TRANSFERRING ACTIVITY PROTOTYPES FOR ZERO-SHOT CROSS-CITY MOBILITY GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory generation produces realistic human mobility data at low cost while preserving privacy. It has become increasingly important for urban planning and management in smart cities. However, training a generative model requires abundant local real trajectory data, which many cities cannot provide. An effective alternative is to transfer behavioral knowledge from data-rich cities. Nevertheless, differences in location semantics and spatial structure hinder the direct reuse of source individuals' activity patterns. To address this challenge, we propose SHAPE, a cross-city mobility generation framework that extracts shared activity patterns as core representations for cross-city transfer. The key idea of SHAPE is to encode individual activity patterns into interpretable, city-independent activity prototypes with residual finite scalar quantization, and transfer them to the target city with its static map and the spatial distribution of its residents, thus generating high-quality individual mobilities with adaptation to the target urban environment. In zero-shot cross-city mobility generation, we compare SHAPE with three families of state-of-the-art baselines on four source-to-target cases. SHAPE ranks first on four of the five metrics and at least second on the remaining one. In an oracle test from New York City to Paris, SHAPE uses no individual trajectory of Paris and reaches 90.1% of the OD CPC of an oracle given those trajectories. The extraction of shared activity patterns, rather than reliance on abundant target-city ground truth, is the primary driver of performance improvements. By introducing a city-independent codebook of activity prototypes, our framework effectively balances the trade-off between model capacity and cross-city transfer robustness. The code is available at https://anonymous.4open.science/r/SHAPE-anonymous-0448/.

1 INTRODUCTION

Human mobility trajectories support smart city planning and public service evaluation by revealing population flows and individual activity patterns within urban environments Pappalardo et al. (2015); Song et al. (2010a; 2014); Chen et al. (2016). However, several factors limit many cities with insufficient fine-grained individual mobility data, including privacy constraints, high data acquisition costs, and uneven geographic coverage. These limitations motivate an important research question: how can mobility data accumulated in other cities support realistic individual-level mobility simulation in cities with limited observations?

Although cities differ in their geographic layouts and facility distributions, individuals organize their daily activities in similar ways across cities in terms of activity type, jump length, and time period González et al. (2008); Song et al. (2010b); Jiang et al. (2016b); Chen et al. (2026). Existing studies Wang et al. (2024c); Yuan et al. (2026); Wang et al. (2025) explore cross-city mobility transfer through shared and city-specific modeling, mobility road network relationships, and generative models, while agentic approaches capture individual differences for personalized mobility generation. Nevertheless, existing cross-city transfer approaches do not explicitly represent individual behavioral structure in their inputs or outputs, leaving any preservation of such structure implicit and difficult to verify. Existing methods transfer a set of parameters, a flow matrix or a distribution conditioned on the population and POI statistics of the target city. None of them transfers individual behavior, so the individual structure of their output is not controlled.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

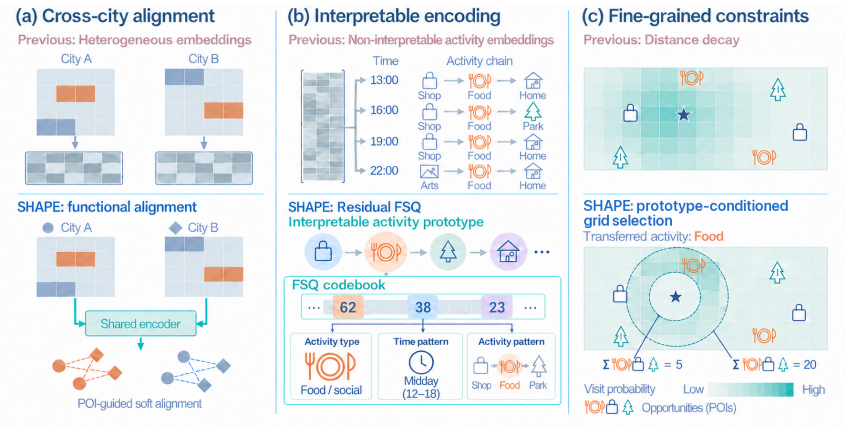


Figure 1: The primary modeling advantage of SHAPE from three aspects. **(a) cross-city alignment.** We design a shared attribute encoder that aligns grid cells across cities by POI composition. **(b) Interpretable encoding.** We design a residual FSQ encoder to quantize activity trajectory into an interpretable activity prototype. **(c) Fine-grained constraints.** We let the activity prototype refine grid cell selection among the reachable cells.

In this work, we study the cross-city mobility generation by transferring the activity sequences of individuals in one city to another. Given individual mobilities with location from a source city and static geographic information from a target city, the objective is to construct new spatial realizations of those activities. Effective cross-city transfer requires aligning individual activity patterns of source-city activities with the urban environment in target city, which presents three interconnected challenges. First, denoted by previous work Wang et al. (2024c), city-specific location representations are not directly transferable across cities due to domain heterogeneity. Second, previous methods Wang et al. (2024c; 2025); Yuan et al. (2026) in cross-city transfer are based on continuous latent representations, which lack interpretability. Third, grounding transferred activities in the target city requires jointly satisfying activity-specific functional requirements and movement constraints. Pappalardo et al. (2016).

To address these challenges, we propose Shared Human Activity Prototype Extrapolation (SHAPE), a cross-city mobility generation framework that uses discrete activity prototypes to mediate the transfer of individual activity sequences across urban environments. Figure 1 illustrates the three complementary advantages in SHAPE. First, to establish alignment between heterogeneous locations, SHAPE constructs a shared embedding space that aligns locations according to their functional characteristics via a shared attribute encoder. Second, SHAPE uses residual Finite Scalar Quantization (FSQ) Mentzer et al. (2024) to encode each activity with its sequential context into a discrete activity prototype, distinguishing activities with the same semantic label but different roles in an individual's routine (Fig. 1b). Third, SHAPE uses these prototypes to guide location selection in the target city with the static map of the target city and the spatial distribution of its residents (Fig. 1c). Together, these designs connect the preservation of individual behavioral structure with adaptation to the target urban environment.

Comparing with three types of frontier methods, we evaluate SHAPE on the aggregate mobility distributions and individual mobility patterns it generates for the target city, and examine the interpretability of its city-independent discrete activity prototypes. SHAPE achieves state-of-the-art performance across multiple metrics, while ablation studies support the contributions of all three core designs. With transferred prototypes and limited target-city information, it reaches 90.1% of the full-information oracle's OD CPC, highlighting the effectiveness of prototype-based behavioral transfer for target-city mobility generation.

To summarize, our work has following contributions:

- We formulate zero-shot cross-city mobility generation, in which the model accesses the target city only through its static map and residential distribution, and address it by transferring activity patterns as discrete, city-independent activity prototypes.

- We propose SHAPE, which learns the prototypes from location-free inputs and places them in the target city by the opportunity a move passes over rather than by distance, so neither step depends on the geography of the source city.
- On four source-to-target cases, SHAPE ranks first on four of the five metrics against three families of baselines. Mechanistic models transfer across cities but score low, whereas data-driven models score high in their own city and drop sharply across cities. SHAPE separates the activity pattern from its placement in the target city. On NYC→Paris, it loses little across cities and reaches 90.1% of the OD CPC of an oracle given the trajectories of Paris.

2 RELATED WORK

Human mobility generation. Human mobility generation spans mechanistic models Pappalardo et al. (2016); Simini et al. (2021), learned generators Ma et al. (2024); Li et al. (2026), and LLM agents Wang et al. (2024a); Chen et al. (2026). Mechanistic models build on bounded exploration, preferential return, and mobility predictability (González et al., 2008; Song et al., 2010b;a). Gravity and radiation models (Zipf, 1946; Simini et al., 2012) generate aggregate flows, while TimeGeo (Jiang et al., 2016a) and DITRAS (Pappalardo & Simini, 2018) separate activity scheduling from location assignment. Learned approaches fit mobility distributions through origin–destination models (Rong et al., 2023; 2024) and diffusion generators; we adopt the evaluation protocol of MIRAGE (Deng et al., 2025). MobAgent (Li et al., 2024) instead generates mobility through natural-language reasoning. Mechanistic models offer transferability with limited expressiveness, whereas learned generators can achieve higher accuracy but often rely on city-specific representations. LLM agents may draw on target-city knowledge acquired during pretraining, complicating control over available information.

Cross-city knowledge transfer. Cross-city transfer reuses knowledge from data-rich cities in less-observed ones (Zheng et al., 2014; Wei et al., 2016; Wang et al., 2019). One line transfers source-fitted components: origin–destination feature mappings in Deep Gravity (Simini et al., 2021), region embeddings and source-city retrieval indices in CRAFT (Li et al., 2025), and road-network topology in GTG (Wang et al., 2025). These components remain vulnerable to cross-city distribution shifts (Zhou et al., 2026). Another line transfers behavioral regularities through d-EPR's exploration parameters (Pappalardo et al., 2016) and WorldMove's diffusion prior conditioned on population, POI, and commuting statistics (Yuan et al., 2026); Map2Traj (Tao et al., 2025) instead generates trajectories from street maps alone. LLM approaches transfer knowledge through MobTA's task vector (Liu et al., 2026b) or UniMob's unified planner (Liu et al., 2026a). In contrast, SHAPE transfers an explicit, inspectable codebook of city-independent activity prototypes, retaining individual detail while separating shared behavior from city-specific geography.

Discrete representation learning. Vector quantization (VQ) learns compact token vocabularies for image generation and style transfer (van den Oord et al., 2017; Esser et al., 2021; Chen et al., 2022; Huang et al., 2023). In traffic modeling, Trajeglish (Philion et al., 2024) tokenizes vehicle motion, and SMART (Wu et al., 2024) extends this approach to zero-shot cross-dataset transfer. Applying discretization to human activity calls for stable quantization and representations of both activity identity and finer variation. VQ can suffer from codebook collapse and dead codes (Huh et al., 2023; Zheng & Vedaldi, 2023); finite scalar quantization (FSQ) (Mentzer et al., 2024) avoids codebook learning by quantizing each dimension onto a fixed grid. Residual quantization captures finer variation by encoding residuals across successive levels (Lee et al., 2022; Zhang et al., 2024; Rajput et al., 2023). SHAPE uses residual FSQ to encode activities as hierarchical activity prototypes interpretable in terms of activity type, jump length, and time period, providing a discrete medium for cross-city behavioral transfer.

3 METHODOLOGY

3.1 PROBLEM DEFINITION

We formulate our task as *zero-shot cross-city mobility generation*. A data-rich source city A provides observed trajectories of its residents, whereas a data-scarce target city B provides no individual

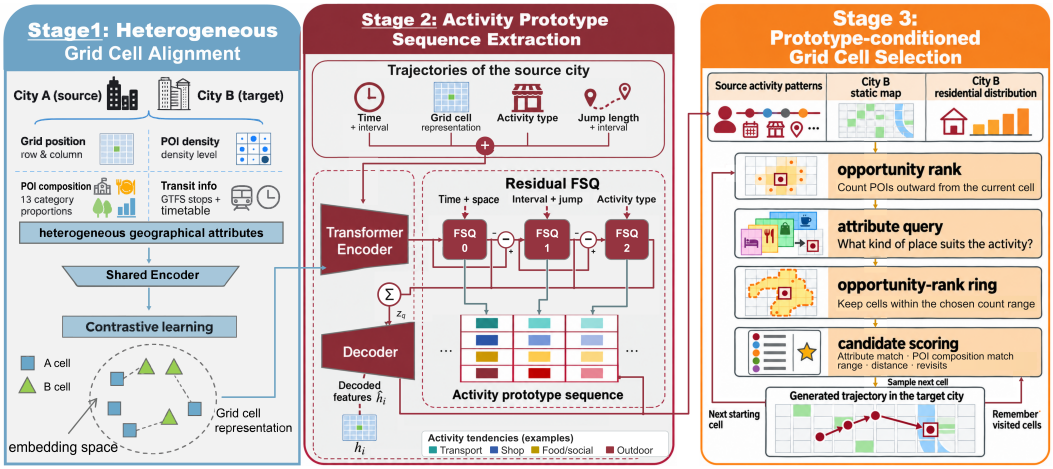


Figure 2: Architecture and flowchart of SHAPE. Stage 1 realizes the cross-city heterogeneous grid cell alignment using contrastive learning. Stage 2 uses RFSQ to compress the trajectories of the source city to activity prototype sequences. Stage 3 achieves prototype-conditioned grid cell selection in the target city by an opportunity-rank ring and candidate scoring.

trajectory, only its static map and the spatial distribution of its residents. Our goal is to generate individual trajectories for B by transferring the activity patterns of the residents of A. For each resident u of A, the trajectory $X_u^A = \big((t_i, g_i^A, c_i)\big)_{i=1}^{T_u}$ records the time, grid cell, and activity type of the i-th activity, where the activity type c_i is derived from the category of the visited POI (Yang et al., 2015b). Activity types and POI categories therefore share the same 13 classes. The corresponding activity pattern of X_u^A is the sequence $\big((t_i, c_i)\big)_{i=1}^{T_u}$ that keeps the activity types and their times but drops the grid cells. We keep this activity pattern as it is and map it onto B by assigning each activity a grid cell g_i^B of B, which turns it into a trajectory $X_u^B = \big((t_i, g_i^B, c_i)\big)_{i=1}^{T_u}$ in B. To carry out this mapping, we use only two inputs from B. The first is its static map $\mathcal{A}^B$, which gives the geographical attributes of each grid cell, including the number of POIs and their composition over the 13 POI categories, as well as its transit information. The second is its residential distribution H^B, which gives the share of the residents of B who live in each grid cell. The static map $\mathcal{A}^A$ of A gives the same attributes for the source city to help the model align the geographical features of two cities. The model is trained on A alone and never uses an individual trajectory of B, so the task is zero-shot at the individual level. In summary, the task is a chain of three mappings,

$$(h^A, h^B) = F_1(\mathcal{A}^A, \mathcal{A}^B), \qquad C_u = F_2(X_u^A, h^A), \qquad X_u^B = F_3(C_u, \mathcal{A}^B, H^B), \tag{1}$$

where F_1 aligns the grid cells of the two cities in a shared embedding space (h^A, h^B), F_2 encodes the activity pattern of a trajectory of A as a sequence of activity prototypes C_u, and F_3 places each prototype in a grid cell of B. B enters this chain only through $\mathcal{A}^B$ and H^B.

Assumption. Based on the shared structural regularities of daily activity patterns across cities (González et al., 2008; Schläpfer et al., 2021; Alessandretti et al., 2020), we assume that the activity pattern of a resident of A can be transferred to B unchanged. Important mechanistic models such as TimeGeo and DITRAS simulate mobility in two steps, first a daily activity schedule and then the locations of each activity (Jiang et al., 2016a; Pappalardo & Simini, 2018). Our assumption follows this split, keeping the schedule from A and choosing only the locations in B.

3.2 Overall Framework

The overall design of SHAPE is illustrated in Figure 2 that comprises three stages. They correspond to F_1, F_2 and F_3 in Eq. equation 1. Stage 1 aligns the grid cells of the two cities in a shared embedding space, Stage 2 encodes the trajectories of the source city into activity prototype sequences, and Stage 3 places each prototype in a grid cell of B. **Stage 1** aligns the grid cells of the two cities in a shared embedding space. A shared encoder maps the attributes of each grid cell, its position, POI density, POI composition and transit service, into this space. Cross-city contrastive learning

trains it so that grid cells serving similar functions have similar representations. **Stage 2** turns each trajectory of the source city into an activity prototype sequence. Each activity is described by its time, its activity type and the semantic representation of its grid cell. A Transformer encoder captures the context of the whole trajectory, and residual finite scalar quantization (RFSQ) then assigns each activity to one prototype from a discrete codebook. This sequence of prototypes is the encoded activity pattern of the trajectory. This quantization is what makes the activity pattern transferable. A decoder maps the codes back to a place feature $\widehat{h}_i$, trained to reconstruct the representation of the visited grid cell. **Stage 3** grounds the activity prototypes in the target city. Each prototype encodes the mobility pattern of an activity, and a query head maps the place feature $\widehat{h}_i$ into the attribute space of the static map, where it is compared with the grid cells of the target city. Each prototype is matched to a cell that fits its pattern. The matched cells form the generated trajectory in the target city. Algorithm 1 in Appendix A.2 lists the full procedure; the three stages are detailed there as well.

4 EXPERIMENT

4.1 EXPERIMENT SETTING

Dataset & Protocol. SHAPE and the baselines use the check-ins of three cities, Foursquare New York City from the NYC and Tokyo check-in dataset (released as `dataset_TSMC2014_NYC`) (Yang et al., 2015b), Foursquare Paris extracted from the global-scale TIST2015 dataset (Yang et al., 2016; 2015a) and Gowalla California (Cho et al., 2011). NYC and California hold far more check-ins and longer activity sequences than Paris, so they serve as source cities, and each is transferred to the other two. This gives four directions, NYC→Paris and CA→Paris from a data-rich city to a data-scarce one, and NYC→CA and CA→NYC between two data-rich cities. Every experiment is run with five seeds. Dataset statistics are given in Appendix A.1 and the protocol in the reproducibility statement.

Information input. All methods receive the same inputs, the trajectories of the source city, the static maps of both cities and the residential distribution of the target city, and each uses the subset its formulation admits. No method observes a trajectory of the target city during generation. The residential distribution is estimated from the training days of the target city, whose test days are used only for evaluation.

Baselines. We compare SHAPE with three families of methods. **(1) Mechanistic models**, which capture context-free or distance-decay dynamics: *d-EPR* Pappalardo et al. (2016) (density-driven exploration with preferential return), *Radiation* Simini et al. (2012) (intervening opportunities), and *DeepGravity* Simini et al. (2021) (gravity-decay spatial interactions). **(2) Supervised OD models**, assessing transferability against direct OD regression: *CommutingODGen* Huang et al. (2024), spanning tree- and graph-based adapters (*ODGen-GBRT*, *ODGen-RF*, and *ODGen-GNN*) that map regional attributes to flow intensities. **(3) Diffusion-based generators**, validating end-to-end trajectory synthesis: *WorldMove* Yuan et al. (2026) (individual trajectory synthesis) and *CRAFT* Li et al. (2025) (regional flow generation via feature-aligned source retrieval).

Evaluation metrics. We use five metrics, following Yuan et al. (2026); Hong et al. (2026); Wang et al. (2024b). Three describe aggregate mobility distributions. **OD CPC** (↑) is the common part of commuters between the generated and the real origin–destination matrices. **Spatiotemporal JSD** (↓) is the divergence between the generated and the real visit distributions over grid cells, averaged over four intraday periods (06–11, 11–17, 17–22, 22–06). **POI-category JSD** (↓) is the divergence between the distributions of the POI categories of visited grid cells. Two describe individual mobility patterns. **RoG JSD** (↓) is the divergence between the distributions of the radius of gyration of residents, and **Jump-length JSD** (↓) the divergence between the distributions of jump lengths between consecutive activities.

4.2 PERFORMANCE AGAINST BASELINES

Table 1 compares all methods on the four source-to-target cases. SHAPE ranks first on OD CPC, Spatiotemporal JSD, Jump-length JSD and POI-category JSD in all four cases, and on RoG JSD in the two cases from CA. On RoG JSD from NYC it ranks second, behind Deep Gravity for Paris and d-EPR for CA. On OD CPC, the gain over the strongest baseline of each case ranges from

Table 1: Zero-shot cross-city generation on four source-to-target directions. Best in bold, second best underlined. ↑ higher is better, ↓ lower is better. CPC: common part of commuters; JSD: Jensen–Shannon divergence; RoG: radius of gyration.

NYC → Paris

Method	Flow: OD CPC ↑	Distribution: Spatio-temporal JSD ↓	Individual pattern: RoG JSD ↓	Individual pattern: Jump-Length JSD ↓	Individual pattern: POI-Category JSD ↓
Radiation	0.5225 ±0.0076	0.2659 ±0.0118	0.0183 ±0.0030	0.1611 ±0.0030	0.0012 ±0.0001
d-EPR	0.4904 ±0.0088	0.2717 ±0.0087	0.0058 ±0.0017	0.0423 ±0.0007	0.0018 ±0.0002
Deep Gravity	0.5203 ±0.0046	0.2769 ±0.0092	**0.0018 ±0.0006**	0.0551 ±0.0010	0.0019 ±0.0001
ODGen-GBRT	0.3295 ±0.0057	0.3047 ±0.0123	0.1146 ±0.0098	0.1378 ±0.0087	0.0031 ±0.0002
ODGen-RF	0.2885 ±0.0125	0.3223 ±0.0112	0.1688 ±0.0126	0.1949 ±0.0069	0.0048 ±0.0002
ODGen-GNN	0.3276 ±0.0221	0.2919 ±0.0100	0.1229 ±0.0095	0.1588 ±0.0128	0.0031 ±0.0004
CRAFT	0.5387 ±0.0159	0.2896 ±0.0101	0.0625 ±0.0077	0.0355 ±0.0072	0.0008 ±0.0002
WorldMove	0.4702 ±0.0241	0.3000 ±0.0177	0.0561 ±0.0167	0.0299 ±0.0081	0.0008 ±0.0003
SHAPE (ours)	**0.5806 ±0.0033**	**0.2500 ±0.0027**	0.0034 ±0.0004	**0.0292 ±0.0013**	**0.0002 ±0.0000**

NYC → CA

Method	Flow: OD CPC ↑	Distribution: Spatio-temporal JSD ↓	Individual pattern: RoG JSD ↓	Individual pattern: Jump-Length JSD ↓	Individual pattern: POI-Category JSD ↓
Radiation	0.5242 ±0.0189	0.5706 ±0.0043	0.1852 ±0.0051	0.1569 ±0.0022	0.0040 ±0.0005
d-EPR	0.5943 ±0.0037	0.8206 ±0.0034	**0.0460 ±0.0020**	0.1270 ±0.0033	0.0217 ±0.0005
Deep Gravity	0.5334 ±0.0129	0.5788 ±0.0182	0.2629 ±0.0242	0.1359 ±0.0155	0.0031 ±0.0018
ODGen-GBRT	0.2202 ±0.0054	0.6455 ±0.0047	0.5917 ±0.0116	0.6193 ±0.0067	0.0104 ±0.0003
ODGen-RF	0.2488 ±0.0068	0.6592 ±0.0027	0.5677 ±0.0107	0.6006 ±0.0067	0.0116 ±0.0003
ODGen-GNN	0.6782 ±0.0096	0.6360 ±0.0046	0.1814 ±0.0060	0.2991 ±0.0082	0.0165 ±0.0006
CRAFT	0.6383 ±0.0038	0.5238 ±0.0019	0.1952 ±0.0019	0.1285 ±0.0044	0.0028 ±0.0008
WorldMove	0.6710 ±0.0099	0.5381 ±0.0036	0.2051 ±0.0021	0.1547 ±0.0012	0.0037 ±0.0007
SHAPE (ours)	**0.8330 ±0.0054**	**0.4058 ±0.0024**	0.1777 ±0.0032	**0.1248 ±0.0013**	**0.0015 ±0.0001**

CA → NYC

Method	Flow: OD CPC ↑	Distribution: Spatio-temporal JSD ↓	Individual pattern: RoG JSD ↓	Individual pattern: Jump-Length JSD ↓	Individual pattern: POI-Category JSD ↓
Radiation	0.4458 ±0.0032	0.4033 ±0.0050	0.2627 ±0.0031	0.1824 ±0.0023	0.0024 ±0.0001
d-EPR	0.4573 ±0.0053	0.4272 ±0.0059	0.2873 ±0.0078	0.1243 ±0.0012	0.0020 ±0.0001
Deep Gravity	0.4593 ±0.1537	0.4642 ±0.0551	0.5098 ±0.1704	0.2097 ±0.1422	0.0024 ±0.0007
ODGen-GBRT	0.2583 ±0.0034	0.5670 ±0.0024	0.6823 ±0.0160	0.4229 ±0.0059	0.0066 ±0.0003
ODGen-RF	0.1895 ±0.0051	0.5990 ±0.0019	0.7538 ±0.0058	0.5243 ±0.0051	0.0083 ±0.0001
ODGen-GNN	0.1449 ±0.0870	0.6439 ±0.0609	0.8046 ±0.0855	0.3611 ±0.1110	0.0239 ±0.0225
CRAFT	0.5066 ±0.0040	0.4066 ±0.0035	0.2542 ±0.0132	0.1231 ±0.0075	0.0029 ±0.0001
WorldMove	0.5193 ±0.0603	0.4786 ±0.0171	0.5305 ±0.0873	0.1229 ±0.0275	0.0036 ±0.0023
SHAPE (ours)	**0.6242 ±0.0092**	**0.3673 ±0.0020**	**0.2152 ±0.0116**	**0.0692 ±0.0032**	**0.0015 ±0.0001**

CA → Paris

Method	Flow: OD CPC ↑	Distribution: Spatio-temporal JSD ↓	Individual pattern: RoG JSD ↓	Individual pattern: Jump-Length JSD ↓	Individual pattern: POI-Category JSD ↓
Radiation	0.5224 ±0.0055	0.3274 ±0.0110	0.0186 ±0.0023	0.1622 ±0.0043	0.0033 ±0.0002
d-EPR	0.4971 ±0.0061	0.3290 ±0.0078	0.0052 ±0.0012	0.0228 ±0.0007	0.0048 ±0.0001
Deep Gravity	0.4757 ±0.0063	0.3071 ±0.0047	0.0153 ±0.0029	0.0458 ±0.0010	0.0025 ±0.0002
ODGen-GBRT	0.1964 ±0.0061	0.3948 ±0.0106	0.2804 ±0.0060	0.3092 ±0.0050	0.0083 ±0.0004
ODGen-RF	0.1985 ±0.0079	0.4026 ±0.0091	0.2828 ±0.0124	0.3065 ±0.0091	0.0088 ±0.0006
ODGen-GNN	0.3790 ±0.1196	0.3555 ±0.1314	0.1751 ±0.1671	0.0800 ±0.0093	0.0071 ±0.0095
CRAFT	0.5114 ±0.0495	0.3012 ±0.0095	0.0834 ±0.0101	0.0346 ±0.0019	0.0022 ±0.0004
WorldMove	0.5007 ±0.0289	0.3135 ±0.0081	0.0265 ±0.0046	0.0364 ±0.0095	0.0028 ±0.0018
SHAPE (ours)	**0.5663 ±0.0030**	**0.2855 ±0.0027**	**0.0025 ±0.0003**	**0.0226 ±0.0015**	**0.0002 ±0.0000**

7.8% to 22.8%, the largest in NYC→CA, where SHAPE reaches 0.8330 against 0.6782 for ODGen-GNN. The margin is largest on POI-category JSD, where SHAPE improves on the best baseline by 25% to 90.9% (median 60.7%), for example 0.0002 against 0.0022 for CRAFT in CA→Paris. The prototype carries the activity type, which accounts for this margin. On Jump-Length JSD, the lead is small in two cases (0.0292 against 0.0299 in NYC→Paris, 0.1248 against 0.1270 in NYC→CA). SHAPE models mobility one activity at a time. This is what makes a pattern transferable, since each step is placed by what the activity needs and how far it goes, both stated in city-independent terms. The trade-off is that the spread of the whole day is not well controlled. The radius of gyration is a measure of this spread. SHAPE has no term for it, so the value is determined only indirectly, by the steps taken over the day. Exploration and preferential return was designed around this statistic (Song et al., 2010a), and d-EPR inherits it. SHAPE is therefore second on RoG JSD in NYC→Paris (0.0034 against 0.0018 for Deep Gravity) and NYC→CA (0.1777 against 0.0460 for d-EPR).

The three families differ in what they model, which determines the metrics on which each is accurate. Mechanistic models describe mobility with a law of a few parameters and are accurate only on the statistic that law parameterizes. d-EPR gives the lowest RoG JSD in NYC→CA (0.0460) and the

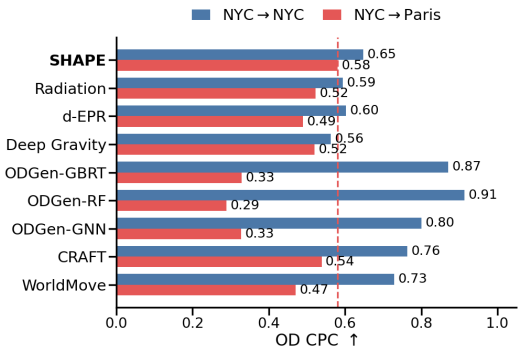
highest Spatiotemporal JSD in the table (0.8206). Supervised OD models regress region-pair flows, which contain no individual-level structure, so the individual metrics fail even when OD CPC is competitive. ODGen-GNN has the second-best OD CPC in NYC→CA (0.6782) with RoG, Jump-length and POI-category JSD of 0.1814, 0.2991 and 0.0165. Diffusion-based generators model the trajectory distribution of the source city conditioned on aggregates of the target city, so they reproduce aggregate distributions rather than individual patterns. Averaged over the four cases, CRAFT and WorldMove are the strongest baselines on OD CPC (0.5488 and 0.5403) and CRAFT the strongest on POI-category JSD (0.0022), while on RoG JSD both rank below d-EPR (0.1488 and 0.2045 against 0.0861). None of the three models individual behavior in a city-independent form; SHAPE does.

Figure 3a plots OD CPC in the intra-city and the cross-city setting. Mechanistic models lose little when crossing cities but reach a low level in both settings. Data-driven models reach the highest intra-city values and drop sharply across cities, because what they fit includes the spatial correlations of one city. SHAPE models the activity pattern and decouples it from the geography of any one city, so its capacity is spent on behaviour rather than on the spatial correlations of the source city, and its drop across cities is small.

4.3 ORACLE ANALYSIS

We run an oracle test on NYC→Paris (Figure 3b). Starting from the standard configuration, we give the model one more piece of information about Paris at a time. The standard configuration uses only the static map and the residential distribution of Paris. The second configuration adds the jump-length distribution of Paris. The third replaces the activity prototypes of NYC with prototypes estimated on the trajectories of Paris. The fourth adds the true home cells and personal location memories from the Paris training days and reaches the oracle.

The standard configuration reaches 90.1% of the oracle on OD CPC and 78.6% on Spatiotemporal JSD. The three additions add the remaining 9.9 and 21.4 points. The jump-length distribution adds 3.5 and 9.5, the Paris prototypes 5.5 and 5.5, and the true homes 0.8 and 6.4. The quantities that SHAPE transfers without units are thus close to the oracle already. The one quantity in kilometres is not. On Jump-length JSD the standard configuration reaches 2.8%, and the jump-length distribution of Paris alone adds 80.2 points. On RoG JSD the true homes add 46.9 points and the other two additions 2.5 together, since the radius of gyration is anchored on the home. On POI-category JSD the standard configuration is above the oracle at 265%. The metric is a population-level share, which SHAPE matches by construction when it renormalizes prototype attraction on the POI counts of Paris.



(a) OD CPC within NYC and from NYC to Paris.

	①	②	③	④
OD CPC ↑	90.1%	93.7% +3.5	99.2% +5.5	100% +0.8
Spatiotemporal JSD ↓	78.6%	88.1% +9.5	93.6% +5.5	100% +6.4
RoG JSD ↓	50.6%	47.8% −2.8	53.1% +5.3	100% +46.9
Jump-length JSD ↓	2.8%	83.0% +80.2	83.8% +0.8	100% +16.2
POI-category JSD ↓	265.0%	176.7% −88.3	196.3% +19.6	100% −96.3

① Paris static map + residential distribution
② + Paris jump-length distribution
③ Replace NYC with Paris activity prototypes
④ + true home cells
+ personal location memories (oracle)

(b) Oracle test on NYC→Paris.

Figure 3: (a) Blue bars are intra-city generation, red bars the transfer to Paris. (b) Columns 1–4 represent configurations with increasing access to Paris data, ending with the oracle. Entries are percentages of the oracle, oriented so that higher is better in every row; the small number is the change from the column on the left, in percentage points. *Shading highlights the main results discussed in the text.*

5 ACTIVITY PROTOTYPES IN THE HIERARCHICAL CODEBOOK

This section analyzes what the activity prototypes encode. For each prototype, the activities assigned to it on held-out NYC user-days give the conditional distributions of activity type, jump-length bin and time period. $L0$ is tied to the time and to $h^A_{g_i}$, the functional representation of the visited grid cell built from its POIs and transit, so besides discretizing the activity pattern it learns a mapping from each prototype to the function of the places where that activity happens. $L1$ is tied to the interval and the jump length, and $L2$ to the activity type; both refine what $L0$ fixes.

Figure 4 shows eight of the 64 $L0$ prototypes. Activity type is the most concentrated attribute. In seven of the eight prototypes the dominant type accounts for 74% to 97% of the activities, so $L0$ organizes activities primarily by the type of place. Prototypes with the same dominant type differ in movement range and time period. Food/social comprises a stationary midday prototype (code 62), a local evening prototype (code 41, local 99%) and a local evening prototype with a long-range share (code 20); Shop comprises a local afternoon prototype (code 47) and a stationary prototype (code 63); Transport comprises a long-range prototype (code 1) and a local prototype (code 23). Time period is the least concentrated attribute, with majorities of 42% to 57% and no majority for the two transport prototypes. A prototype therefore specifies a type of place, a movement range and a time period, none of which refers to a location. A local evening food/social prototype specifies a food or social venue within local range of the previous cell in the evening, which Paris provides from its own venues.

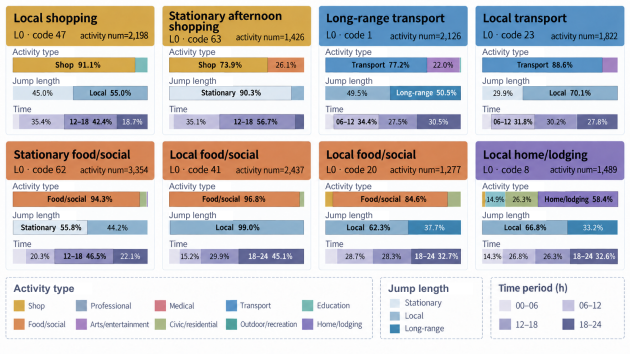


Figure 4: Eight of the 64 $L0$ activity prototypes. Each card names one prototype and gives its code index and the number n of held-out activities assigned to it. The three bars show the tendency of activity pattern in activity type, jump length, and time.

$L1$ and $L2$ refine the prototypes given the preceding codes. Measured as the reduction in held-out probe cross-entropy, $L0$ carries 2.106 bits on activity type and 0.661 bits on jump-length bin, $L1$ adds 0.835 and 0.131 bits, and $L2$ adds at most 0.068 bits on any attribute (Table 6 in Appendix A.4).

In our model, quantization does not collapse individuals, because a resident is represented not by a single code but by the distribution of prototypes over the trajectory, refined by the three residual layers. Held-out user retrieval from the code distributions alone rises with residual depth, from 3.5% to 6.6% to 7.9% top-1 against a 0.092% chance level. The chance level corresponds to about 1,100 candidate residents, so the codes pick out the right resident about 86 times more often than chance, and each residual layer adds to this.

6 ABLATION STUDY

We ablate the three stages of SHAPE on NYC→Paris, grid cell alignment (Stage 1), activity prototype extraction (Stage 2) and prototype-conditioned grid cell selection (Stage 3). Table 2 covers all seven non-empty combinations, evaluated with the metrics of Table 1.

Cross-city Grid Cell Alignment (Stage 1). Stage 1 alone is the weakest setting on every metric (OD CPC 0.1823, RoG JSD 0.7006), since an alignment of grid cells generates no mobility by itself. Its effect appears when it is added to a setting that does generate mobility. Added to Stage 2, Stage 3 or Stage 2+3, it raises OD CPC by 59%, 27% and 37% and lowers RoG JSD by 78%, 49% and

96%, whereas Spatiotemporal JSD changes by 4% to 11% only. The three removals from the full model make the same point. Removing Stage 1 lowers OD CPC by 26.8% and raises RoG JSD by a factor of 27 and Jump-length JSD by a factor of 2.3, but raises POI-category JSD by a factor of 9 and Spatiotemporal JSD by 6%, against factors of 107 and 26% for removing Stage 2 and 215 and 28% for removing Stage 3. Stage 1 thus determines where activities land relative to one another, which the flow and spread metrics measure, and contributes little to which kind of place or which hour they land in.

Activity Prototype Encoding (Stage 2). Stage 2 alone recovers little flow (OD CPC 0.2146), since prototypes state how people move and what they do, not where. Added to Stage 1, Stage 3 or Stage 1+3, it lowers Jump-length JSD by 82%, 69% and 83%, POI-category JSD by 73%, 94% and 99%, and Spatiotemporal JSD by 51%, 19% and 21%, and raises OD CPC by 87%, 11% and 19%. Removing it from the full model raises Jump-length JSD by a factor of 5.8 (0.0292 to 0.1682), the largest such penalty of the three removals, and POI-category JSD by a factor of 107 (0.0002 to 0.0214), while OD CPC falls by 16.0%, the smallest loss. Stage 2 thus determines how far an activity moves, what kind of place it visits and when, and of the three stages it contributes least to how much flow passes between places.

Prototype-conditioned Grid Cell Selection (Stage 3). Stage 3 alone recovers two thirds of the full model's OD CPC (0.3833 of 0.5806) and is the best single stage on four of the five metrics; only Jump-length JSD favours Stage 2 (0.1782 against 0.2205). Added to Stage 1, Stage 2 or Stage 1+2, it raises OD CPC by 167%, 98% and 70%, the largest gains of any stage. Removing it from the full model lowers OD CPC by 41.2% (0.5806 to 0.3412) and raises RoG JSD by a factor of 37.6 and POI-category JSD by a factor of 215, the largest penalty of the three removals on each of these metrics. Stage 3 places every prototype within a ring of opportunity rank around the previous cell, so the physical scale of each move is set by the map of the target city. None of the three stages suffices alone.

Table 2: Ablation of the three stages on NYC to Paris. Each row names the stages retained, and FULL is the complete model. Arrows mark the preferred direction. Entries are mean ± s.d. over five generation seeds; best in bold.

Setting	OD CPC ↑	Spatiotemporal JSD ↓	RoG JSD ↓	Jump-Length JSD ↓	POI-Category JSD ↓
Stage 1 only	0.1823 ± 0.0075	0.6532 ± 0.0145	0.7006 ± 0.0244	0.6356 ± 0.0122	0.1597 ± 0.0185
Stage 2 only	0.2146 ± 0.0055	0.3564 ± 0.0120	0.5844 ± 0.0385	0.1782 ± 0.0028	0.1216 ± 0.0728
Stage 3 only	0.3833 ± 0.0015	0.3269 ± 0.0010	0.1933 ± 0.0151	0.2205 ± 0.0013	0.0325 ± 0.0011
Stage 1+2	0.3412 ± 0.0013	0.3189 ± 0.0015	0.1280 ± 0.0073	0.1115 ± 0.0049	0.0430 ± 0.0004
Stage 1+3	0.4875 ± 0.0062	0.3147 ± 0.0028	0.0990 ± 0.0037	0.1682 ± 0.0016	0.0214 ± 0.0014
Stage 2+3	0.4249 ± 0.0071	0.2651 ± 0.0027	0.0921 ± 0.0010	0.0682 ± 0.0016	0.0018 ± 0.0002
FULL	**0.5806 ± 0.0033**	**0.2500 ± 0.0027**	**0.0034 ± 0.0004**	**0.0292 ± 0.0013**	**0.0002 ± 0.0000**

Appendix A.3 compares FSQ with EMA-VQ for Stage 2. EMA-VQ collapses onto fewer than five codes, which is why Stage 2 uses FSQ.

7 CONCLUSION

We studied zero-shot cross-city mobility generation, in which the model accesses the target city only through its static map and residential distribution. SHAPE transfers activity patterns as discrete, city-independent activity prototypes and places them with the geography of the target city. On four transfer cases it ranks first on four of the five metrics and at least second on the remaining one. Without any individual trajectory of the target city, it reaches 90.1% of the OD CPC of an oracle given those trajectories. Each prototype is readable by activity type, jump length and time period. Two limitations remain. First, the generated trajectories keep the timestamps of the trajectories of the source city. Second, because SHAPE places activities one at a time, it does not constrain the spatial extent of a whole day, and on RoG JSD it ranks second to mechanistic models in two of the four cases. Modeling the day-level structure of a trajectory is left for future work.

AI USE STATEMENT

In this work, we used generative AI tools for tasks whose disclosure is required: helping develop conceptual framings of the method, proposing and refining hypotheses, giving feedback on research methodology and experiments, implementing methods, cleaning and reformatting datasets, supporting data analysis, interpreting results, and assisting with translation. We have not used generative AI tools to generate synthetic datasets; the trajectories our method produces are outputs of the model we propose, and no data used for training or evaluation was produced by a generative AI tool. The remaining required-disclosure tasks, namely formulating mathematical claims, providing critical ingredients for proving them, and assisting in the writing of proofs, are not applicable to this work, which contains no formal theorems or proofs. Additionally, we used generative AI tools for tasks whose disclosure is recommended: identifying and summarizing relevant literature, searching for information, brainstorming, decomposing the work into subtasks, creating and editing software code, creating and modifying scientific figures, drafting parts of the manuscript, editing it for readability, suggesting its structure, and formatting references.

We have reviewed all AI-assisted work, and the review differed by task.

For literature and ideation, every reference suggested by a model was located and read by the authors before it entered the paper, and suggestions that could not be traced to a real and accessible source were discarded. Ideas produced during brainstorming were checked against the literature for overlap with existing work, and were kept only after the authors judged them feasible with the data and compute available to us and consistent with the ethical standards of the field.

For software, all AI-assisted code was read line by line by the authors. We identified and corrected errors introduced by the models, including fabricated interfaces and plausible but incorrect implementations, and we re-ran every affected experiment after each correction. All numbers reported in this paper come from runs of the corrected code.

For data analysis and figures, generative AI tools helped produce summaries, tables and scientific figures that make the results easier to read. The choice of what to measure, the interpretation of each result, and every claim drawn from it were made by the authors. We checked each figure against the underlying data and removed AI-generated statements that the data did not support.

For writing, generative AI tools produced drafts of some sections and suggested revisions for clarity. The authors rewrote and verified the final text, confirming that each sentence states what we intend it to state, that terminology is consistent across the paper, and that the wording reads naturally.

The authors take full responsibility for the final content of this work, including all text, claims, figures, code and experimental results produced with the aid of generative AI.

ETHICS STATEMENT

This work uses three public check-in datasets, Foursquare New York City, Foursquare Paris and Gowalla California, in which users appear only as numeric identifiers. We report only aggregate statistics and synthetic trajectories. The generated trajectories are outputs of the model and do not describe any real resident of the target city. Check-in data over-represent some population groups, and the transferred activity patterns inherit this coverage. The generated trajectories are an input for planning analyses, not a record of real behavior.

REPRODUCIBILITY STATEMENT

Hyperparameters. SHAPE consists of three stages—Stage 1: Heterogeneous Grid Cell Alignment; Stage 2: Activity Prototype Sequence Extraction; Stage 3: Prototype-conditioned Grid Cell Selection—whose hyperparameters are listed in the Table 3.

Baseline adaptation. To accommodate the baselines, d-EPR replaces endogenous waiting-time sampling with the shared activity timestamps; WorldMove preserves its diffusion backbone while adapting its geographic encoding interface, imputing missing conditions via source-domain exemplars, and mapping half-hourly trajectories to the shared timestamps; the ODGen variants predict intraday consecutive-activity flows from unified regional features and distances, with ODGen-GNN utilizing a neighborhood aggregation encoder and edge decoder to reconstruct activity sequences via

Table 3: Hyperparameters of the three stages. Stage 1 and Stage 2 are trained on the source city only; the Stage 3 query head is trained on source activities and frozen at transfer task, and this table lists the fixed settings shared by all four transfer directions.

Stage 1: Heterogeneous Grid Cell Alignment	
Static input dimension D_{static}	20
Transformer layers / heads	3 / 8
Hidden / feed-forward dimension	64 / 512
Projection dimension	64
Dropout	0.3
Negatives (random / hard)	90 / 30
Contrastive temperature τ_{cont}	0.07
Optimizer	AdamW
Learning rate	3×10^{-4}
Weight decay	1×10^{-5}
Batch size	256
Max epochs / patience	200 / 30
Stage 2: Activity Prototype Sequence Extraction	
Raw event dimension D_{raw}	136
Target / model dimension	64 / 256
Encoder / decoder layers	3 / 2
Attention heads / feed-forward	4 / 512
Dropout	0.1
Residual FSQ levels	3, $C = (4, 4, 4)$
Codebook capacity per level	64
Pre-quantization noise σ	0.1
Jump / POI auxiliary weight	0.125 / 0.075
FSQ diversity / semantic weight	0.01 / 0.02
Optimizer	AdamW
Learning rate	5×10^{-4}
Weight decay	1×10^{-4}
Gradient clipping	1.0
Batch size	196
Max epochs / patience	50 / 10

Stage 3: Query head (trained, then frozen)	
Head dimensions (in / hidden / out)	64 / 128 / 20
Optimizer	AdamW
Learning rate	1×10^{-3}
Weight decay	1×10^{-5}
Batch size	4096
Max epochs / patience	20 / 5
Attribute normalization	pooled q_{05}/q_{95}
Stage 3: Trajectory Generation (frozen)	
Distance semantics	opportunity rank
Coarse rank ring edges	$(0, 1, 5, 20, \infty)$
Fine segments per coarse ring	4
Candidate pool K_{cand}	512
Sampling temperature τ_{sample}	0.75
Distance mixture temperature	1.25
Attraction strength (raw)	1.0
Revisit weight β	1.0
Personal ring tilt λ	0
Personal revisit modulation κ	0
Attribute top-k	50
Home anchor top-k / neighbours	2 / 25
Travel-cost exponent γ	fitted on source train
Generation seeds	$\{42, \ldots, 46\}$
Evaluation shells / shell seed	200 / 42031

a unified sampling interface; and CRAFT is adapted to use the unified static attributes and grid adjacency structure, with its sequential adaptation results reported separately as a diagnostic reference.

To ensure time-scale and grid-scale consistency, the mobility sequences are aligned to local time and projected onto 500 m $\times$ 500 m grids. Check-in records from the target city are strictly withheld during training to ensure zero-shot transfer; target-domain residential distributions and distance statistics are exclusively utilized during generation and evaluation calibration.

REFERENCES

Laura Alessandretti, Ulf Aslak, and Sune Lehmann. The scales of human mobility. *Nature*, 587 (7834):402–407, 2020.

Linyao Chen, Qinlao Zhao, Zechen Li, Mingming Li, Likun Ni, Jinyu Chen, Yuhao Yao, Xuan Song, Noboru Koshizuka, and Hiroki Kobayashi. Towards efficient and evidence-grounded mobility prediction with llm-driven agent, 2026. URL https://arxiv.org/abs/2606.05130.

Quanjun Chen, Xuan Song, Harutoshi Yamada, and Ryosuke Shibasaki. Learning deep representation from big and heterogeneous data for traffic accident inference. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 338–344. AAAI Press, 2016.

Yu-Jie Chen, Shin-I Cheng, Wei-Chen Chiu, Hung-Yu Tseng, and Hsin-Ying Lee. Vector quantized image-to-image translation. In *Computer Vision – ECCV 2022*, volume 13676 of *Lecture Notes in Computer Science*, pp. 440–456. Springer Nature Switzerland, 2022. doi: 10.1007/978-3-031-19787-1_25. URL https://doi.org/10.1007/978-3-031-19787-1_25.

Eunjoon Cho, Seth A. Myers, and Jure 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*, pp. 1082–1090, 2011.

Bangchao Deng, Xin Jing, Tianyue Yang, Bingqing Qu, Dingqi Yang, and Philippe Cudré-Mauroux. Revisiting synthetic human trajectories: Imitative generation and benchmarks beyond datasaurus. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 201–212. Association for Computing Machinery, 2025. doi: 10.1145/3690624.3709180. URL https://doi.org/10.1145/3690624.3709180.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021. doi: 10.1109/cvpr46437.2021.01268. URL https://openaccess.thecvf.com/content/CVPR2021/html/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.html.

Marta C. González, César A. Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008. doi: 10.1038/nature06958. URL https://doi.org/10.1038/nature06958.

Marta C. González, César A. Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008. ISSN 1476-4687. doi: 10.1038/nature06958. URL http://dx.doi.org/10.1038/nature06958.

Ye Hong, Yatao Zhang, Konrad Schindler, and Martin Raubal. Deep generative model for human mobility behavior. *Transportation Research Part C: Emerging Technologies*, 192:105916, 2026. doi: 10.1016/j.trc.2026.105916. URL https://doi.org/10.1016/j.trc.2026.105916.

Siyu Huang, Jie An, Donglai Wei, Jiebo Luo, and Hanspeter Pfister. QuantArt: Quantizing image style transfer towards high visual fidelity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5947–5956, 2023. doi: 10.1109/cvpr52729.2023.00576. URL https://openaccess.thecvf.com/content/CVPR2023/html/Huang_QuantArt_Quantizing_Image_Style_Transfer_Towards_High_Visual_Fidelity_CVPR_2023_paper.html.

Yan Huang, Shun Yang, Fan Zhou, Ting Li, and Goce Trajcevski. Commutingodgen: A benchmark for commuting origin-destination flow generation. In *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, pp. 149–152, 2024. doi: 10.1145/3677386.3682079. URL https://doi.org/10.1145/3677386.3682079.

Minyoung Huh, Brian Cheung, Pulkit Agrawal, and Phillip Isola. Straightening out the straight-through estimator: Overcoming optimization challenges in vector quantized networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14096–14113. PMLR, 2023. URL https://proceedings.mlr.press/v202/huh23a.html.

Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. González. The TimeGeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences*, 113(37):E5370–E5378, 2016a. doi: 10.1073/pnas.1524261113. URL https://doi.org/10.1073/pnas.1524261113.

Shan Jiang, Yingxiang Yang, Siddharth Gupta, Daniele Veneziano, Shounak Athavale, and Marta C. González. The timegeo modeling framework for urban mobility without travel surveys. *Proceedings of the National Academy of Sciences*, 113(37):E5370–E5378, 2016b. doi: 10.1073/pnas.1524261113. URL https://www.pnas.org/doi/abs/10.1073/pnas.1524261113.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11523–11532, 2022. doi: 10.1109/cvpr52688.2022.01123.

URL https://openaccess.thecvf.com/content/CVPR2022/html/Lee_Autoregressive_Image_Generation_Using_Residual_Quantization_CVPR_2022_paper.html.

Peiran Li, Jiawei Wang, Haoran Zhang, Xiaodan Shi, Noboru Koshizuka, Chihiro Shimizu, and Renhe Jiang. Trajflow: Nation-wide pseudo gps trajectory generation with flow matching models, 2026. URL https://arxiv.org/abs/2603.15009.

Xuchuan Li, Fei Huang, Jianrong Lv, Zhixiong Xiao, Guolong Li, and Yang Yue. Be more real: Travel diary generation using LLM agents and individual profiles. arXiv preprint arXiv:2407.18932, 2024. URL https://arxiv.org/abs/2407.18932.

Yudong Li, Jingyuan Wang, Xie Yu, Peiyu Wang, and Qian Huang. Cross city traffic flow generation via retrieval augmented diffusion model. In *Advances in Neural Information Processing Systems*, volume 38, pp. 70025–70055. Curran Associates, Inc., 2025. doi: 10.52202/085713-2354. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/654f8bf54c8ad618dcca6cd43dd0830f-Abstract-Conference.html.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. doi: 10.1109/18.61115.

Bo Liu, Tong Li, Zhu Xiao, Ruihui Li, Geyong Min, Zhuo Tang, and Kenli Li. All cities are equal: A unified human mobility generation model enabled by LLMs. arXiv preprint arXiv:2602.19694, 2026a. URL https://arxiv.org/abs/2602.19694.

Shuai Liu, Ning Cao, Yile Chen, Yue Jiang, and Gao Cong. Bus-conditioned zero-shot trajectory generation via task arithmetic. arXiv preprint arXiv:2602.13071, 2026b. URL https://arxiv.org/abs/2602.13071.

Xuebin Ma, Zinan Ding, and Xiaoyan Zhang. St-trajgan: A synthetic trajectory generation algorithm for privacy preservation. *Future Generation Computer Systems*, 161:226–238, 2024. ISSN 0167-739X. doi: https://doi.org/10.1016/j.future.2024.07.011. URL https://www.sciencedirect.com/science/article/pii/S0167739X2400373X.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, pp. 51772–51783, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/e2dd53601de57c773343a7cdf09fae1c-Abstract-Conference.html.

Anastasios Noulas, Salvatore Scellato, Renaud Lambiotte, Massimiliano Pontil, and Cecilia Mascolo. A tale of many cities: Universal patterns in human urban mobility. *PLOS ONE*, 7(5):e37027, 2012. doi: 10.1371/journal.pone.0037027.

Luca Pappalardo and Filippo Simini. Data-driven generation of spatio-temporal routines in human mobility. *Data Mining and Knowledge Discovery*, 32(3):787–829, 2018. doi: 10.1007/s10618-017-0548-4. URL https://doi.org/10.1007/s10618-017-0548-4.

Luca Pappalardo, Filippo Simini, Salvatore Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-László Barabási. Returners and explorers dichotomy in human mobility. *Nature communications*, 6(1):8166, 2015.

Luca Pappalardo, Salvatore Rinzivillo, and Filippo Simini. Human mobility modelling: Exploration and preferential return meet the gravity model. *Procedia Computer Science*, 83:934–939, 2016. doi: 10.1016/j.procs.2016.04.188. URL https://doi.org/10.1016/j.procs.2016.04.188.

Jonah Philion, Xue Bin Peng, and Sanja Fidler. Trajeglish: Traffic modeling as next-token prediction. In *International Conference on Learning Representations*, 2024.

Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, Maciej Kula, Ed Chi, and Maheswaran Sathiamoorthy. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*, volume 36, pp. 10299–10315. Curran Associates, Inc., 2023. doi: 10.52202/075280-0452. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/20dcab0f14046a5c6b02b61da9f13229-Abstract-Conference.html.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Can Rong, Huandong Wang, and Yong Li. Origin-destination network generation via gravity-guided GAN. arXiv preprint arXiv:2306.03390, 2023. URL https://arxiv.org/abs/2306.03390.

Can Rong, Jingtao Ding, Yan Liu, and Yong Li. A large-scale benchmark dataset for commuting origin-destination matrix generation. arXiv preprint arXiv:2407.15823, 2024. URL https://arxiv.org/abs/2407.15823.

Markus Schläpfer, Lei Dong, Kevin O'Keeffe, Paolo Santi, Michael Szell, Hadrien Salat, Samuel Anklesaria, Mohammad Vazifeh, Carlo Ratti, and Geoffrey B. West. The universal visitation law of human mobility. *Nature*, 593(7860):522–527, 2021.

Filippo Simini, Marta C. González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *Nature*, 484(7392):96–100, 2012. doi: 10.1038/nature10856. URL https://doi.org/10.1038/nature10856.

Filippo Simini, Gianni Barlacchi, Massimilano Luca, and Luca Pappalardo. A deep gravity model for mobility flows generation. *Nature Communications*, 12(1):6576, 2021. doi: 10.1038/s41467-021-26752-4. URL https://doi.org/10.1038/s41467-021-26752-4.

Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. Modelling the scaling properties of human mobility. *Nature Physics*, 6(10):818–823, 2010a. doi: 10.1038/nphys1760. URL https://doi.org/10.1038/nphys1760.

Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010b. doi: 10.1126/science.1177170. URL https://doi.org/10.1126/science.1177170.

Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. Prediction of human emergency behavior and their mobility following large-scale disaster. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pp. 5–14, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450329569. doi: 10.1145/2623330.2623628. URL https://doi.org/10.1145/2623330.2623628.

Samuel A. Stouffer. Intervening opportunities: A theory relating mobility and distance. *American Sociological Review*, 5(6):845–867, 1940. doi: 10.2307/2084520.

Zhenyu Tao, Wei Xu, and Xiaohu You. Map2Traj: Street map piloted zero-shot trajectory generation method for wireless network optimization. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 9366–9374, 2025. doi: 10.24963/ijcai.2025/1041. URL https://www.ijcai.org/proceedings/2025/1041.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://papers.nips.cc/paper_files/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, Noboru Koshizuka, and Chuan Xiao. Large language models as urban residents: An llm agent framework for personal mobility generation. In A. Globerson, L. Mackey, D. Belgrave,

A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 124547–124574. Curran Associates, Inc., 2024a. doi: 10.52202/079017-3957. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/e142fd2b70f10db2543c64bca1417de8-Paper-Conference.pdf.

Jiawei Wang, Renhe Jiang, Chuang Yang, Zengqing Wu, Makoto Onizuka, Ryosuke Shibasaki, Noboru Koshizuka, and Chuan Xiao. Large language models as urban residents: An LLM agent framework for personal mobility generation. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124547–124574, 2024b. doi: 10.52202/079017-3957. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/e142fd2b70f10db2543c64bca1417de8-Abstract-Conference.html.

Jingyuan Wang, Yujing Lin, and Yudong Li. GTG: Generalizable trajectory generation model for urban mobility. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(1):834–842, 2025. doi: 10.1609/aaai.v39i1.32067. URL https://ojs.aaai.org/index.php/AAAI/article/view/32067.

Leye Wang, Xu Geng, Xiaojuan Ma, Feng Liu, and Qiang Yang. Cross-city transfer learning for deep spatio-temporal prediction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pp. 1893–1899, 2019. doi: 10.24963/ijcai.2019/262. URL https://www.ijcai.org/proceedings/2019/262.

Yu Wang, Tongya Zheng, Yuxuan Liang, Shunyu Liu, and Mingli Song. Cola: Cross-city mobility transformer for human trajectory simulation. In *Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 3509–3520. ACM, May 2024c. doi: 10.1145/3589334.3645469. URL http://dx.doi.org/10.1145/3589334.3645469.

Ying Wei, Yu Zheng, and Qiang Yang. Transfer knowledge between cities. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1905–1914. Association for Computing Machinery, 2016. doi: 10.1145/2939672.2939830. URL https://doi.org/10.1145/2939672.2939830.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. SMART: Scalable multi-agent real-time motion generation via next-token prediction. In *Advances in Neural Information Processing Systems*, 2024.

Dingqi Yang, Daqing Zhang, Longbiao Chen, and Bingqing Qu. NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs. *Journal of Network and Computer Applications*, 55:170–180, 2015a.

Dingqi Yang, Daqing Zhang, Vincent W. Zheng, and Zhiyong Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2015b.

Dingqi Yang, Daqing Zhang, and Bingqing Qu. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology*, 7(3):30:1–30:23, 2016.

Yuan Yuan, Yuheng Zhang, Jingtao Ding, and Yong Li. WorldMove, a global open data for human mobility. *Scientific Data*, 13(1):549, 2026. doi: 10.1038/s41597-026-06555-2. URL https://doi.org/10.1038/s41597-026-06555-2.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speech-Tokenizer: Unified speech tokenizer for speech language models. In *International Conference on Learning Representations*, pp. 31798–31818, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/86d1ab582afb247ccaa84bec4a7e24f7-Abstract-Conference.html.

Chuanxia Zheng and Andrea Vedaldi. Online clustered codebook. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22798–22807, 2023. doi: 10.1109/iccv51070.2023.02084. URL https://openaccess.thecvf.com/content/ICCV2023/html/Zheng_Online_Clustered_Codebook_ICCV_2023_paper.html.

Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: Concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):38:1–38:55, 2014. doi: 10.1145/2629592. URL https://doi.org/10.1145/2629592.

Zhiyong Zhou, Song Gao, Qianheng Zhang, Feng Zhang, and Zhenhong Du. Quantifying geographic domain shift to decouple the geospatial transferability of human mobility flow generation models. arXiv preprint arXiv:2608.21567, 2026. URL https://arxiv.org/abs/2608.21567.

George Kingsley Zipf. The P_1P_2/D hypothesis: On the intercity movement of persons. *American Sociological Review*, 11(6):677–686, 1946. doi: 10.2307/2087063. URL https://doi.org/10.2307/2087063.

A Appendix

A.1 Dataset Statistics

The detail of the dataset we used can be checked here. All the developments and benchmarks are based on Foursquare New York City from the NYC and Tokyo check-in dataset (released as `dataset_TSMC2014_NYC.txt`) (Yang et al., 2015b), Foursquare Paris extracted from the global-scale TIST2015 dataset (Yang et al., 2016; 2015a) and Gowalla California (Cho et al., 2011).

Table 4 summarizes the source data pools and target test cohorts for SHAPE. For each table row, we combine each user's daily sequences into one user history. The > 10 columns count daily sequences or user histories with more than 10 check-ins. The total counts also include shorter sequences. The > 10 columns count sequences with at least 11 check-ins and do not specify a data filter. Table 5 reports the dataset usage in the tokenizer training of Stage 2.

Table 4: Source data pools and target test cohorts for SHAPE.

City	Data subset	Check-ins	Users	Daily sequences	Daily sequences > 10	User histories > 10
NYC	Source pool	227,427	1,083	93,862	1,688	1,083
CA	Source pool	172,573	1,070	53,816	1,707	1,070
NYC	Test: CA→NYC	22,772	1,082	9,400	166	922
CA	Test: NYC→CA	25,743	4,087	8,870	225	550
Paris	Test: NYC/CA→Paris	9,731	2,296	5,938	20	205

* Source pools include training, validation, and test splits. CA source users have at least 50 valid check-ins and five active days. Target test counts include single-event days. NYC test counts are verified for seed 42. The CA test cohort covers only day lengths that occur in the NYC source input. The Paris cohort is shared by the two transfer directions.

Table 5: Source-city data for tokenizer training in Stage 2.

Transfer direction	Training users	Training-split check-ins	Check-ins per epoch
NYC→CA	864	181,336	181,336
CA→NYC / CA→Paris	854	136,998	124,658
NYC→Paris	864	181,336	Not verified

* NYC→CA uses complete training sequences. Each CA-source run retains the first 512 check-ins per user, or the complete sequence if shorter. NYC→Paris reuses an earlier tokenizer whose truncation setting could not be verified from the available checkpoint records.

A.2 Detail of SHAPE methodology

Algorithm 1 lists the full procedure of SHAPE; the three stages are detailed below.

Algorithm 1 SHAPE: zero-shot cross-city mobility generation

Require: $\{X_u^A\}_u,\ \mathcal{A}^A,\ \mathcal{A}^B,\ H^B$.
Ensure: $\{X_u^B\}_u$, with the source activity pattern (t_i, c_i) unchanged.

Stage 1: Heterogeneous Grid Cell Alignment

Train ϕ_{stage1} with $\mathcal{L}_{align}$ on $(\mathcal{A}^A, \mathcal{A}^B)$. ▷ Eqs. (2)–(4)
$(h^A, h^B) \leftarrow F_1(\mathcal{A}^A, \mathcal{A}^B)$.

Stage 2: Activity Prototype Sequence Extraction

Train Stage 2 on $(\{X_u^A\}_u, h^A)$ with $\mathcal{L}_{stage2}$. ▷ Eq. (8)
for each trajectory X_u^A of A **do**
  Compute $\Delta_i = t_i - t_{i-1}$ and normalized jump length δ_i.
  $v_i \leftarrow f_{\text{time}}(t_i, \Delta_i) + f_{\text{poi}}(c_i) + f_{\text{move}}(\delta_i, \Delta_i) + f_{\text{grid}}(h^A_{g_i})$.
  $(z_1, \ldots, z_{T_u}) \leftarrow \phi_{stage2}(v_1, \ldots, v_{T_u})$; $r_i^{(0)} \leftarrow z_i$ for all i.
  for $l = 0, 1, 2$ **do** ▷ RFSQ
    $q_i^{(l)} \leftarrow \eta_l Q_l(r_i^{(l)})$; $\quad r_i^{(l+1)} \leftarrow r_i^{(l)} - \text{sg}(q_i^{(l)})$. ▷ Record $k_i^{(l)}$, for all i
  end for
  $k_i \leftarrow (k_i^{(0)}, k_i^{(1)}, k_i^{(2)})$; $\quad C_u \leftarrow (k_1, \ldots, k_{T_u})$. ▷ $F_2(X_u^A, h^A)$
  $\bar{z}_i \leftarrow \sum_{l=0}^{2} q_i^{(l)}$; $\quad \hat{h}_i \leftarrow \text{DECODE}(\bar{z}_i)$ for all i. ▷ decoded place features
  Store C_u with $(\hat{h}_i, t_i, c_i)_{i=1}^{T_u}$.
end for

Stage 3: Prototype-conditioned Grid Cell Selection

$w_g \leftarrow m_g/\bar{m}$; $\quad R(o, g) \leftarrow \sum_{g':\, d(o,g')<d(o,g)} w_{g'}$ in each city.
Fit Q_φ, $p_k(b)$, $\pi_k(c)$, prototype category profiles and γ on A; freeze Q_φ.
Set d_{min} to the smallest positive jump length in A.
for each activity prototype sequence C_u **do**
  Allocate a home cell using H^B; $n_1(g) \leftarrow 0$ for all target cells g.
  for $i = 1, \ldots, T_u$ **do**
    if first activity of a day **then** set $\hat{g}_{i-1}^B$ to the home cell.
    $q_i \leftarrow Q_\varphi(\hat{h}_i)$; $\quad b_i \sim \text{Categorical}(p_{k_i^{(0)}})$.
    $\mathcal{E}_i \leftarrow \{g : R(\hat{g}_{i-1}^B, g) \in b_i\}$. ▷ opportunity-rank ring
    while $\mathcal{E}_i = \varnothing$ **do** grow the ring into adjacent bins and recompute $\mathcal{E}_i$.
    for each $g \in \mathcal{E}_i$ **do** ▷ Candidate scoring, Eq. (12)
      $S_i(g) = \sigma \cos(q_i, a_g^B) + \frac{1}{\tau} \log \frac{\pi_{k_i^{(0)}}(c_g)}{|\mathcal{E}_i \cap c_g|} + \log \frac{A_{k_i^{(0)}}(g)^\rho}{\overline{A^\rho}(c_g)} - \gamma \log \frac{d(\hat{g}_{i-1}^B, g)}{d_{min}} + \beta \log(1 + n_i(g))$.
    end for
    Sample $\hat{g}_i^B$ from $\text{softmax}_{g \in \mathcal{E}_i} S_i(g)$.
    $n_{i+1}(\cdot) \leftarrow n_i(\cdot)$; $\quad n_{i+1}(\hat{g}_i^B) \leftarrow n_i(\hat{g}_i^B) + 1$.
  end for
  $X_u^B \leftarrow ((t_i, \hat{g}_i^B, c_i))_{i=1}^{T_u}$. ▷ $F_3(C_u, \mathcal{A}^B, H^B)$
end for
return $\{X_u^B\}_u$.

A.2.1 Stage 1: Heterogeneous Grid Cell Alignment

Shared encoder. The grid cells of both cities pass through one encoder with shared weights. The geographical attributes of a city are $\{(a_g, p_g)\}_{g \in \mathcal{G}}$, where $\mathcal{G}$ is its set of grid cells, $a_g \in \mathbb{R}^{20}$ stacks the POI density level, the POI composition and the transit information. $p_g \in \mathbb{Z}^2$ is the row and column index of g within its spatial extent $\mathcal{N}_g$, which enters the encoder through a sinusoidal positional encoding. The extent of a city is divided into a 32×32-cell block. Like a convolution kernel, the encoder represents a grid cell from its own attributes and those of the cells around it, so the block is padded with four cells on every side, and the resulting 40×40 area is the spatial extent $\mathcal{N}_g$. The shared encoder ϕ_{stage1} is a Transformer encoder that outputs a 64-dimensional vector:

$$h_g^c = \text{norm}\big(\phi_{stage1}(a_g^c, p_g^c; \mathcal{N}_g^c)\big) \in \mathbb{R}^{64}, \qquad c \in \{A, B\}. \tag{2}$$

Applied to every grid cell of a city, the encoder yields the representation of each city, h^A or h^B in Eq. equation 1. To make the representation of two cities comparable, the encoder lets the representation of a cell depend on what surrounds it and not on the city it lies in. It measures the relative positions within the spatial extent. We also keep the size of the area $\mathcal{N}_g$ the same everywhere in

both cities, so a cell in a sparsely covered suburb is described over the same range as one in a dense centre.

Alignment by contrastive learning. The shared encoder only models the geographical attributes of both cities in one feature space. We also require contrastive learning to align the cells of the two cities by urban function, so that a cell of the source city and the target-city cells of similar function lie close together in the embedding space. Let π_i be the POI composition of cell i, the share of each of the 13 POI categories among its POIs, and let $\mathrm{JSD}(\pi_i, \pi_j)$ be the Jensen–Shannon divergence between two compositions (Lin, 1991). For a cell i in city A , a softmax over its divergences from the cells in city B in the batch gives the target weights, which are larger for cells with a more similar composition:

$$w_{ij} = \frac{\exp\big(-\mathrm{JSD}(\pi_i, \pi_j)/\tau_p\big)}{\sum_{k\in\mathcal{B}_i} \exp\big(-\mathrm{JSD}(\pi_i, \pi_k)/\tau_p\big)}, \qquad j \in \mathcal{B}_i, \tag{3}$$

where i indexes a cell of $\mathcal{A}$, $\mathcal{B}_i$ is the set of cells of B in the current batch, and $\tau_p > 0$ controls how sharply the weight concentrates on the closest compositions. The embeddings are trained so that their similarities reproduce w_{ij}:

$$P_{ij} = \frac{\exp\big((h_i^A)^\top h_j^B/\tau_e\big)}{\sum_{k\in\mathcal{K}} \exp\big((h_i^A)^\top h_k^B/\tau_e\big)}, \qquad \mathcal{L}_{A\to B} = -\frac{1}{N}\sum_i \sum_{j\in\mathcal{B}_i} w_{ij} \log P_{ij}, \tag{4}$$

where $\tau_e > 0$ is the temperature of the similarity softmax and N is the number of cells of A in the batch. Since h is ℓ_2-normalized, $(h_i^A)^\top h_j^B$ is a cosine similarity. The target w_{ij} and the similarity softmax P_{ij} are computed over different groups of cells of B on purpose, so that a cell with no target weight still acts as a negative and is pushed away from h_i^A. w_{ij} puts weight only on the cells in $\mathcal{B}_i$. The softmax of P_{ij} is normalized over a larger set $\mathcal{K} \supseteq \mathcal{B}_i$, all cells of B in the batch plus extra sampled negatives. The objective is symmetric, defined by $\mathcal{L}_{\text{align}} = \frac{1}{2}(\mathcal{L}_{A\to B} + \mathcal{L}_{B\to A})$. Because w_{ij} spreads its weight over all cells of B with a similar composition instead of singling out one of them, the alignment does not lock a cell of A to a fixed counterpart in B. Several cells of B may serve the same function, and Stage 3 decides which of them an activity is placed in.

A.2.2 Stage 2: Activity Prototype Sequence Extraction

Activity encoding. The jump length δ_i is normalized by the diagonal of the spatial extent of A. Four branches embed the inputs into a common d-dimensional space, one each for the time and interval, the activity type, the jump length, and the representation of g_i^A from h^A. Their outputs are summed.

$$v_i = f_{\text{time}}(t_i, \Delta_i) + f_{\text{poi}}(c_i) + f_{\text{move}}(\delta_i, \Delta_i) + f_{\text{grid}}(h^A_{g_i}). \tag{5}$$

They correspond to the four input boxes of Figure 2. The Transformer encoder then encodes the whole sequence:

$$(z_1, \ldots, z_{T_u}) = \phi_{stage2}(v_1, \ldots, v_{T_u}), \tag{6}$$

where $v_i \in \mathbb{R}^{256}$ is the embedding of activity i on its own and $z_i \in \mathbb{R}^{256}$ is its embedding in the context of the whole trajectory. Every component of v_i is city-independent. The time of day is a relative variable. The activity type comes from the 13 classes shared by both cities, the jump length is normalized by the spatial extent of A, and the representation of the grid cell from h^A is already aligned across cities by Stage 1. The codebook learned on A is transferable. The activity prototypes capture regularities shared across residents.

3-Layer Residual Finite Scalar Quantization. This component discretizes the representations z_i with three layers of finite scalar quantization (Mentzer et al., 2024), each applied to the residual left by the layers before it. Each layer first projects its 256-dimensional input to three scalars with a learned linear map. The scalars are normalized and then passed through a hyperbolic tangent, which bounds them to $[-1, 1]$. Each scalar is then rounded and clipped to one of the four values $\{-1, -\frac{1}{3}, \frac{1}{3}, 1\}$. The value choice follows from the choice of $L = 4$ levels. FSQ spaces the levels evenly over the range $[-1, 1]$ of the hyperbolic tangent, at $-1 + 2k/(L-1)$ for $k = 0, \ldots, L-1$, so the outer two sit at the bounds and the spacing is $2/3$. With three scalars at four levels each, a layer has $4^3 = 64$ codes, and each activity receives a triple of codes $k_i = (k_i^{(0)}, k_i^{(1)}, k_i^{(2)})$, one

from each layer. Gradients pass through the rounding with the straight-through estimator, as in FSQ. Implementation details are given in Appendix A.2.

The three layers are chained on residuals. Let Q_l denote the quantization of layer l described above and η_l a learnable output scale, initialized to one. Layer l quantizes what the layers before it have left unexplained:

$$r_i^{(0)} = z_i, \qquad q_i^{(l)} = \eta_l \, Q_l\big(r_i^{(l)}\big), \qquad r_i^{(l+1)} = r_i^{(l)} - \mathrm{sg}\big(q_i^{(l)}\big), \qquad l = 0, 1, 2, \tag{7}$$

where sg is the stop-gradient operator. The three layers are trained together through the sum $\bar{z}_i = \sum_l q_i^{(l)}$, which the decoder receives. The stop-gradient keeps the residual passed to layer $l+1$ free of gradient from layer l, so each layer is optimized for its own contribution to that sum. Each layer l gives activity i one code $k_i^{(l)}$, and the three codes together are its prototype k_i. The recursion does not by itself force later layers to carry finer or independent information. That division is imposed by $\mathcal{L}_{\mathrm{sem}}$, which supervises each layer with a different group of activity features, and Table 6 reports how much each layer adds. The decoder maps $\bar{z}_i$ to a 64-dimensional vector $\widehat{h}_i$. It is trained to reconstruct $h^A_{g_i}$, the representation of the grid cell where activity i took place ($\mathcal{L}_{\mathrm{rec}}$). Therefore, each prototype encodes the place of its activity in the same embedding space as the grid cell representations, and Stage 3 maps $\widehat{h}_i$ to an attribute query to select a matching cell in the target city.

Stage 2 is trained end-to-end. Its loss adds to the reconstruction error five terms that shape the codes:

$$\mathcal{L}_{\mathrm{stage2}} = \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{div}}\mathcal{L}_{\mathrm{div}} + \alpha_e\mathcal{L}_{\mathrm{temp}} + \lambda_{\mathrm{sem}}\mathcal{L}_{\mathrm{sem}} + \lambda_{\mathrm{jump}}\mathcal{L}_{\mathrm{jump}} + \lambda_{\mathrm{poi}}\mathcal{L}_{\mathrm{poi}}, \tag{8}$$

where $\mathcal{L}_{\mathrm{rec}}$ is the squared error between $\widehat{h}_i$ and the representation $h^A_{g_i}$ of the grid cell that activity i visits; $\mathcal{L}_{\mathrm{div}}$ is the standard entropy-difference objective on the soft code assignments; $\mathcal{L}_{\mathrm{temp}}$ stops consecutive activities from collapsing onto one prototype, which would flatten a trajectory into repeats of the same code, by penalizing neighbours that share a code or have near-identical embeddings and rewarding varied code transitions; $\mathcal{L}_{\mathrm{sem}}$ ties the input of each quantizer layer to one group of activity features, which is what makes a code readable; $\mathcal{L}_{\mathrm{jump}}$ and $\mathcal{L}_{\mathrm{poi}}$ are cross-entropy losses that predict the jump-length bin and the activity type of an activity from its quantized vector. The jump length is taken relative to the spatial extent of A to exclude the city-specific features, and cut into bins. The weights $\lambda_\bullet$ are fixed, α_e is warmed up over the first epochs, and their values are listed in the reproducibility statement.

A.2.3 Stage 3: Prototype-conditioned Grid Cell Selection

Opportunity rank and attribute query. Physical distance is not comparable between cities, because the same number of kilometres covers different amounts of opportunity in different places. The number of opportunities between origin and destination is comparable, which is a regularity that holds across cities (Stouffer, 1940; Noulas et al., 2012). We therefore measure a step by the opportunity it passes over, with opportunity counted by POIs. Each grid cell g carries an opportunity weight:

$$w_g = \frac{m_g}{\bar{m}}, \tag{9}$$

where m_g is the number of POIs in g and $\bar{m}$ is the mean POI count over the grid cells of the same city, so w_g is the opportunity of g in units of an average cell. For an origin o and a candidate g, the opportunity-weighted rank is

$$R(o, g) = \sum_{g' \colon d(o,g') < d(o,g)} w_{g'}, \tag{10}$$

where $d(\cdot, \cdot)$ is the distance in kilometres between grid-cell centres. $R(o, g)$ is the opportunity that lies strictly closer to o than g does, in units of an average cell. Distance enters only as the order in which cells are counted, so one kilometre across a dense centre passes far more opportunity than one kilometre across a park. Both cities use this definition with their own $\bar{m}$, so equal values of R mean equal opportunity passed over.

The attribute query is the second measure. A query head $Q_\varphi : \mathbb{R}^{64} \to \mathbb{R}^{20}$, trained on A and then frozen, maps the decoded place features $\widehat{h}_i$ to a query $q_i = Q_\varphi(\widehat{h}_i)$ in the attribute space of the static map. The match between activity i and a cell g is the cosine similarity between q_i and the attribute vector a^B_g of g. The rank measures how far a step goes and the query what kind of cell it needs.

Ring of opportunity rank. We estimate a distribution $p_k(b)$ for each layer-0 code k over four rank bins from the trajectories of A. For activity i, a bin is sampled from the distribution of its layer-0 code, and the cells whose rank from the previous cell lies in that bin are eligible:

$$b_i \sim \text{Categorical}\big(p_{k_i^{(0)}}\big), \qquad \mathcal{E}_i = \big\{\, g :\ R\big(\widehat{g}^B_{i-1}, g\big) \in b_i \,\big\}. \tag{11}$$

Here $\widehat{g}^B_{i-1}$ is the cell chosen for the previous activity, and for the first activity of a day it is the home cell. Cells outside $\mathcal{E}_i$ are excluded from this step. If $\mathcal{E}_i$ is empty, the ring grows into the adjacent bins until it contains a cell.

Candidate scoring. Every cell in $\mathcal{E}_i$ receives a score with five terms,

$$S_i(g) = \underbrace{\sigma \cos\big(q_i, a^B_g\big)}_{\text{attribute term}} + \underbrace{\frac{1}{\tau} \log \frac{\pi_{k_i^{(0)}}(c_g)}{|\mathcal{E}_i \cap c_g|}}_{\text{sub-ring term}} + \underbrace{\log \frac{A_{k_i^{(0)}}(g)^\rho}{\overline{A^\rho}(c_g)}}_{\text{attraction term}} - \underbrace{\gamma \log \frac{d\big(\widehat{g}^B_{i-1}, g\big)}{d_{\min}}}_{\text{distance term}} + \underbrace{\beta \log\big(1 + n_i(g)\big)}_{\text{revisit term}}, \tag{12}$$

The five terms are logarithms, so the softmax of S_i multiplies five preferences. The attribute term is the cosine similarity between the query q_i and the attribute vector a^B_g, weighted by $\sigma > 0$. The sub-ring term spreads the probability of the ring over its finer rank sub-bins. c_g is the sub-bin that contains g, $\pi_k(c)$ is the probability of sub-bin c under code k, estimated on A, and dividing by the number of eligible cells in c_g turns the probability of a sub-bin into that of one cell in it. $\tau > 0$ sets the sharpness. The attraction term prefers cells whose POI composition suits the activity. $A_k(g)$ scores the POI composition of g against the category profile of prototype k, estimated on A, $\rho > 0$ sets its strength, and dividing by $\overline{A^\rho}(c_g)$, the mean over the eligible cells of the same sub-bin, keeps the term from changing the probability of the sub-bin. The distance term penalizes long steps, with $\gamma > 0$ fitted on A and $d_{\min}$ the smallest positive jump length in A. The revisit term favours cells the trajectory has already visited, where $n_i(g)$ counts those visits and $\beta > 0$ sets its strength.

A softmax over $\mathcal{E}_i$ turns S_i into a distribution and one cell is sampled from it. That cell is recorded as $\widehat{g}^B_i$, becomes the origin of the next step, and updates the revisit counts. Repeating this over C_u gives the trajectory X^B_u.

A.3 Choice of the quantizer

The standard way to learn a discrete codebook is vector quantization with exponential moving average updates (EMA-VQ), as in VQ-VAE (van den Oord et al., 2017; Razavi et al., 2019). Figure 5 shows why Stage 2 uses finite scalar quantization (FSQ) instead. In EMA-VQ a code is updated only when it is selected. Codes that lose out early stop being updated, fall out of reach of the encoder, and are never selected again. On our data, this collapse is severe. After epoch 15, EMA-VQ uses 30% to 58% of the $L0$ codebook (Figure 5e) and its aggregate perplexity stays below 5 of 64 (f), so almost all activities fall on fewer than five codes, while its training loss climbs and oscillates (d). FSQ has no learned codebook. Its codes are fixed combinations of the four levels of each scalar, and any of them is reached by rounding the encoder output, so no code can fall out of use; $\mathcal{L}_{\text{div}}$ further spreads the assignments. As a result the loss decreases monotonically (a), more than 80% of the codes at each of the three layers stay in use after epoch 30 (b), and the aggregate perplexity exceeds 40 of 64 (c). Stage 2 needs a codebook whose codes all remain in use, because Stage 3 conditions on the $L0$ code and residents differ only in their prototype sequences.

Figure 6 projects the $L0$ latents of both quantizers onto their first two principal components for three controlled pairs of activities that differ in one attribute only, separately time period (food/social, local, 12–18 against 18–24), jump length (transport, 06–12, local against long range) and activity type (stationary, 12–18, shop against outdoor recreation). Under EMA-VQ the two members of every pair overlap almost completely. The silhouette scores lie between 0.011 and 0.068. Under FSQ the two activity types form two disjoint clusters (10-nearest-neighbour purity 1.000, silhouette 0.748), and the two jump lengths are almost as well separated (0.953, 0.565). Both attributes are therefore resolved at $L0$, the layer on which the ring of Stage 3 conditions. The two time periods remain mixed under FSQ as well (0.685, 0.100).

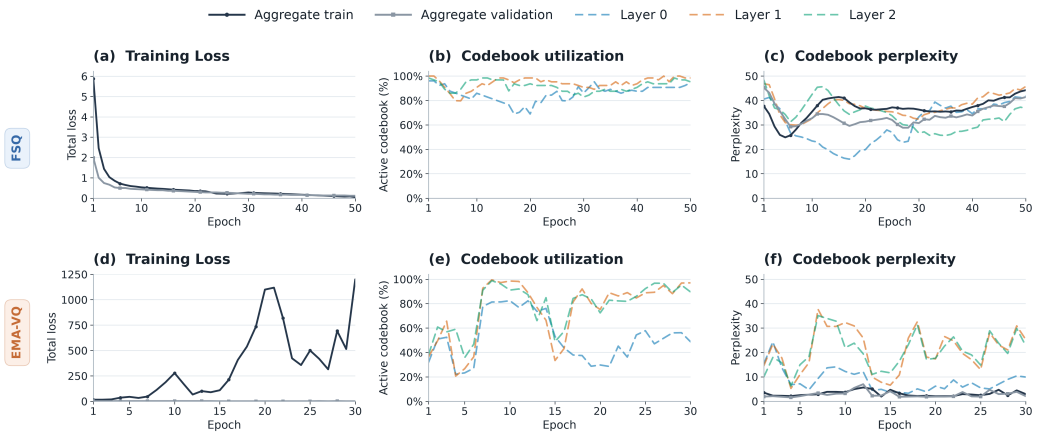


Figure 5: Training loss and codebook statistics for FSQ and EMA-VQ. The top row (a–c) shows FSQ, and the bottom row (d–f) shows EMA-VQ. From left to right, the columns show total loss, codebook utilization (percentage of active code), and codebook perplexity. Perplexity expresses the effective number of codes based on their assignment frequencies.

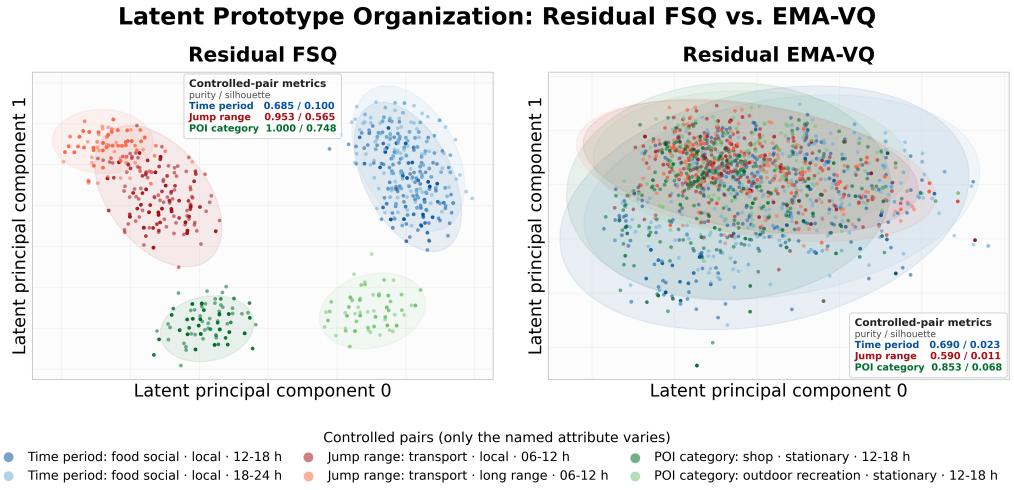


Figure 6: $L0$ latents of residual FSQ and residual EMA-VQ, projected onto their first two principal components (per panel). Activities come from three controlled pairs that differ in one attribute only; the two shades of a colour are the two values of that attribute, and each ellipse covers one group. The inset gives the 10-nearest-neighbour purity and the silhouette score of each pair.

Table 6: Incremental information from successive codebook layers. Rows give the layers available, from the first layer alone to all three layers; columns give three attributes of an activity: its activity type, its jump-length bin and its time period. Entries are in bits; a negative value means the layer added in that row makes the attribute more predictable.

Layers available	Activity type	Jump-length bin	Time period
L0	-2.106 bits	-0.661 bits	-0.069 bits
L0+L1	-0.835 bits	-0.131 bits	-0.078 bits
L0+L1+L2	-0.020 bits	-0.068 bits	-0.066 bits

A.4 EXAMPLE OF CODEBOOK INTERPRETATION

To help readers better understand the mechanism of hierarchical codebook, here two specific examples are mentioned, the information refinement process of which is illustrated in the Figure 7.

For example, $L0$ code 39 has a mixed activity-type distribution, with 45% transport and additional mass on professional, arts, and outdoor categories. Conditioning on $L1$ separates this mixture into prefixes $(39, 8)$, $(39, 44)$, and $(39, 32)$, whose the corresponding observed POI shares are 100% transport, 100% outdoor, and 100% arts, respectively. Similarly, prefix $(62, 38)$ mixes time periods and jump-length bins. Adding $L2$ yields $(62, 38, 42)$, dominated by 18:00–24:00, and $(62, 38, 23)$, dominated by 12:00–18:00. The stationary shares range from 0.32 to 0.61 across the refined prefixes. By optimizing these refinements under a shared objective without layerwise semantic injection, the residual layers establish a hierarchy that naturally supports a coarse-to-fine interpretation of the three-layer residual FSQ representation.

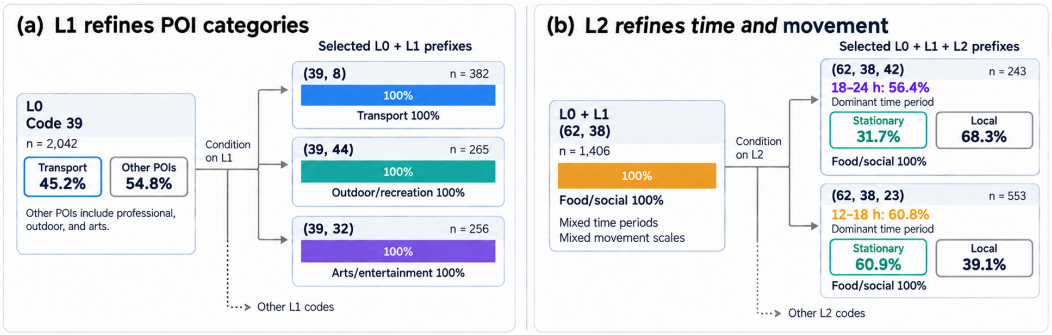


Figure 7: Conditional refinement in residual FSQ on NYC held-out activities. Each panel reads from left to right. Arrows divide activities by the code at the next layer. (a) L1 groups within L0 code 39 are shown by POI category. (b) L2 groups within prefix $(62, 38)$ are shown by dominant time period and jump-length bin. All displayed L2 groups contain only Food/social activities. Tuples list code IDs from L0 onward. Percentages are proportions within each group, and n is the activity count. Dotted branches indicate groups not shown.