where sequencing is often treated as a static rule or approximated through aggregated transition statistics.

4.3 Main quantitative results

Table 1 summarises test performance across hierarchy levels. The main empirical finding is not that neural models dominate all hierarchy levels, but that the value of recurrent memory is scale-dependent.

At Level 1, the GRU substantially improves over the first-order transition baseline, increasing accuracy from 0.291 to 0.520 and top-5 accuracy from 0.712 to 0.877. This indicates that macro-district order depends on more than the immediately preceding district, and that the recurrent hidden state captures useful route-level structure. However, this result should be interpreted as predictive evidence only; it does not by itself prove that the resulting sequences reduce downstream routing cost or runtime.

Table 1: Main test results. Metrics are token-level and exclude padding. Higher is better.

Model	**Level**	**Acc.**	**Top-5**
Most-frequent	L1	0.0960	0.3040
Transition-frequency	L1	0.2912	0.7122
Permanent sequence	L1	0.1247	0.1878
GRU (ours)	L1	**0.5201**	**0.8770**
Most-frequent	L2	0.3240	0.3540
Transition-frequency	L2	0.4400	0.8470
Permanent sequence	L2	0.1050	0.3250
GRU + parent (ours)	L2	**0.4690**	**0.8590**
Most-frequent	L3	0.4420	0.4520
Transition-frequency	L3	0.6260	0.9510
Permanent sequence	L3	0.1050	0.5140
GRU + parent (ours)	L3	**0.6330**	**0.9630**

At Level 2, the improvement is smaller but still positive: accuracy increases from 0.440 to 0.469, while top-5 accuracy increases from 0.847 to 0.859. This suggests that parent conditioning and local transition structure already explain a large part of the prediction problem, leaving less room for recurrent modelling to improve over empirical transition counts.

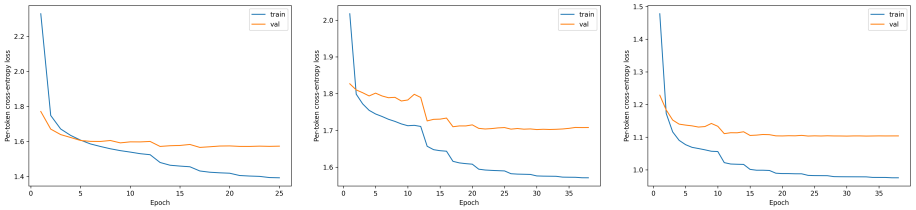
At Level 3, the transition-frequency baseline is very difficult to improve upon. The GRU obtains a slightly higher accuracy, 0.633 compared with 0.626, and a higher top-5 accuracy, 0.963 compared with 0.951. However, the gain is modest because the finest-level task is close to a local child-selection problem within a restricted parent district.

This pattern is useful from an operational perspective. It suggests that recurrent models should be prioritised where the sequencing decision has broader spatial scope, while simpler empirical transition rules may be sufficient for highly local child-level decisions.

4.4 Training dynamics

Figure 1 shows the training and validation cross-entropy curves for the three hierarchy levels. In all cases, training loss decreases steadily, while validation loss improves rapidly in the first epochs and then plateaus, indicating convergence. The growing gap between training and validation loss at later epochs suggests mild overfitting, which motivates the use of early stopping and, potentially, additional regularisation (e.g., dropout) in future experiments.

Summary. Overall, the results confirm that district-level sequencing is learnable from historical traces, with the clearest gains at the coarsest level. At finer granularities, strong transition-based baselines reduce the performance gap, indicating that additional context or model capacity may be necessary to consistently outperform first-order policies.



(a) Level 1 (b) Level 2 (c) Level 3

Fig. 1: Training and validation cross-entropy loss for GRU sequencing models at each hierarchy level.

5 Discussion

The experiments show that the usefulness of recurrent modelling depends strongly on the spatial resolution of the sequencing task. At Level 1, the GRU clearly improves over the transition-frequency baseline, suggesting that macro-district prediction benefits from information stored in the recurrent hidden state. This indicates that the immediately previous district is not sufficient to explain many coarse-level transitions.

At Level 2, the improvement becomes smaller. This suggests that once the active Level 1 parent is known, many child-district transitions are already captured by empirical transition counts. The GRU still improves over the transition-frequency baseline, but the margin is modest, indicating that the recurrent model adds useful but limited information at this resolution.

At Level 3, the task becomes even more local. The transition-frequency baseline is already very strong, and the GRU provides only a small improvement. This result is important because it shows that more complex sequence models are not automatically preferable at every level of the hierarchy. In highly restricted child-level prediction tasks, simple transition statistics may be sufficient for many operational cases.

The main implication is therefore diagnostic rather than universal: recurrent sequence memory is most useful when the district-order decision has wider spatial scope. For very local child-level prediction, the additional complexity of a GRU may only be justified if extra covariates are introduced, such as day-of-week, time-of-day, demand intensity, vehicle information, or estimated travel conditions.

Several limitations should be noted. The evaluation is restricted to teacher-forced next-token prediction, which means that improved predictive accuracy is not automatically equivalent to improved routing performance. This distinction is important because routing objectives may be misaligned with next-token prediction objectives: a model can reproduce historical district transitions more accurately without necessarily producing lower-cost downstream routes. We therefore interpret the results as evidence that recurrent models can learn district-

order regularities, not as direct evidence of routing-cost improvement. Full autoregressive rollout, centroid-based macro-order distance, and downstream routing evaluation require an additional decoding and routing layer and are outside the scope of this GRU-focused study. In addition, the current models use only district identifiers and parent context, so future work should test whether operational covariates improve recurrent prediction, especially at finer hierarchy levels.

6 Conclusion

This paper presented a GRU-based recurrent benchmark for hierarchical district-sequence prediction in last-mile delivery. Using historical delivery data from Porto, we converted executed routes into district sequences over a three-level polygon hierarchy and trained separate recurrent models for each spatial resolution. Parent-district conditioning was used at the two finer levels to keep child-level predictions consistent with the hierarchy.

The results show that recurrent modelling is most valuable at the coarsest level, where the GRU clearly improves over frequency-based, transition-based, and static-order baselines. At finer levels, the performance gap narrows because parent conditioning restricts the feasible child districts and makes local transition statistics highly informative. This indicates that the usefulness of recurrent sequence memory is not uniform across the hierarchy. At the same time, these findings should be viewed as prediction-level results rather than as direct evidence of improved routing solutions.

Future work should evaluate the predicted sequences under free-running decoding, measure their effect on downstream routing cost and runtime, and incorporate additional operational covariates such as day-of-week, time-of-day, demand intensity, and estimated travel conditions. Another direction is to compare recurrent models with alternative sequence architectures under the same route-disjoint evaluation protocol.

Acknowledgments. This article is co-funded by the European Regional Development Fund (ERDF) through the Innovation and Digital Transition Programme (COMPETE 2030) under Portugal 2030 and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P. (Portuguese Foundation for Science and Technology) within project AIOTacitR, with reference 15051 (COMPETE2030-FEDER-00870300) (https://www.compete2030.gov.pt/operacoes/listagem/COMPETE2030-FEDER-00870300/).

Disclosure of Interests. Competing Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Boysen, N., Fedtke, S., Schwerdfeger, S.: Last-mile delivery concepts: a survey from an operational research perspective. OR Spectrum **43**, 1–58 (2021). https://doi.org/10.1007/s00291-020-00607-8

2. Silva, J.R., Ramos, A.G., Salimi, F.: An Integrated Framework to Address Last-Mile Delivery Problem in Large-Scale Cities by Combination of Machine Learning and Optimisation. SN Computer Science **6**, 650 (2025). https://doi.org/10.1007/s42979-025-04180-1
3. Lu, Y., Hao, J.-K., Wu, Q.: Solving the clustered traveling salesman problem via traveling salesman problem methods. PeerJ Computer Science **8**, e972 (2022). https://doi.org/10.7717/peerj-cs.972
4. Park, J., Choi, I., Kim, H.J.: A Multi-View Attention-Based Encoder-Decoder Framework for Clustered Traveling Salesman Problem. IEEE Robotics and Automation Letters **11**(1), 137–144 (2026). https://doi.org/10.1109/LRA.2025.3632724
5. Gendreau, M., Tarantilis, C.D.: Solving large-scale vehicle routing problems with time windows: The state-of-the-art. Cirrelt Montreal, Montreal (2010)
6. Mandi, J., Canoy, R., Bucarey Lopez, V., Guns, T.: Data Driven VRP: A Neural Network Model to Learn Hidden Preferences for VRP. In: Michel, L.D. (ed.) 27th International Conference on Principles and Practice of Constraint Programming (CP 2021), LIPIcs, vol. 210, pp. 42:1–42:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany (2021). https://doi.org/10.4230/LIPIcs.CP.2021.42
7. Özarık, S.S., Da Costa, P., Florio, A.M.: Machine Learning for Data-Driven Last-Mile Delivery Optimization. SSRN Electronic Journal (2022). https://doi.org/10.2139/ssrn.4012376
8. Laporte, G.: The vehicle routing problem: An overview of exact and approximate algorithms. European Journal of Operational Research **59**(3), 345–358 (1992). https://doi.org/10.1016/0377-2217(92)90192-C
9. Braekers, K., Ramaekers, K., Van Nieuwenhuyse, I.: The vehicle routing problem: State of the art classification and review. Computers & Industrial Engineering **99**, 300–313 (2016). https://doi.org/10.1016/j.cie.2015.12.007
10. Zhang, Z., Qi, G., Guan, W.: Coordinated multi-agent hierarchical deep reinforcement learning to solve multi-trip vehicle routing problems with soft time windows. IET Intelligent Transport Systems **17**(10), 2034–2051 (2023). https://doi.org/10.1049/itr2.12394