A CONTINUOUS VIEW OF URBAN SPACE: NEURAL DIRICHLET RANDOM MEASURES FOR URBAN SPATIAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Urban regions aggregate different information depending on specific partitions, but existing data-driven urban region representations are typically learned on a fixed spatial partition, limiting their applicability. To address this limitation, we propose a neural Dirichlet random measure: instead of learning representations for regions in a fixed partition, we learn a continuous intensity field over the entire spatial domain. Thus, integrating the field over any region gives its quantitative representation while ensuring conservation. Moreover, this continuous representation can also be learned directly from observations on different urban partitions, leading to a more flexible and generalizable way. We conduct experiments from both perspectives: as a predictor on arbitrary partitions, and as a learner of partition-agnostic urban spatial embeddings. As a predictor, experiments on ten major U.S. cities with widely used urban spatial embeddings show that our method guarantees quantitative conservation and also yields generally better predictive quality than conventional direct prediction. As a learner, cross-city transfer validation shows that our method learns partition-agnostic embeddings directly from observations on different partitions, and that these embeddings transfer to unseen cities and generally outperform those trained in the conventional fixed-partition way. These indicate that it is feasible to remove the dependence on fixed spatial partitions, which provides a new direction for urban representation learning.

1 INTRODUCTION

Urban representation learning aims to learn representations of urban space from multi-source data related to human activities, including satellite imagery (Xi et al., 2022; Jean et al., 2019; Cong et al., 2022), points of interest (POIs) (Xi et al., 2022; Zhang et al., 2021; Huang et al., 2023), and human mobility (Wang & Li, 2017; Yao et al., 2018; Liu et al., 2020), etc. It distills the key information and compresses, abstracts, and embeds it into a low-dimensional, dense representation space, so that the representations can support diverse downstream tasks simply by attaching task-specific models on top of them. Such representations have been used to estimate socioeconomic indicators (Han et al., 2020a; Park et al., 2022; Han et al., 2020b; Ahn et al., 2023; Rong et al., 2026; Li et al., 2022; Liu et al., 2023), poverty (Jean et al., 2016; Yeh et al., 2020), and health-related indicators (Han et al., 2023; Maharana & Nsoesie, 2018; Suel et al., 2019), as well as to classify land use (Zhang et al., 2021; Li et al., 2023), among many other applications. These tasks are crucial for monitoring and advancing sustainable development (Xi et al., 2023; Burke et al., 2021; Yeh et al., 2021). The value of urban representations lies largely in their reusability: once learned, a representation can be directly adopted by other researchers for their own tasks, without collecting new data or training new models (Rolf et al., 2021; Hao et al., 2025; Zhang et al., 2025; Mai et al., 2023b). Such representations can be regarded as foundation models for the urban domain (Zhang et al., 2025; Mai et al., 2023b; Bommasani et al., 2021; Mai et al., 2023a; Zhang et al., 2024). Such general-purpose geospatial representation models, e.g., RemoteCLIP (Liu et al., 2024), SatCLIP (Klemmer et al., 2025) and AlphaEarth (Brown et al., 2025), have also emerged along this line.

However, these works share two fundamental limitations. First, they require a predefined spatial partition and learn embeddings for each of the corresponding regions. Such partitions consist of either irregular polygons, such as administrative districts and census units (Zhang et al., 2021; Huang

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.

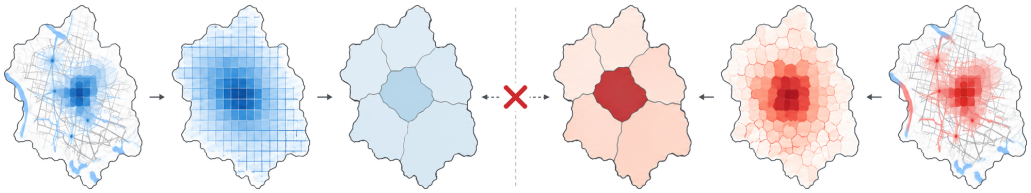


Figure 1: An illustration of quantity non-conservation in existing urban representation methods.

et al., 2023; Li et al., 2023), or regular grid cells (Ahn et al., 2023; Sun et al., 2025). The learned representations are therefore bound to a specific partition, and switching to a different partition requires retraining the model. In geography, this dependence of aggregated statistics on the partition is known as the modifiable areal unit problem (Wong, 2004). Second, they do not conserve quantities, i.e., the quantity predicted for a region is not guaranteed to equal the sum of those predicted for its subregions. Although several works claim to learn continuous representations (Mai et al., 2023b; Klemmer et al., 2025; Brown et al., 2025), these models have an intrinsic minimum resolution at which they are trained, and the quantity of a region must be obtained by pooling the representations of these minimal units and passing the result through a downstream model. Consequently, even when representations are aggregated over larger spatial units for additional training (Han et al., 2020a; Park et al., 2022; Sun et al., 2025), quantity conservation still cannot be guaranteed, as shown in Figure 1. Moreover, their predictions are tightly coupled to the value range of the training data: once the quantity of a region falls outside the range seen in training, e.g., for regions much larger or smaller than the training units, the predictions are no longer reliable (Xu et al., 2020). These two limitations confine the reuse of urban representations to a specific partition and value range.

In this work, we propose the *neural Dirichlet random measure*, which learns a continuous representation of urban space, to overcome the two limitations above. Instead of learning a set of embeddings for each region of a predefined partition, we learn a continuous intensity field over the entire urban space, and the quantity of any region equals the integral of the field over that region. Since integration is inherently additive, the quantity of a region always equals the sum of the quantities of its subregions, so quantity conservation is guaranteed by the model structure itself. Meanwhile, since the field describes the intensity per unit area, the predicted quantity scales naturally with the size of the region and is not confined to the value range of the training data. We further take this integral as the base measure of a Dirichlet random measure (Ferguson, 1973). Given the total of a region, the counts over any partition of the region follow a Dirichlet–multinomial distribution (Mosimann, 1962). Every sample sums exactly to the known total, the predictive distributions on different partitions are mutually consistent, and the uncertainty is naturally quantified. This continuous representation brings two benefits. As a predictor, the trained field can be directly integrated over any partition to produce predictions that conserve the totals, without retraining. As a learner, observations on any partition can be expressed as integrals of the same field, so the model can be trained directly on data from different partitions to learn partition-agnostic representations. As a result, the neural Dirichlet random measure removes the dependence on fixed partitions and guarantees quantity conservation at any scale, making representations truly reusable over arbitrary regions.

We evaluate the neural Dirichlet random measure as both a predictor and a learner in ten major cities in the U.S. Conditioned on the totals of census tracts, we predict the population and POI counts of the census blocks within them, covering 223,617 blocks, with widely used geospatial representations, i.e., either AlphaEarth (Brown et al., 2025) or GHSL (Pesaresi & Politis, 2023), as inputs. As a predictor, our method achieves quantity conservation. In contrast, the conventional practice of pooling and prediction fails to guarantee such consistency, with a median deviation of 86% between the direct prediction for a region and the sum of its subregions. Furthermore, the prediction error decreases in all 10 cities, by 36–41% for population and about 17% for POI on average compared with direct prediction. In addition, on regions that are finer than the training units and unaligned with them, evaluated against exact point-level ground truth, our model also produces predictions that conserve the totals and remain consistent across scales. As a learner, we pretrain an encoder on AlphaEarth embeddings in nine cities with data defined on different partitions, and transfer the learned representations to the test city. For population, these representations outperform those pretrained in the conventional pooling manner in all cross validations, reducing the prediction error by about 15%, and for POIs they are better in most cities. These results show that it is feasible to learn continuous, quantity-conserving urban representations free from fixed partitions, which opens a new direction for urban representation learning.

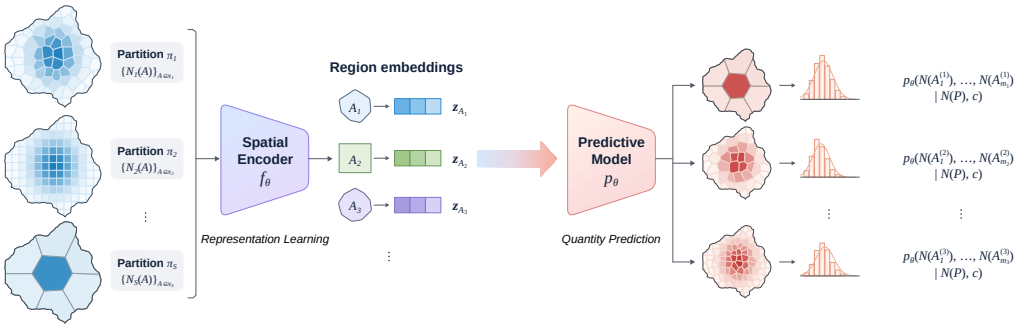


Figure 2: An illustration of the relationship between representation learning (PROBLEM 2) and urban prediction (PROBLEM 1).

Our main contributions are summarized as follows.

- We represent urban space by an intensity field whose integral gives the quantity of any region, free from fixed partitions and conserving quantities.
- We propose the neural Dirichlet random measure, which yields count distributions that exactly conserve the total and stay consistent over any partition. It can serve as both predictor and learner across different partitions.
- We conduct experiments on ten major U.S. cities, showing that our method conserves quantities exactly and reduces the prediction error by 17–41% as a predictor and by about 15% as an embedding learner.

2 Preliminaries

2.1 Notations and Problem Definitions

Definition 1 (Urban Space and Partitions)**.** The urban space of a city is a bounded domain $\Omega \subset \mathbb{R}^2$, and a region is a measurable subset $A \subseteq \Omega$. A partition of a region $P \subseteq \Omega$ is a finite set of pairwise disjoint regions $\pi = \{A_1, \ldots, A_m\}$ whose union is P, i.e., $\bigcup_{i=1}^{m} A_i = P$. Taking $P = \Omega$ gives a partition of the whole city.

Definition 2 (Spatial Context)**.** Each location $x \in \Omega$ is associated with a context vector $c(x) \in \mathbb{R}^d$ that describes its local environment and urban functions. The context is defined at every location and does not depend on any partition.

Definition 3 (Region Embedding)**.** A region embedding maps a region $A \subseteq \Omega$ to a vector $\mathbf{z}_A \in \mathbb{R}^k$ that summarizes its spatial context. A downstream model g then predicts a quantity of the region, such as its population count $N(A)$, as $\hat{N}(A) = g(\mathbf{z}_A)$.

Problem 1. ***Urban Prediction.*** Given the spatial context c and the known count $N(P)$ of a region P, the task is to predict the counts over any partition $\pi = \{A_1, \ldots, A_m\}$ of P, including partitions unseen during training, as a predictive distribution

$$p_\theta\big(N(A_1), \ldots, N(A_m) \mid N(P), c\big). \tag{1}$$

Problem 2. ***Representation Learning.*** Given counts of different quantities observed on different partitions, $\{(\pi_s, \{N_s(A)\}_{A \in \pi_s})\}_{s=1}^{S}$, the task is to learn an encoder f_θ of the spatial context, from which the region embedding $\mathbf{z}_A$ of any region A can be obtained.

2.2 Existing Urban Spatial Learning

Existing studies on urban spatial learning fall roughly into two successive categories, as shown in Figure 2. The first is urban representation learning, which typically learns reusable representations from multi-source data through pretraining and self-supervised learning (see Appendix A.1 for a detailed introduction). The second is downstream task modeling, which designs predictive models for specific tasks using these representations or raw data (see Appendix A.2 for a detailed introduction).

However, existing representations are either learned for the regions of a fixed partition or pooled within each region, so they are inconsistent across partitions. Downstream models built on them

usually give a point estimate for each region separately, and the prediction for a region need not equal the sum over its subregions. This motivates us to model urban quantities as a continuous field over the entire space, whose integral gives the quantity of any region, in the spirit of change-of-support models in spatial statistics (Gotway & Young, 2002). A single representation can then give conserving predictions on any partition and learn from data on different partitions.

3 METHODOLOGY

Our model connects three components in sequence, namely a continuous intensity field, an integral readout, and a Dirichlet random measure (DRM),

$$\underbrace{c(x) \xrightarrow{f_\theta} h(x) \xrightarrow{\phi} a(x)}_{\text{continuous intensity field}} \underbrace{\xrightarrow{\int_A} \alpha(A)}_{\text{integral readout}} \underbrace{\xrightarrow{\text{DRM}} p_\theta\big(N(A_1), \ldots, N(A_m) \mid N(P), c\big)}_{\text{Dirichlet random measure}}. \tag{2}$$

The continuous intensity field maps the context at each location to a nonnegative intensity (Section 3.1). The integral readout turns the field into the quantity of any region (Section 3.2). The Dirichlet random measure turns these quantities into count distributions over any partition of a region with a known total (Section 3.3). The same components are trained as a predictor for Problem 1 and as a learner for Problem 2 (Section 3.4).

3.1 CONTINUOUS INTENSITY FIELD

The field takes the spatial context $c(x)$ in Definition 2 as input, such as AlphaEarth embeddings (Brown et al., 2025) or GHSL built-up layers (Pesaresi & Politis, 2023). A shared encoder f_θ maps the context of each location to a location representation,

$$h(x) = f_\theta\big(c(x)\big) \in \mathbb{R}^k. \tag{3}$$

The encoder acts on each location independently and involves no region or partition. It is therefore partition-free, and it is the encoder to be learned in Problem 2. For each quantity q, such as population or POI counts, an output head ϕ_q maps the representation to a nonnegative intensity,

$$a_q(x) = \mathrm{softplus}\big(\phi_q(h(x))\big) + \varepsilon_0, \tag{4}$$

where the softplus keeps the intensity nonnegative, and the small background intensity $\varepsilon_0 > 0$ keeps the quantity of every region positive, as required by the Dirichlet random measure in Section 3.3. The intensity $a_q(x)$ is the quantity per unit area at location x, as in spatial point processes (Daley & Vere-Jones, 2003). It does not depend on the size of any region. Only its integral over a region grows with the area, as described in Section 3.2. The encoder can be built directly on existing embedding fields such as AlphaEarth. Rather than replacing them, it learns an integrable intensity field on top of them.

3.2 INTEGRAL READOUT

The integral readout defines the quantity of any region $A \subseteq \Omega$ as the integral of the intensity over the region,

$$\alpha_q(A) = \int_A a_q(x)\, dx. \tag{5}$$

By the linearity of integration, for any partition $\{A_1, \ldots, A_m\}$ of A,

$$\alpha_q(A) = \sum_{i=1}^{m} \alpha_q(A_i). \tag{6}$$

This additivity holds for every partition, nested or not, at any scale. The quantity of a region is thus conserved across partitions. Section 3.3 further extends this conservation from quantities to counts.

The key difference from existing methods lies in the order of pooling and readout. Existing methods (Han et al., 2020a; Park et al., 2022; Sun et al., 2025) first pool the context within a region and

then read out, while our method first reads out at each location and then integrates,

$$\underbrace{g\Big(\frac{1}{|A|}\int_A c(x)\,dx\Big)}_{\text{pool, then read out}} \qquad \text{vs.} \qquad \underbrace{\int_A a_q(x)\,dx}_{\text{read out, then integrate}} \quad . \tag{7}$$

Since g is nonlinear, the pooled readout is generally not additive. Mean pooling also discards the size of the region, so the pooled prediction does not grow with the area. Our integral readout is additive and grows with the area by construction.

In practice, we evaluate the integral on a fixed lattice of atoms $\{x_j\}$ with areas $\{w_j\}$, shared by all regions,

$$\alpha_q(A) \approx \sum_{x_j \in A} a_q(x_j)\, w_j. \tag{8}$$

Each atom is assigned to exactly one region of any partition by its location, so the additivity holds exactly on the lattice, regardless of its resolution. The resolution only limits how finely region boundaries can be resolved.

3.3 Dirichlet Random Measure

Consider a region P with a known total $N(P)$, such as a census tract, and any partition $\pi = \{A_1, \ldots, A_m\}$ of P. For simplicity, we drop the subscript q in this section. Given the total, the counts over the partition follow a Dirichlet–multinomial distribution (Mosimann, 1962),

$$\big(N(A_1), \ldots, N(A_m)\big) \mid N(P) \sim \mathrm{DirMult}\big(N(P),\, \tau\alpha(A_1), \ldots, \tau\alpha(A_m)\big), \tag{9}$$

which gives the predictive distribution p_θ in Problem 1. Here $\tau > 0$ is a global concentration multiplier. Every sample is a vector of nonnegative integers that sums exactly to $N(P)$, so the counts are conserved exactly. The expected share of A_i is $\alpha(A_i)/\alpha(P)$, and τ only controls the dispersion around it.

This distribution is the marginal on π of a Dirichlet process with base measure $\tau\alpha$ on P, from which $N(P)$ points are drawn (Ferguson, 1973). The construction requires α to be a measure, which holds because α integrates a nonnegative intensity (Section 3.2). By the aggregation property of the Dirichlet distribution, merging the regions of a finer partition yields exactly the distribution on the coarser partition (Ferguson, 1973). The predictive distributions on different partitions are therefore mutually consistent.

A query region R may intersect several regions $P_1, \ldots, P_T$ with known totals, for example when it is not aligned with census tracts. The count within each intersection then follows a beta-binomial distribution (Skellam, 1948),

$$N(R \cap P_t) \mid N(P_t) \sim \mathrm{BetaBin}\big(N(P_t),\, \tau\alpha(R \cap P_t),\, \tau\alpha(P_t \setminus R)\big). \tag{10}$$

The counts in different intersections are independent, and the count of R is their sum.

3.4 Training as a Predictor and a Learner

As a predictor. We fit the model on training regions with known totals, such as census tracts. Each training region P has observed counts $\mathbf{n}_P$ on its finest available partition $\{A_1, \ldots, A_m\}$, such as census blocks. The loss is the negative log-likelihood of Eq. (9),

$$\mathcal{L}(\theta) = -\sum_P \log \mathrm{DirMult}\big(\mathbf{n}_P \mid N(P),\, \tau\alpha(A_1), \ldots, \tau\alpha(A_m)\big). \tag{11}$$

The encoder and the output head are trained jointly, and τ is selected on validation regions. At inference, the trained field is integrated over any partition, including partitions unseen during training, without retraining. Algorithm 1 summarizes the procedure.

As a learner. We pretrain the encoder on several datasets. Each dataset observes one quantity on its own partition, possibly in a different city. All quantities share the encoder f_θ and use their own heads ϕ_q. Each dataset contributes a loss of the form in Eq. (11), computed on its own partition by integrating the same field. The partitions thus need no resampling to a common grid. To transfer to a new city, we freeze f_θ and train a new head with a few labels. Algorithm 2 summarizes the procedure, and Appendix B gives the implementation details.

Algorithm 1 Neural DRM as a predictor

Require: Context c, training regions $\{P\}$ with totals $N(P)$ and counts $\mathbf{n}_P$ on their finest partitions, validation regions
Ensure: Trained field a and concentration multiplier τ
Assign each atom x_j to the regions that contain it
repeat
  for each training region P **do**
    Compute $a(x_j)$ for the atoms in P by Eq. (4)
    Compute $\alpha(A_i)$ for each cell of its partition by Eq. (8)
  end for
  Update θ by a gradient step on $\mathcal{L}(\theta)$ in Eq. (11)
until the validation loss stops decreasing
Select τ on the validation regions
Inference. For a region P with total $N(P)$ and any partition π of P, compute α on π by Eq. (8) and predict by Eq. (9), or by Eq. (10) for a query region that spans several regions with known totals

Algorithm 2 Neural DRM as a learner

Require: Source datasets $\{(\pi_s, \{N_s(A)\}_{A\in\pi_s})\}_{s=1}^{S}$ with region totals, a target city with a few labels
Ensure: Pretrained encoder f_θ and a head ϕ for the target city
Initialize the shared encoder f_θ and the heads $\{\phi_q\}$
repeat
  for each dataset s with quantity q_s **do**
    Compute $a_{q_s}(x_j)$ with f_θ and ϕ_{q_s} by Eq. (4)
    Compute α_{q_s} on its own partition π_s by Eq. (8)
    Compute its loss $\mathcal{L}_s$ by Eq. (11)
  end for
  Update f_θ and $\{\phi_q\}$ by a gradient step on $\sum_s \mathcal{L}_s$
until the validation loss stops decreasing
Transfer. Freeze f_θ, train a new head ϕ on the labeled regions of the target city by Eq. (11), and select τ on validation
Predict on any partition of the target city as in Algorithm 1

4 Experiments

In this section, we conduct comprehensive experiments to answer following three research questions.

- **RQ1.** As a predictor, does the neural Dirichlet random measure conserve quantities across partitions while predicting more accurately than the conventional pooled readout?
- **RQ2.** As a learner, does the encoder learned from data on different partitions transfer to new cities better than encoders pretrained in the conventional way?
- **RQ3.** Can learning continuous fields, instead of representations on discrete partitions, serve as a new direction for representation learning in urban foundation models?

4.1 Experimental Settings

Data. We use the ten major cities in the U.S., with 223,617 census blocks in 7,131 census tracts. The targets are the block population from the 2020 Census (U.S. Census Bureau, 2021a) and the block counts of OpenStreetMap POIs (OpenStreetMap contributors, 2017). Given the total of each tract, the task is to predict the counts of its blocks. To evaluate on partitions unseen in training, we also partition each test tract in two other ways. The first uses block groups, each consisting of whole blocks. The second uses pieces cut by a rotated square grid with a side of 300, 150, or 80 m. The pieces are not aligned with block boundaries. We evaluate only POIs on them, because only POIs have exact point locations to count. As inputs, we use AlphaEarth embeddings (Brown et al., 2025) and GHSL built-up layers (Pesaresi & Politis, 2023) separately. Both are aggregated to a lattice of atoms with a spacing of 40 m. Appendix C gives the data details.

Table 1: Comparison of our predictor with the pooled readout and with allocation by area. Each cell gives the mean relative reduction in MAE over the ten cities. Direct and two-step are the two ways for the pooled readout to predict on a partition. The last row gives the median additivity error.

		AlphaEarth			GHSL		
Partition	Target	Direct	Two-step	By area	Direct	Two-step	By area
Blocks	Population	41%	–	42%	36%	–	36%
	POI	17%	–	21%	17%	–	17%
Block groups	Population	49%	33%	44%	38%	24%	36%
	POI	23%	9%	23%	19%	8%	17%
Pieces, 300 m	POI	14%	9%	20%	12%	6%	14%
Pieces, 150 m	POI	7%	8%	17%	5%	6%	11%
Pieces, 80 m	POI	4%	6%	12%	2%	4%	8%
Additivity error		Ours 9.3×10^{-10}			Pooled readout 0.86		

Implementation details. For the predictor, we train on the training tracts of each fold and select the hyperparameters and τ on 10% of them for validation. The pooled readout in Eq. (7) is trained with the same MLP family, the same Dirichlet–multinomial likelihood, and the same selection procedure, differing from our method only in the readout. For the learner, we pretrain the encoder on nine cities and transfer it to the test one, where 5%, 20%, or 100% of the training tracts are labeled. All learner variants share the same downstream head and optimizer. Appendix B gives more details.

Evaluation protocol. We evaluate on the test tracts of three spatial folds, and the test tracts of each fold form a contiguous region (Roberts et al., 2017). We measure prediction error by the mean absolute error (MAE) of the predictive mean. We measure how well the predictive distribution fits the observations by the negative log-likelihood (NLL) per region. Up to a constant that does not depend on the model, its expectation equals the KL divergence from the data distribution to the predictive distribution (Gneiting & Raftery, 2007), so we compare methods by their NLL differences rather than by absolute values. For the NLL, we pair the difference between our method and a compared variant on each test tract and compute 95% confidence intervals by a tract-level bootstrap (Efron & Tibshirani, 1994). A city counts as significantly better when its interval lies entirely below zero, significantly worse when it lies entirely above zero, and tied otherwise. Appendix C gives the formulas of these metrics.

4.2 Neural DRM as a Predictor (RQ1)

We compare with the pooled readout, its size-aware density and size-input variants (Appendix C), and allocation by area (Goodchild & Lam, 1980). On unseen partitions, the pooled readout predicts each region directly or in two steps, aggregating its block predictions by area. We also compute MAE confidence intervals.

Conservation. We measure conservation by the additivity error, the relative deviation between the quantity computed directly for a block group and the sum over its blocks. Our method conserves quantities exactly. Its additivity error has a median of 9.3×10^{-10} and a maximum of 1.2×10^{-8} over all cities, folds, and settings, which is at the level of numerical precision. The median for the pooled readout is 0.86. Adding the size of a region to the pooled readout reduces the median to between 0.04 and 0.15 with the density variant and between 0.44 and 0.53 with the size-input variant (Table 4), but it does not remove the error. Conservation thus comes from the integral readout itself, not from patching the pooled readout.

Accuracy on the training partition. Table 1 summarizes the MAE, and Appendix D gives the significance tests and the results in each city. On blocks, the partition used in training, our method has a lower MAE than the pooled readout in all 10 cities under all four settings, and the reduction is significant in 9 or 10 cities. The MAE decreases on average by 41% with AlphaEarth and 36% with GHSL for population, and by about 17% for POIs. For the NLL, our method is significantly better in 8 cities for population with either input (Table 5). For POIs, where most blocks have none, it is significantly better in 6 cities with AlphaEarth and 3 with GHSL, and most other cities are tied. Against allocation by area, it is significantly better in every city and setting in both metrics, so the

Table 2: Reduction in MAE by our pretrained encoder on blocks, relative to each compared encoder and computed over the test tracts of all ten cities. Columns give the fraction of labeled tracts in the target city.

	Population			POI		
Compared encoder	5%	20%	100%	5%	20%	100%
Fixed-partition pretraining	16%	15%	15%	2%	1%	3%
Pooled pretraining	8%	7%	7%	−1%	1%	1%
Input features	6%	3%	2%	1%	1%	2%
Random encoder	25%	24%	24%	4%	7%	7%
Target-city training	4%	0%	−12%	−3%	−4%	−7%

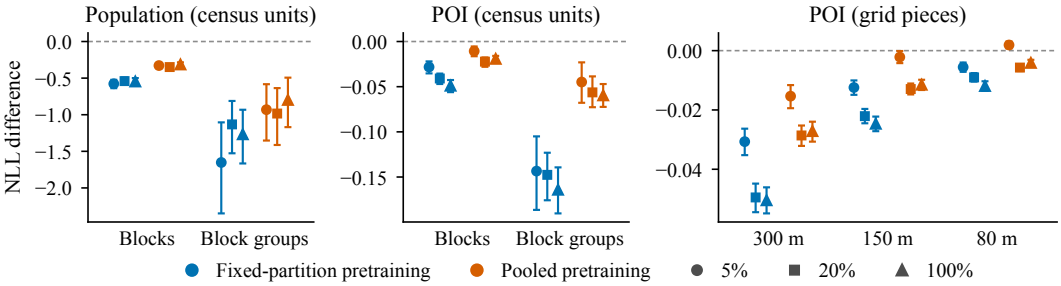


Figure 3: Paired NLL difference between our pretrained encoder and the two conventional pretraining schemes, with 95% confidence intervals. Negative values favor our encoder. Markers give the fraction of labeled tracts in the target city.

spatial context is informative. The size-aware variants largely close the gap in NLL (Table 4), and the size-input variant has a significantly lower NLL in 3 to 7 cities. However, our MAE is still significantly lower in 5 to 10 cities and significantly higher in only one city, and neither variant conserves quantities.

Prediction on unseen partitions. We apply the trained models to block groups and pieces without retraining, and our method simply integrates the same field over them. Compared with the two-step prediction, our method has a significantly lower NLL in 7 to 10 cities for pieces at every scale with both inputs, and no city is significantly worse (Table 6). Compared with the direct prediction on pieces, it is better in most cities at 300 m and 150 m, and the results are mixed at 80 m. On block groups, it is never significantly worse with AlphaEarth, while the two conventional predictions are significantly better in 2 to 7 cities with GHSL. In MAE, our method is never significantly worse than either conventional prediction on any unseen partition and is significantly better in 5 to 10 cities. The size-aware variants have a significantly lower NLL in many cities on some unseen partitions, such as 80 m pieces with AlphaEarth and block groups with GHSL, but they do not conserve quantities.

4.3 Neural DRM as a Learner (RQ2)

In pretraining, the population is observed on blocks and the POIs on block groups, a coarser partition. We evaluate on each city in turn. We compare with fixed-partition pretraining, the conventional scheme, and with pooled pretraining on the same two partitions, the input features without an encoder, a random encoder, and target-city training (Appendix C).

Transfer to new cities. Against fixed-partition pretraining, our encoder reduces the MAE on blocks by about 15% for population at every label fraction and by 1% to 3% for POIs (Table 2). Its NLL is significantly lower at every fraction, and lower in all 10 cities for population and in 8 or 9 cities for POIs (Figure 3 and Table 7). It also has a significantly lower NLL than pooled pretraining on the same two partitions, so the gain comes from the integral readout rather than from the data. Both conventional schemes even have a significantly higher NLL than the input features in 11 of the 12 settings, which indicates negative transfer (Wang et al., 2019) (Table 8). In contrast, our encoder is significantly better than the input features for population, and for POIs with 20% labels. The random encoder is much worse, so the gain does not come from the extra nonlinearity. Target-city training is the strongest competitor. Our encoder has a significantly lower NLL for population with

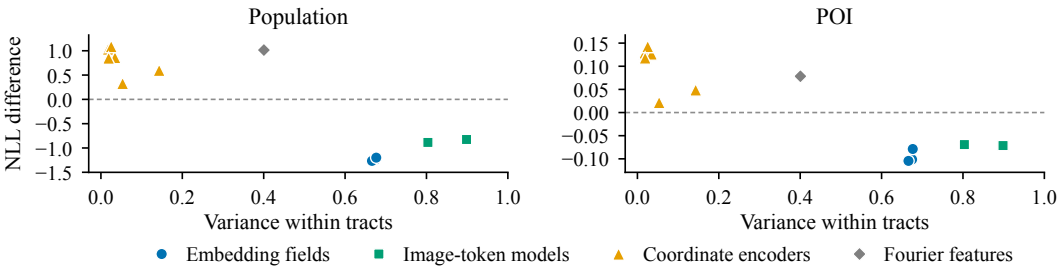


Figure 4: Each point is an input representation. The horizontal axis gives the fraction of its variance within tracts. The vertical axis gives the NLL difference between our field and allocation by area on blocks, averaged over the ten cities, where negative values favor our field.

5% and 20% labels but is worse with all labels, and it has a higher MAE for POIs. Pretraining thus helps most when labels are scarce.

Transfer to unseen partitions. On block groups and pieces, without retraining, our encoder has a significantly lower NLL than fixed-partition pretraining at every label fraction (Figure 3). It is also significantly better than pooled pretraining, except on 80 m pieces with 5% labels. Against the input features, it is significantly better on pieces only with 20% labels and is not better on block groups. Target-city training is better on pieces (Table 7).

4.4 Toward Partition-Free Urban Foundation Models (RQ3)

Resolution of input representations. A field can only be as fine as its input. Figure 4 compares twelve input representations (Appendix C). AlphaEarth and GHSL keep about two thirds of their variance within tracts, and our field beats allocation by area in every city. Image-token models, such as RemoteCLIP (Liu et al., 2024) and DINOv2 (Oquab et al., 2023), vary even more within tracts and beat it in 9 or 10 cities. Coordinate encoders, such as GeoCLIP (Vivanco Cepeda et al., 2023), SatCLIP (Klemmer et al., 2025), SINR (Cole et al., 2023), and CSP (Mai et al., 2023b), keep only 2% to 14% of their variance within tracts. With them, our field is never significantly better than allocation by area for population and is significantly worse in 47 of the 60 cases (Table 9). Fourier features (Tancik et al., 2020) of the coordinates vary within tracts but carry little urban information, and they do not help either. Field-based urban foundation models thus need input embeddings that are both fine and informative.

5 Discussion and Conclusion

Limitations. Our method has three limitations. First, the predictor is conditioned on the totals of regions, so it learns only the shares of subregions and not the absolute level of the quantity. Second, the resolution of the field is limited by that of its input representation. Representations that barely vary within regions, such as coordinate encoders, cannot support it. Third, our experiments cover only ten U.S. cities and two quantities, and the learner is a small-scale test on top of AlphaEarth.

Future work. Three directions are promising. The first is to replace the Dirichlet random measure with Poisson or negative binomial random measures (Kingman, 1967; Zhou & Carin, 2013), so that the model can predict without totals. The second is to learn the field end to end from raw data such as imagery, rather than on top of existing embeddings. The third is to pretrain at scale on the many statistics that countries publish on different partitions, covering more cities and more quantities.

Conclusion. We propose the neural Dirichlet random measure, which represents urban space as a continuous intensity field whose integral over any region gives its quantity. As a predictor, it conserves quantities exactly and predicts more accurately than the conventional pooled readout on both the training partition and unseen partitions. As a learner, it learns directly from data on different partitions and transfers to new cities and new partitions better than conventional fixed-partition pretraining. A single field thus serves all partitions, without learning a representation for each of them. This shows that learning continuous fields, rather than representations on discrete partitions, is a feasible new direction for urban representation learning.

AI USE STATEMENT

In this work, we used generative AI tools to assist in writing code for implementing our methods and experiments. We have not used generative AI tools for the other tasks that require disclosure, such as generating data, developing theoretical models, formulating mathematical claims, proposing hypotheses, designing the research methodology, or interpreting results. Additionally, we used generative AI tools to visualize experimental results, to search the literature for related work that our survey had missed, and to polish the text and correct obvious writing errors. We have reviewed all AI-assisted work. The authors reviewed and tested all AI-assisted code, checked every reported number against the experiment outputs, and verified every suggested reference against its original source. We take responsibility for the final content of this work, including text and claims produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithms 1 and 2 describe our method. Appendix B gives the architectures, optimization, and hyperparameter grids. Appendix C describes the cities, targets, input representations, spatial folds, unseen partitions, and compared methods, and gives the formulas of all metrics. Appendix D reports the complete results, including those in each city. All data are public, and Section 4.1 and Appendix C cite their sources. Our code is available at `https://anonymous.4open.science/r/NeuralDRM-9246`, with instructions to prepare the data, run all experiments, and regenerate the tables and figures.

REFERENCES

Donghyun Ahn, Jeasurk Yang, Meeyoung Cha, Hyunjoo Yang, Jihee Kim, Sangyoon Park, Sungwon Han, Eunji Lee, Susang Lee, and Sungwon Park. A human-machine collaborative approach measures economic development using satellite imagery. *Nature Communications*, 14(1):6811, 2023.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Christopher F Brown, Michal R Kazmierski, Valerie J Pasquarella, William J Rucklidge, Masha Samsikova, Chenhui Zhang, Evan Shelhamer, Estefania Lahera, Olivia Wiles, Simon Ilyushchenko, et al. Alphaearth foundations: An embedding field model for accurate and efficient global mapping from sparse label data. *arXiv preprint arXiv:2507.22291*, 2025.

Marshall Burke, Anne Driscoll, David B Lobell, and Stefano Ermon. Using satellite imagery to understand and promote sustainable development. *Science*, 371(6535):eabe8628, 2021.

Elijah Cole, Grant Van Horn, Christian Lange, Alexander Shepard, Patrick Leary, Pietro Perona, Scott Loarie, and Oisin Mac Aodha. Spatial implicit neural representations for global-scale species mapping. In *International conference on machine learning*, pp. 6320–6342. PMLR, 2023.

Yezhen Cong, Samar Khanna, Chenlin Meng, Patrick Liu, Erik Rozi, Yutong He, Marshall Burke, David Lobell, and Stefano Ermon. Satmae: Pre-training transformers for temporal and multi-spectral satellite imagery. *Advances in Neural Information Processing Systems*, 35:197–211, 2022.

Daryl J Daley and David Vere-Jones. *An introduction to the theory of point processes: volume I: elementary theory and methods*. Springer, 2003.

Matthias Drusch, Umberto Del Bello, Sébastien Carlier, Olivier Colin, Veronica Fernandez, Ferran Gascon, Bianca Hoersch, Claudia Isola, Paolo Laberinti, Philippe Martimort, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote sensing of Environment*, 120:25–36, 2012.

Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*. Chapman and Hall/CRC, 1994.

Thomas S Ferguson. A bayesian analysis of some nonparametric problems. *The annals of statistics*, pp. 209–230, 1973.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Michael F Goodchild and N Siu-Ngan Lam. Areal interpolation: A variant of the traditional spatial problem. *Geo-processing*, 1(3):297–312, 1980.

Carol A Gotway and Linda J Young. Combining incompatible spatial data. *Journal of the American Statistical Association*, 97(458):632–648, 2002.

Sungwon Han, Donghyun Ahn, Hyunji Cha, Jeasurk Yang, Sungwon Park, and Meeyoung Cha. Lightweight and robust representation of economic scales from satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 428–436, 2020a.

Sungwon Han, Donghyun Ahn, Sungwon Park, Jeasurk Yang, Susang Lee, Jihee Kim, Hyunjoo Yang, Sangyoon Park, and Meeyoung Cha. Learning to score economic development from satellite imagery. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2970–2979, 2020b.

Zhenyu Han, Yanxin Xi, Tong Xia, Yu Liu, and Yong Li. Devil in the landscapes: Inferring epidemic exposure risks from street view imagery. In *Proceedings of the 31st ACM International Conference on Advances in Geographic Information Systems*, pp. 1–4, 2023.

Xixuan Hao, Wei Chen, Yibo Yan, Siru Zhong, Kun Wang, Qingsong Wen, and Yuxuan Liang. Urbanvlp: Multi-granularity vision-language pretraining for urban socioeconomic indicator prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28061–28069, 2025.

Weiming Huang, Daokun Zhang, Gengchen Mai, Xu Guo, and Lizhen Cui. Learning urban region representations with pois and hierarchical graph infomax. *ISPRS Journal of Photogrammetry and Remote Sensing*, 196:134–145, 2023.

Neal Jean, Marshall Burke, Michael Xie, W Matthew Alampay Davis, David B Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353(6301): 790–794, 2016.

Neal Jean, Sherrie Wang, Anshul Samar, George Azzari, David Lobell, and Stefano Ermon. Tile2vec: Unsupervised representation learning for spatially distributed data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3967–3974, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

John Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

Konstantin Klemmer, Esther Rolf, Caleb Robinson, Lester Mackey, and Marc Rußwurm. Satclip: Global, general-purpose location embeddings with satellite imagery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 4347–4355, 2025.

Ho Chung Law, Dino Sejdinovic, Ewan Cameron, Tim Lucas, Seth Flaxman, Katherine Battle, and Kenji Fukumizu. Variational learning on aggregate outputs with gaussian processes. *Advances in neural information processing systems*, 31, 2018.

Tong Li, Shiduo Xin, Yanxin Xi, Sasu Tarkoma, Pan Hui, and Yong Li. Predicting multi-level socioeconomic indicators from structural urban imagery. In *Proceedings of the 31st ACM international conference on information & knowledge management*, pp. 3282–3291, 2022.

Yi Li, Weiming Huang, Gao Cong, Hao Wang, and Zheng Wang. Urban region representation learning with openstreetmap building footprints. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1363–1373, 2023.

Fan Liu, Delong Chen, Zhangqingyun Guan, Xiaocong Zhou, Jiale Zhu, Qiaolin Ye, Liyong Fu, and Jun Zhou. Remoteclip: A vision language foundation model for remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–16, 2024.

Yu Liu, Xin Zhang, Jingtao Ding, Yanxin Xi, and Yong Li. Knowledge-infused contrastive learning for urban imagery-based socioeconomic prediction. In *Proceedings of the ACM web conference 2023*, pp. 4150–4160, 2023.

Zhicheng Liu, Fabio Miranda, Weiting Xiong, Junyan Yang, Qiao Wang, and Claudio Silva. Learning geo-contextual embeddings for commuting flow prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 808–816, 2020.

Adyasha Maharana and Elaine Okanyene Nsoesie. Use of deep learning to examine the association of the built environment with prevalence of neighborhood adult obesity. *JAMA network open*, 1 (4):e181535, 2018.

Gengchen Mai, Weiming Huang, Jin Sun, Suhang Song, Deepak Mishra, Ninghao Liu, Song Gao, Tianming Liu, Gao Cong, Yingjie Hu, et al. On the opportunities and challenges of foundation models for geospatial artificial intelligence. *arXiv preprint arXiv:2304.06798*, 2023a.

Gengchen Mai, Ni Lao, Yutong He, Jiaming Song, and Stefano Ermon. Csp: Self-supervised contrastive spatial pre-training for geospatial-visual representations. In *International Conference on Machine Learning*, pp. 23498–23515. PMLR, 2023b.

Rohin Manvi, Samar Khanna, Gengchen Mai, Marshall Burke, David Lobell, and Stefano Ermon. Geollm: Extracting geospatial knowledge from large language models. In *International Conference on Learning Representations*, volume 2024, pp. 38791–38807, 2024.

Jeremy Mennis and Torrin Hultgren. Intelligent dasymetric mapping and its application to areal interpolation. *Cartography and Geographic Information Science*, 33(3):179–194, 2006.

Nando Metzger, John E Vargas-Muñoz, Rodrigo C Daudt, Benjamin Kellenberger, Thao Ton-That Whelan, Ferda Ofli, Muhammad Imran, Konrad Schindler, and Devis Tuia. Fine-grained population mapping from coarse census counts and open geodata. *Scientific Reports*, 12(1):20085, 2022.

James E Mosimann. On the compound multinomial distribution, the multivariate β-distribution, and correlations among proportions. *Biometrika*, 49(1/2):65–82, 1962.

Maya Okawa, Tomoharu Iwata, Takeshi Kurashima, Yusuke Tanaka, Hiroyuki Toda, and Naonori Ueda. Deep mixture point processes: Spatio-temporal event prediction with rich contextual information. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 373–383, 2019.

OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Sungwon Park, Sungwon Han, Donghyun Ahn, Jaeyeon Kim, Jeasurk Yang, Susang Lee, Seunghoon Hong, Jihee Kim, Sangyoon Park, Hyunjoo Yang, et al. Learning economic indicators by aggregating multi-level geospatial information. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 12053–12061, 2022.

Martino Pesaresi and Panagiotis Politis. GHS-BUILT-S R2023A - GHS built-up surface grid, derived from Sentinel2 composite and Landsat, multitemporal (1975-2030). Dataset, 2023. URL `http://data.europa.eu/89h/9f06f36f-4b11-47ec-abb0-4f8b7b1d72ea`. PID: http://data.europa.eu/89h/9f06f36f-4b11-47ec-abb0-4f8b7b1d72ea.

David R Roberts, Volker Bahn, Simone Ciuti, Mark S Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, et al. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929, 2017.

Esther Rolf, Jonathan Proctor, Tamma Carleton, Ian Bolliger, Vaishaal Shankar, Miyabi Ishihara, Benjamin Recht, and Solomon Hsiang. A generalizable and accessible approach to machine learning with global satellite imagery. *Nature communications*, 12(1):4392, 2021.

Can Rong, Xin Zhang, Yanxin Xi, Hongjie Sui, Jingtao Ding, and Yong Li. Satellites reveal mobility: A commuting origin-destination flow generator for global cities. *Advances in Neural Information Processing Systems*, 38, 2026.

Clayton Scott and Jianxin Zhang. Learning from label proportions: A mutual contamination framework. *Advances in neural information processing systems*, 33:22256–22267, 2020.

John Gordon Skellam. A probability distribution derived from the binomial distribution by regarding the probability of success as variable between the sets of trials. *Journal of the Royal Statistical Society. Series B (Methodological)*, 10(2):257–261, 1948.

Forrest R Stevens, Andrea E Gaughan, Catherine Linard, and Andrew J Tatem. Disaggregating census data for population mapping using random forests with remotely-sensed and ancillary data. *PloS one*, 10(2):e0107042, 2015.

Esra Suel, John W Polak, James E Bennett, and Majid Ezzati. Measuring social, environmental and health inequalities using deep learning and street imagery. *Scientific reports*, 9(1):6229, 2019.

Fengze Sun, Yanchuan Chang, Egemen Tanin, Shanika Karunasekera, and Jianzhong Qi. Flexireg: Flexible urban region representation learning. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 2702–2713, 2025.

Yusuke Tanaka, Toshiyuki Tanaka, Tomoharu Iwata, Takeshi Kurashima, Maya Okawa, Yasunori Akagi, and Hiroyuki Toda. Spatially aggregated gaussian processes with multivariate areal outputs. *Advances in Neural Information Processing Systems*, 32, 2019.

Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

Waldo R Tobler. Smooth pycnophylactic interpolation for geographical regions. *Journal of the American Statistical Association*, 74(367):519–530, 1979.

U.S. Census Bureau. 2020 census state redistricting data (public law 94-171) summary file. `https://www.census.gov/programs-surveys/decennial-census/about/rdo/summary-files.html`, 2021a. Accessed: 2026-05-01.

U.S. Census Bureau. 2020 TIGER/Line Shapefiles: Places. `https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.2020.html`, 2021b.

Vicente Vivanco Cepeda, Gaurav Kumar Nayak, and Mubarak Shah. Geoclip: Clip-inspired alignment between locations and images for effective worldwide geo-localization. *Advances in Neural Information Processing Systems*, 36:8690–8701, 2023.

Hongjian Wang and Zhenhui Li. Region representation learning via mobility flow. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 237–246, 2017.

Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11285–11294. IEEE, 2019.

David WS Wong. The modifiable areal unit problem (maup). In *WorldMinds: geographical perspectives on 100 problems: commemorating the 100th anniversary of the association of American geographers 1904–2004*, pp. 571–575. Springer, 2004.

Yanxin Xi, Tong Li, Huandong Wang, Yong Li, Sasu Tarkoma, and Pan Hui. Beyond the first law of geography: Learning representations of satellite imagery by leveraging point-of-interests. In *Proceedings of the ACM web conference 2022*, pp. 3308–3316, 2022.

Yanxin Xi, Yu Liu, Tong Li, Jingtao Ding, Yunke Zhang, Sasu Tarkoma, Yong Li, and Pan Hui. A satellite imagery dataset for long-term sustainable development in united states cities. *Scientific data*, 10(1):866, 2023.

Keyulu Xu, Mozhi Zhang, Jingling Li, Simon S Du, Ken-ichi Kawarabayashi, and Stefanie Jegelka. How neural networks extrapolate: From feedforward to graph neural networks. *arXiv preprint arXiv:2009.11848*, 2020.

Yibo Yan, Haomin Wen, Siru Zhong, Wei Chen, Haodong Chen, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. Urbanclip: Learning text-enhanced urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of the ACM Web Conference 2024*, pp. 4006–4017, 2024.

Zijun Yao, Yanjie Fu, Bin Liu, Wangsu Hu, and Hui Xiong. Representing urban functions through zone embedding with human mobility patterns. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)*, 2018.

Christopher Yeh, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon, and Marshall Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature communications*, 11(1):2583, 2020.

Christopher Yeh, Chenlin Meng, Sherrie Wang, Anne Driscoll, Erik Rozi, Patrick Liu, Jihyeon Lee, Marshall Burke, David B Lobell, and Stefano Ermon. Sustainbench: Benchmarks for monitoring the sustainable development goals with machine learning. *arXiv preprint arXiv:2111.04724*, 2021.

Mingyang Zhang, Tong Li, Yong Li, and Pan Hui. Multi-view joint graph representation learning for urban region embedding. In *Proceedings of the twenty-ninth international conference on international joint conferences on artificial intelligence*, pp. 4431–4437, 2021.

Weijia Zhang, Jindong Han, Zhao Xu, Hang Ni, Hao Liu, and Hui Xiong. Urban foundation models: A survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6633–6643, 2024.

Weijia Zhang, Jindong Han, Zhao Xu, Hang Ni, Tengfei Lyu, Hao Liu, and Hui Xiong. Towards urban general intelligence: A review and outlook of urban foundation models. *ACM Transactions on Intelligent Systems and Technology*, 2025.

Yivan Zhang, Nontawat Charoenphakdee, Zhenguo Wu, and Masashi Sugiyama. Learning from aggregate observations. *Advances in Neural Information Processing Systems*, 33:7993–8005, 2020.

Mingyuan Zhou and Lawrence Carin. Negative binomial process count and mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):307–320, 2013.

A Related Works

A.1 Related Works on Urban Representation Learning

Urban representation learning has evolved through roughly four stages. Early work borrows the idea of word embeddings. It learns region representations from co-occurrence or proximity in a single data source, such as human mobility (Wang & Li, 2017; Yao et al., 2018) or neighboring image tiles (Jean et al., 2019). Later work fuses multi-view and multi-modal data (Zhang et al.,

2021). Contrastive learning then becomes the dominant paradigm. It pulls together different views of a region or different modalities of a location. It is applied to POI graphs (Huang et al., 2023) and building footprints (Li et al., 2023). It also aligns imagery with POIs or knowledge graphs (Xi et al., 2022; Liu et al., 2023), locations with imagery (Mai et al., 2023b; Vivanco Cepeda et al., 2023; Klemmer et al., 2025), and imagery with text (Yan et al., 2024; Liu et al., 2024; Hao et al., 2025). Generative learning has also been introduced, including masked reconstruction (Cong et al., 2022) and knowledge extraction from large language models (Manvi et al., 2024). Most recently, AlphaEarth (Brown et al., 2025) fuses multi-source and multi-temporal observations into a global embedding field at 10 m resolution. Along this line, data sources become richer. Learning paradigms shift from co-occurrence to contrastive and generative objectives. The scale expands from single cities to the globe. Representations also become finer, moving from regions to locations and embedding fields. However, even an embedding field must be pooled within a region before use.

A.2 Related Works on Downstream Task Modeling

Downstream task modeling can be organized by how its predictions relate to spatial units. The most common setting predicts on the units where labels are available. A task-specific model is trained on these units and applied to the same units. Typical tasks include estimating poverty (Jean et al., 2016; Yeh et al., 2020), economic development (Han et al., 2020b; Ahn et al., 2023), socioeconomic indicators (Li et al., 2022; Liu et al., 2023), and health outcomes (Maharana & Nsoesie, 2018; Suel et al., 2019), as well as classifying land use (Zhang et al., 2021; Li et al., 2023). The predictions of these models are defined only on their training units.

Labels are often available only for coarse administrative units. Some methods therefore learn from coarse labels and predict on finer units. For example, they aggregate information across geographic levels (Park et al., 2022), or encode areas of arbitrary size into fixed-length vectors (Han et al., 2020a). More generally, learning from label proportions (Scott & Zhang, 2020) and from aggregate observations (Zhang et al., 2020) trains instance-level models with labels given only for groups of instances.

The setting closest to ours allocates the known total of a coarse unit to finer units. Classical areal interpolation redistributes totals by area or by ancillary data (Tobler, 1979; Mennis & Hultgren, 2006). Machine learning methods learn the allocation weights from covariates and produce fine-grained population maps (Stevens et al., 2015; Metzger et al., 2022). Other models describe the quantity as a latent field and integrate it over regions, such as aggregated Gaussian processes (Law et al., 2018; Tanaka et al., 2019) and neural point processes (Okawa et al., 2019).

These methods leave a gap. Methods in the first two settings either predict only on particular units or pool representations over a region, so their predictions do not conserve quantities across partitions. Allocation methods preserve the known totals, but they mainly give point estimates without calibrated uncertainty. Latent field models are additive across regions, but their predictions are not guaranteed to sum to the known totals. Area-based allocation is consistent across partitions, but it ignores spatial context. None of these methods aims to learn reusable representations from data on different partitions.

B Implementation Details

Shared lattice. Each city uses a lattice of atoms with a spacing of 40 m. Atoms are assigned to blocks and, independently, to block groups by point-in-polygon tests.

Predictor. The encoder and the output head together form an MLP of the form $32 \to w \to w \to 1$ with a softplus output. Each input representation is reduced to 32 dimensions by a PCA fitted on atoms pooled from all ten cities, except the four GHSL channels, and is then standardized per city. We train with full-batch Adam (Kingma & Ba, 2014) for at most 700 steps and evaluate the validation loss every 20 steps for early stopping. The learning rate is selected from $\{10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$ and the width w from $\{32, 64, 128\}$ on the validation tracts. The multiplier τ is selected from $\{2^{-4}, \ldots, 2^{6}\}$ on the same tracts.

Learner. The encoder is a two-layer ReLU MLP of the form $32 \to 64 \to 64$ on the 32-dimensional AlphaEarth features, and each head is a linear layer $64 \to 1$ followed by the softplus. The background intensity is $\varepsilon_0 = 10^{-3}$. Pretraining minimizes the mean over source cities of the sum of the population and POI losses, with full-batch Adam at a learning rate of 10^{-2} for at most 700 steps, and keeps the encoder with the best validation loss. For transfer, the encoder is frozen and a new linear head is trained on 5%, 20%, or 100% of the training tracts of the target city. Early stopping and $\tau \in \{2^{-4}, \ldots, 2^{6}\}$ are selected on 10% of the training tracts, held out for validation before subsampling.

C Experimental Details

Table 3: Numbers of census blocks, census tracts, people, and OSM POIs in the ten cities.

City	Blocks	Tracts	Population	OSM POI
New York	37,672	2,264	8,766,893	92,952
Los Angeles	28,050	1,112	3,896,880	20,552
Chicago	38,746	791	2,745,924	20,313
Houston	28,419	912	2,290,023	18,652
Phoenix	18,409	382	1,606,716	22,118
Philadelphia	17,497	403	1,603,779	10,050
San Antonio	17,244	334	1,434,930	5,461
San Diego	10,929	335	1,384,465	12,455
Dallas	18,804	364	1,298,889	8,170
San Jose	7,847	234	1,011,809	9,524
Total	223,617	7,131	—	220,247

Cities and units. We use the ten largest U.S. cities by 2020 Census population. We keep the 2020 Census blocks whose centroids lie inside the TIGER 2020 Census Place polygon (U.S. Census Bureau, 2021b) of each city, and remove the tracts with fewer than three blocks or zero population. Both targets use the same set of blocks. Table 3 lists the numbers of blocks, tracts, people, and POIs in each city. The cities differ widely in size. New York has the most people, and San Jose has the fewest blocks and tracts. OSM coverage varies even more. New York has 92,952 POIs, about 17 times as many as San Antonio, while its population is only about 6 times larger.

Targets. The population target is the block count in the 2020 Census PL 94-171 redistricting file (U.S. Census Bureau, 2021a). The POI target counts the OpenStreetMap nodes tagged amenity or shop, retrieved through the Overpass API (OpenStreetMap contributors, 2017) in September 2026 and assigned to blocks by point-in-polygon tests. OSM coverage varies strongly across cities, which is a property of the data source rather than of the methods.

Inputs. AlphaEarth provides the annual 2023 satellite embedding with 64 channels at 10 m, averaged to the 40 m atoms. GHSL provides four channels, namely the built-up surface fraction and the non-residential built-up fraction in 2018 at 10 m, the average net building height in 2018 at 100 m, and the logarithm of the built-up volume in 2020 at 100 m. Section 4.4 also compares ten other representations. AlphaEarth 3y stacks the embeddings of 2019, 2021, and 2023 into 192 channels. The image-token models encode cloud-free Sentinel-2 L2A composites of summer 2023 at 10 m (Drusch et al., 2012) with RemoteCLIP ViT-L/14 (Liu et al., 2024) and DINOv2 ViT-B/14 (Oquab et al., 2023), and each patch token covers 140 m. The coordinate encoders are GeoCLIP (Vivanco Cepeda et al., 2023), SatCLIP with $L = 10$ and $L = 40$ (Klemmer et al., 2025), SINR (Cole et al., 2023), and CSP pretrained on iNaturalist and on fMoW (Mai et al., 2023b). Fourier features of the coordinates (Tancik et al., 2020) serve as an encoding without pretraining. The coordinate encodings use shared hyperparameters for both readouts, with a learning rate of 3×10^{-3} and a width of 64.

Spatial folds. For each city, we project the tract centroids on the directions of 0, 60, and 120 degrees, and split the tracts 80/20 along each direction. The 20% side forms a contiguous test

region, and 10% of the training tracts are held out for validation. For the learner, pretraining uses the first fold in each source city.

Unseen partitions. Block groups are the census block groups, each consisting of whole blocks. Pieces are the intersections of each test tract with a square grid of side $L \in \{300, 150, 80\}$ m, rotated by an angle of $\pi/6 + 0.4$ and shifted by an offset that differs across folds. The POI count of a piece is the number of POI points inside it. To integrate the field over pieces, we split each atom into 4×4 sub-cells of 10 m that share its intensity, and assign each sub-cell to the piece that contains its center. The few POIs in pieces without any sub-cell center, at most 0.02% per city and scale, are assigned to the nearest sub-cell. All trained models are applied to these partitions without retraining, with their τ unchanged.

Compared methods for the predictor. For a region A, let $|A| = \sum_{x_j \in A} w_j$ be its area and $\bar{c}_A$ the mean representation of its atoms. All methods use an MLP g of the same family and the likelihood in Eq. (9). They differ only in the quantity of a region,

$$\alpha^{\text{ours}}(A) = \sum_{x_j \in A} \big(\text{softplus}(g(c(x_j))) + \varepsilon_0\big)\, w_j, \tag{12}$$

$$\alpha^{\text{pool}}(A) = \text{softplus}\big(g(\bar{c}_A)\big), \tag{13}$$

$$\alpha^{\text{dens}}(A) = |A|\, \big(\text{softplus}(g(\bar{c}_A)) + \varepsilon_0\big), \tag{14}$$

$$\alpha^{\text{size}}(A) = \text{softplus}\big(g([\bar{c}_A, s_A])\big), \tag{15}$$

$$\alpha^{\text{area}}(A) = |A|, \tag{16}$$

for our method, the pooled readout, the density variant, the size-input variant, and allocation by area. Here s_A is $\log |A|$ standardized over the training blocks, and allocation by area selects τ on the training tracts. On a region R of an unseen partition, the direct prediction evaluates these quantities on R itself. The two-step prediction first predicts the blocks b of the tract and then aggregates them by area,

$$\alpha^{\text{two-step}}(R) = \sum_b \alpha(b)\, \frac{|b \cap R|}{|b|}, \tag{17}$$

which sums whole blocks for block groups and spreads blocks over pieces. Our own two-step prediction in Appendix D applies this rule to α^{ours}, so it ignores the variation of the field within blocks.

Compared encoders for the learner. All encoders and heads follow Appendix B. In the target city, a new linear head ϕ is trained on blocks with the integral readout

$$\alpha(A) = \sum_{x_j \in A} \big(\text{softplus}(\phi(h(x_j))) + \varepsilon_0\big)\, w_j, \tag{18}$$

where h is the representation given by each compared encoder. The representation is frozen except in target-city training. Our encoder is pretrained with the integral readout, with the population observed on blocks and the POIs on block groups. Pooled pretraining uses the same data and architecture but the pooled readout $\alpha_q(A) = \text{softplus}(\phi_q(f_\theta(\bar{c}_A)))$. Fixed-partition pretraining aggregates both targets onto block groups and also uses the pooled readout, which is the conventional setting of region representation learning. The input features use $h(x) = c(x)$ without an encoder. The random encoder is f_θ at a random initialization. Target-city training trains f_θ and ϕ jointly from scratch on the labeled tracts of the target city.

Metrics. Consider a test tract P with the known total $N(P)$ and a partition $\{A_1, \ldots, A_m\}$ of P with observed counts $\mathbf{n} = (n_1, \ldots, n_m)$. Let $\alpha(A_i)$ be the quantity that the evaluated method gives to A_i. The predictive mean of $N(A_i)$ under Eq. (9) is $\hat{n}_i = N(P)\, \alpha(A_i) / \sum_{l=1}^{m} \alpha(A_l)$. The MAE and the NLL of the tract are

$$\text{MAE}(P) = \frac{1}{m} \sum_{i=1}^{m} \big|\hat{n}_i - n_i\big|, \tag{19}$$

$$\text{NLL}(P) = -\frac{1}{m} \log \text{DirMult}\big(\mathbf{n} \mid N(P),\, \tau\alpha(A_1), \ldots, \tau\alpha(A_m)\big), \tag{20}$$

where the Dirichlet–multinomial probability with parameters $\beta_1, \ldots, \beta_m$ and $\beta_0 = \sum_{i=1}^{m} \beta_i$ is

$$\mathrm{DirMult}(\mathbf{n} \mid N, \beta_1, \ldots, \beta_m) = \frac{N!\,\Gamma(\beta_0)}{\Gamma(N+\beta_0)} \prod_{i=1}^{m} \frac{\Gamma(n_i+\beta_i)}{n_i!\,\Gamma(\beta_i)}. \tag{21}$$

Both metrics are averaged over the test tracts of each fold and then over the three folds. Let p^* be the data distribution of the counts and p_θ the predictive distribution. The expected NLL decomposes as

$$\mathbb{E}_{\mathbf{n}\sim p^*}\big[-\log p_\theta(\mathbf{n})\big] = D_{\mathrm{KL}}\big(p^* \,\|\, p_\theta\big) + H\big(p^*\big), \tag{22}$$

where the entropy $H(p^*)$ does not depend on the model. The difference in NLL between two methods thus estimates the difference in their KL divergences. For a compared method, the paired difference on a test tract is

$$d(P) = \mathrm{NLL}^{\mathrm{ours}}(P) - \mathrm{NLL}^{\mathrm{comp}}(P), \tag{23}$$

and likewise for the MAE. The differences of a city are pooled over its three folds. The 95% confidence interval is given by the 2.5th and 97.5th percentiles of the mean of d over 2,000 bootstrap resamples of the test tracts. The relative reduction in MAE in Table 1 is averaged over the ten cities,

$$r = \frac{1}{10} \sum_{c=1}^{10} \left(1 - \frac{\mathrm{MAE}_c^{\mathrm{ours}}}{\mathrm{MAE}_c^{\mathrm{comp}}}\right), \tag{24}$$

where MAE_c is the MAE in city c. For a block group B that consists of the blocks $b_1, \ldots, b_k$, the relative deviation is

$$e(B) = \frac{\big|\alpha(B) - \sum_{j=1}^{k} \alpha(b_j)\big|}{\sum_{j=1}^{k} \alpha(b_j)}, \tag{25}$$

where $\alpha(B)$ is computed directly from the atoms of B for our method and from the pooled representation of B for the pooled readout and its size-aware variants. The additivity error of a fold is the median of $e(B)$ over its block groups. For the learner, the paired differences are pooled over the test tracts of all ten cities before the bootstrap, and the reduction in MAE in Table 2 uses the MAE over these tracts.

Resolution diagnostic. Let $\tilde{c}(x_j)$ be the standardized representation of atom x_j in a city. The fraction of its variance within tracts is

$$\rho = \frac{\sum_j \big\|\tilde{c}(x_j) - \bar{c}_{t(j)}\big\|^2}{\sum_j \big\|\tilde{c}(x_j) - \bar{c}\big\|^2}, \tag{26}$$

where $t(j)$ is the tract that contains x_j, $\bar{c}_t$ is the mean over the atoms of tract t, and $\bar{c}$ is the mean over the city. It is zero when the representation is constant within every tract. The within-tract R^2 is that of a linear regression of the log population density of blocks on their mean representations, after the tract means are removed from both.

D Additional Results

D.1 Predictor

Table 4: Significance tests of our predictor against all compared methods on blocks. Each cell gives the numbers of cities in which our method is significantly better, significantly worse, or tied. The last rows give the median additivity error of each method over the cities and folds.

	Population				POI			
	AlphaEarth		GHSL		AlphaEarth		GHSL	
Ours versus	NLL	MAE	NLL	MAE	NLL	MAE	NLL	MAE
Pooled readout	8/0/2	10/0/0	8/1/1	10/0/0	6/1/3	9/0/1	3/0/7	9/0/1
Density variant	4/0/6	10/0/0	1/5/4	6/0/4	3/1/6	6/0/4	3/1/6	5/0/5
Size-input variant	3/3/4	8/0/2	0/7/3	6/1/3	0/6/4	8/0/2	0/5/5	7/0/3
Allocation by area	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0	10/0/0
Median additivity error								
Ours	2.5×10^{-10}		3.7×10^{-10}		4.3×10^{-9}		4.5×10^{-9}	
Pooled readout	0.87		0.89		0.85		0.85	
Density variant	0.059		0.037		0.15		0.14	
Size-input variant	0.53		0.44		0.49		0.5	

Table 5: Paired NLL difference per block between our method and the pooled readout on blocks, with 95% confidence intervals from a tract-level bootstrap. Negative values favor our method, and bold entries are significant.

	Population (2020 Census blocks)		POI count (OSM)	
City	AlphaEarth	GHSL built-up	AlphaEarth	GHSL built-up
New York	**-0.11 [-0.14,-0.08]**	**-0.13 [-0.15,-0.11]**	**+0.046 [+0.028,+0.065]**	+0.012 [-0.004,+0.030]
Los Angeles	**-0.22 [-0.27,-0.16]**	-0.06 [-0.12,+0.02]	**-0.020 [-0.032,-0.008]**	-0.007 [-0.017,+0.004]
Chicago	+0.03 [-0.01,+0.09]	**+0.14 [+0.08,+0.20]**	**-0.017 [-0.031,-0.003]**	+0.003 [-0.008,+0.015]
Houston	**-0.12 [-0.18,-0.05]**	**-0.09 [-0.14,-0.03]**	-0.020 [-0.046,+0.011]	+0.004 [-0.023,+0.037]
Phoenix	**-0.30 [-0.38,-0.22]**	**-0.22 [-0.26,-0.18]**	**-0.024 [-0.039,-0.008]**	**-0.040 [-0.053,-0.027]**
Philadelphia	**-0.15 [-0.25,-0.03]**	**-0.12 [-0.20,-0.01]**	**-0.022 [-0.039,-0.007]**	+0.003 [-0.016,+0.028]
San Antonio	**-0.49 [-0.55,-0.42]**	**-0.29 [-0.37,-0.21]**	**-0.019 [-0.034,-0.004]**	**-0.022 [-0.030,-0.013]**
San Diego	**-0.42 [-0.51,-0.31]**	**-0.28 [-0.36,-0.18]**	-0.003 [-0.024,+0.019]	+0.001 [-0.017,+0.020]
Dallas	**-0.15 [-0.23,-0.07]**	**-0.11 [-0.18,-0.03]**	-0.009 [-0.024,+0.005]	**-0.019 [-0.031,-0.006]**
San Jose	-0.07 [-0.35,+0.30]	**-0.15 [-0.27,-0.02]**	**-0.039 [-0.064,-0.015]**	-0.006 [-0.029,+0.018]
Better / worse / tie	8 / 0 / 2	8 / 1 / 1	6 / 1 / 3	3 / 0 / 7
Mean paired gap	-0.200	-0.131	-0.013	-0.007
Cities with lower MAE	10/10	10/10	10/10	10/10
Gap vs. allocation by area	-1.24 (10/10)	-1.20 (10/10)	-0.10 (10/10)	-0.08 (10/10)

Table 4 compares our predictor with every method on blocks. Against the pooled readout, our MAE is significantly lower in 9 or 10 cities in every setting. Our NLL is significantly lower in most cities, except for POIs with GHSL, where 7 cities are tied. The two size-aware variants narrow the gap in NLL. The density variant is tied with our method in most cities, except for population with GHSL, where it is better in 5 cities. The size-input variant has a significantly lower NLL in 3 to 7 cities. Against both variants, our MAE is still significantly lower in 5 to 10 cities and significantly higher only once, for population with GHSL against the size-input variant. The additivity errors show that neither variant conserves quantities. The density variant, which multiplies its output by the area, comes closest with medians from 0.04 to 0.15, while the size-input variant stays around 0.5. Against allocation by area, our method is significantly better in every city in both metrics.

Table 5 gives the paired NLL difference against the pooled readout on blocks in each city. For population, our method is significantly better in 8 cities with either input, and the gains are largest in San Antonio and San Diego. The only significantly worse cases are Chicago for population with

GHSL and New York for POIs with AlphaEarth. For POIs, the differences are small and mostly tied with GHSL, since most blocks have no POI.

Table 6: Significance tests of our predictor against all compared methods on unseen partitions. Each cell gives the numbers of cities in which our method is significantly better, significantly worse, or tied. Ours, two-step denotes our block predictions spread over pieces by area.

	AlphaEarth		GHSL	
Ours versus	NLL	MAE	NLL	MAE
Block groups, population				
Pooled readout, direct	6/0/4	10/0/0	2/3/5	10/0/0
Pooled readout, two-step	3/0/7	10/0/0	2/4/4	10/0/0
Density variant, direct	10/0/0	10/0/0	5/0/5	10/0/0
Density variant, two-step	9/0/1	8/0/2	0/3/7	2/0/8
Size-input variant, direct	0/1/9	9/0/1	0/6/4	7/0/3
Size-input variant, two-step	3/0/7	8/0/2	0/6/4	1/0/9
Allocation by area	9/0/1	10/0/0	8/0/2	10/0/0
Block groups, POI				
Pooled readout, direct	9/0/1	8/0/2	2/2/6	9/0/1
Pooled readout, two-step	1/0/9	6/0/4	1/7/2	5/0/5
Density variant, direct	8/0/2	6/0/4	7/0/3	2/0/8
Density variant, two-step	4/0/6	1/0/9	4/0/6	1/0/9
Size-input variant, direct	1/1/8	9/0/1	0/8/2	7/0/3
Size-input variant, two-step	2/2/6	3/0/7	0/7/3	1/0/9
Allocation by area	9/0/1	7/0/3	9/0/1	7/0/3
Pieces, 300 m, POI				
Pooled readout, direct	9/0/1	9/0/1	6/1/3	10/0/0
Pooled readout, two-step	10/0/0	9/0/1	7/0/3	8/0/2
Density variant, direct	5/1/4	7/0/3	8/0/2	6/1/3
Size-input variant, direct	1/1/8	6/0/4	1/3/6	5/0/5
Ours, two-step	6/0/4	9/0/1	9/0/1	10/0/0
Allocation by area	10/0/0	10/0/0	10/0/0	9/0/1
Pieces, 150 m, POI				
Pooled readout, direct	7/0/3	8/0/2	6/0/4	8/0/2
Pooled readout, two-step	9/0/1	10/0/0	10/0/0	8/0/2
Density variant, direct	1/3/6	8/0/2	5/1/4	6/0/4
Size-input variant, direct	0/5/5	3/0/7	3/4/3	4/3/3
Ours, two-step	4/4/2	10/0/0	6/0/4	10/0/0
Allocation by area	10/0/0	10/0/0	10/0/0	9/0/1
Pieces, 80 m, POI				
Pooled readout, direct	4/1/5	9/0/1	5/3/2	8/0/2
Pooled readout, two-step	7/0/3	9/0/1	10/0/0	7/0/3
Density variant, direct	1/8/1	10/0/0	5/3/2	7/0/3
Size-input variant, direct	0/8/2	3/1/6	3/4/3	5/2/3
Ours, two-step	2/5/3	10/0/0	4/2/4	10/0/0
Allocation by area	8/0/2	10/0/0	9/0/1	9/0/1

Table 6 reports the same comparisons on unseen partitions. On block groups with AlphaEarth, our method is never significantly worse than either conventional prediction. With GHSL, the conventional predictions are significantly better in 2 to 7 cities in NLL. Our MAE on block groups is significantly lower than that of both conventional predictions in 5 to 10 cities. On pieces, our method is significantly better than the two-step prediction in 7 to 10 cities at every scale, and better than the direct prediction in most cities at 300 m and 150 m. The size-aware variants are strongest on the finest pieces. At 80 m with AlphaEarth, both have a significantly lower NLL than our field in 8 cities, although neither conserves quantities. We further compare with our own two-step prediction, which spreads our block predictions over pieces by area and thus ignores the variation of the field within blocks. Our field has a significantly lower MAE than this prediction in 9 or 10 cities at every scale. Its NLL is lower in most cities at 300 m, but not at finer scales with AlphaEarth. A piece of 80 m covers only about four atoms of the lattice, which leaves the field little room to vary within it.

D.2 LEARNER

Table 7: Paired NLL difference between our pretrained encoder and each compared encoder, with 95% confidence intervals over the test tracts of all ten cities. Numbers in parentheses count the cities in which our encoder has a lower NLL.

Ours versus	5% labels	20% labels	100% labels
Population, blocks			
Fixed-partition pretraining	-0.577 [-0.637, -0.523] (10)	-0.541 [-0.590, -0.492] (10)	-0.548 [-0.598, -0.499] (10)
Pooled pretraining	-0.328 [-0.365, -0.294] (10)	-0.349 [-0.389, -0.311] (10)	-0.316 [-0.350, -0.284] (10)
Input features	-0.183 [-0.207, -0.160] (8)	-0.185 [-0.203, -0.165] (9)	-0.164 [-0.180, -0.148] (10)
Random encoder	-0.811 [-0.861, -0.761] (10)	-0.799 [-0.847, -0.752] (10)	-0.767 [-0.812, -0.723] (10)
Target-city training	-0.185 [-0.224, -0.152] (9)	-0.058 [-0.089, -0.030] (7)	$+0.055$ [$+0.040$, $+0.069$] (2)
Population, block groups			
Fixed-partition pretraining	-1.652 [-2.348, -1.104] (9)	-1.132 [-1.527, -0.812] (10)	-1.270 [-1.666, -0.932] (8)
Pooled pretraining	-0.931 [-1.353, -0.583] (9)	-0.984 [-1.413, -0.636] (10)	-0.800 [-1.170, -0.494] (9)
Input features	$+0.056$ [-0.052, $+0.180$] (8)	$+0.034$ [-0.031, $+0.100$] (6)	$+0.027$ [-0.032, $+0.086$] (5)
Random encoder	-1.273 [-1.597, -0.984] (9)	-1.130 [-1.412, -0.886] (9)	-1.024 [-1.254, -0.817] (9)
Target-city training	-0.385 [-0.536, -0.246] (9)	-0.154 [-0.281, -0.032] (7)	$+0.134$ [$+0.089$, $+0.181$] (3)
POI, blocks			
Fixed-partition pretraining	-0.028 [-0.035, -0.022] (8)	-0.041 [-0.047, -0.035] (9)	-0.049 [-0.056, -0.043] (9)
Pooled pretraining	-0.011 [-0.016, -0.005] (7)	-0.022 [-0.028, -0.018] (9)	-0.019 [-0.023, -0.016] (10)
Input features	-0.005 [-0.011, $+0.001$] (4)	-0.008 [-0.012, -0.004] (6)	-0.002 [-0.006, $+0.001$] (7)
Random encoder	-0.025 [-0.032, -0.019] (8)	-0.045 [-0.052, -0.039] (9)	-0.044 [-0.050, -0.038] (10)
Target-city training	-0.003 [-0.012, $+0.004$] (6)	$+0.002$ [-0.004, $+0.008$] (6)	$+0.019$ [$+0.014$, $+0.024$] (3)
POI, block groups			
Fixed-partition pretraining	-0.143 [-0.186, -0.105] (7)	-0.148 [-0.176, -0.123] (8)	-0.164 [-0.190, -0.139] (8)
Pooled pretraining	-0.045 [-0.068, -0.023] (7)	-0.056 [-0.073, -0.039] (7)	-0.060 [-0.072, -0.047] (9)
Input features	$+0.020$ [$+0.001$, $+0.042$] (5)	$+0.002$ [-0.015, $+0.020$] (4)	$+0.005$ [-0.008, $+0.018$] (6)
Random encoder	-0.098 [-0.125, -0.072] (8)	-0.151 [-0.182, -0.124] (8)	-0.130 [-0.149, -0.111] (10)
Target-city training	-0.009 [-0.030, $+0.012$] (6)	$+0.020$ [$+0.001$, $+0.040$] (7)	$+0.066$ [$+0.049$, $+0.083$] (4)
POI, pieces of 300 m			
Fixed-partition pretraining	-0.031 [-0.035, -0.026] (9)	-0.050 [-0.054, -0.045] (9)	-0.051 [-0.055, -0.046] (9)
Pooled pretraining	-0.015 [-0.019, -0.012] (9)	-0.029 [-0.032, -0.025] (9)	-0.027 [-0.031, -0.024] (10)
Input features	-0.003 [-0.008, $+0.001$] (6)	-0.012 [-0.015, -0.009] (8)	-0.001 [-0.004, $+0.002$] (7)
Random encoder	-0.031 [-0.036, -0.026] (9)	-0.052 [-0.057, -0.048] (10)	-0.049 [-0.053, -0.044] (10)
Target-city training	$+0.007$ [$+0.002$, $+0.012$] (8)	$+0.007$ [$+0.002$, $+0.012$] (7)	$+0.031$ [$+0.026$, $+0.037$] (2)
POI, pieces of 150 m			
Fixed-partition pretraining	-0.012 [-0.015, -0.010] (9)	-0.022 [-0.024, -0.020] (8)	-0.025 [-0.027, -0.022] (8)
Pooled pretraining	-0.002 [-0.004, -0.000] (6)	-0.013 [-0.015, -0.011] (7)	-0.012 [-0.013, -0.010] (9)
Input features	-0.001 [-0.005, $+0.002$] (6)	-0.007 [-0.009, -0.005] (7)	-0.001 [-0.003, $+0.001$] (5)
Random encoder	-0.008 [-0.011, -0.006] (6)	-0.020 [-0.022, -0.017] (10)	-0.018 [-0.020, -0.016] (8)
Target-city training	$+0.010$ [$+0.007$, $+0.013$] (6)	$+0.010$ [$+0.008$, $+0.012$] (5)	$+0.015$ [$+0.011$, $+0.019$] (4)
POI, pieces of 80 m			
Fixed-partition pretraining	-0.006 [-0.007, -0.004] (8)	-0.009 [-0.010, -0.008] (8)	-0.012 [-0.014, -0.010] (8)
Pooled pretraining	$+0.002$ [$+0.001$, $+0.003$] (6)	-0.006 [-0.007, -0.005] (4)	-0.004 [-0.005, -0.003] (6)
Input features	$+0.000$ [-0.002, $+0.002$] (5)	-0.003 [-0.004, -0.002] (6)	-0.000 [-0.002, $+0.001$] (3)
Random encoder	-0.001 [-0.002, $+0.001$] (6)	-0.006 [-0.007, -0.005] (6)	-0.006 [-0.007, -0.004] (7)
Target-city training	$+0.008$ [$+0.007$, $+0.010$] (5)	$+0.008$ [$+0.006$, $+0.009$] (3)	$+0.005$ [$+0.003$, $+0.007$] (5)

Table 7 gives the paired NLL difference between our encoder and each compared encoder on every partition. Our encoder is significantly better than both conventional pretraining schemes on every partition and label fraction, except against pooled pretraining on 80 m pieces with 5% labels. It is also significantly better than the random encoder in all settings but one. Against the input features, it is significantly better for population on blocks at every fraction, and for POIs on blocks and pieces only with 20% labels. It is not better on block groups, and it is worse there for POIs with 5% labels. Target-city training is worse than our encoder for population with 5% and 20% labels, on blocks and on block groups, but it is better with all labels. For POIs, target-city training is never significantly worse and is significantly better on every piece partition.

Table 8: Paired NLL difference between each pretrained encoder and the input features on blocks, with 95% confidence intervals. Positive values mean that pretraining hurts. Numbers in parentheses count the cities in which the pretrained encoder has a lower NLL than the input features.

Target	Labels	Fixed-partition	Pooled	Ours
Population	5%	$+0.394$ $[+0.335, +0.459]$ (0)	$+0.145$ $[+0.104, +0.190]$ (3)	-0.183 $[-0.207, -0.160]$ (8)
	20%	$+0.356$ $[+0.310, +0.405]$ (0)	$+0.164$ $[+0.124, +0.209]$ (0)	-0.185 $[-0.203, -0.165]$ (9)
	100%	$+0.384$ $[+0.338, +0.435]$ (0)	$+0.152$ $[+0.119, +0.189]$ (1)	-0.164 $[-0.180, -0.148]$ (10)
POI	5%	$+0.024$ $[+0.017, +0.030]$ (4)	$+0.006$ $[-0.002, +0.013]$ (4)	-0.005 $[-0.011, +0.001]$ (4)
	20%	$+0.033$ $[+0.027, +0.040]$ (3)	$+0.014$ $[+0.010, +0.019]$ (2)	-0.008 $[-0.012, -0.004]$ (6)
	100%	$+0.047$ $[+0.040, +0.053]$ (1)	$+0.017$ $[+0.013, +0.021]$ (1)	-0.002 $[-0.006, +0.001]$ (7)

Table 8 compares each pretrained encoder with the input features on blocks. Both conventional schemes have a higher NLL than the input features at every label fraction, significantly so in 11 of the 12 settings. Fixed-partition pretraining is worse than the input features in every city for population. Our encoder is the only pretrained encoder that improves on the input features, significantly for population at every fraction and for POIs with 20% labels.

D.3 Input representations

Table 9: Resolution diagnostic and results of our predictor on blocks with each input representation. The first two columns are averaged over the ten cities. The other columns give the numbers of cities in which our field is significantly better, significantly worse, or tied.

			Population		POI	
Representation	Variance within tracts	R^2 within tracts	vs. area	vs. pooled	vs. area	vs. pooled
Embedding fields						
AlphaEarth	0.68	0.49	10/0/0	8/0/2	10/0/0	6/1/3
AlphaEarth 3y	0.67	0.50	10/0/0	8/0/2	10/0/0	5/1/4
GHSL	0.68	0.44	10/0/0	8/1/1	10/0/0	3/0/7
Image-token models						
S2 + RemoteCLIP	0.90	0.26	10/0/0	3/4/3	9/0/1	7/1/2
S2 + DINOv2	0.80	0.30	10/0/0	3/2/5	9/0/1	7/0/3
Coordinate encoders						
GeoCLIP	0.05	0.03	0/6/4	3/3/4	2/4/4	8/0/2
SatCLIP-L10	0.14	0.01	0/8/2	0/10/0	1/5/4	4/1/5
SatCLIP-L40	0.02	0.01	0/7/3	2/6/2	0/9/1	5/2/3
SINR	0.03	0.02	0/8/2	4/2/4	1/7/2	6/3/1
CSP-iNat	0.03	0.01	0/9/1	1/3/6	3/7/0	6/2/2
CSP-fMoW	0.02	0.01	0/9/1	5/2/3	1/8/1	5/0/5
Fourier features	0.40	0.02	0/9/1	0/8/2	1/7/2	6/0/4

Table 9 relates the resolution of each input representation to the results of our predictor. Embedding fields and image-token models keep most of their variance within tracts, and our field beats allocation by area in 9 or 10 cities with each of them. AlphaEarth and GHSL also explain the most within-tract variation in population density, with R^2 from 0.44 to 0.50, and give the largest gains over the pooled readout for population. Image-token models vary even more within tracts but explain less, with R^2 of 0.26 and 0.30. Against the pooled readout, their results are mixed for population but favorable for POIs. Coordinate encoders keep less than 15% of their variance within tracts, and their R^2 is at most 0.03. With them, our field is never significantly better than allocation by area for population, and the comparison with the pooled readout is mixed. Fourier features vary within tracts but explain almost none of the population density, and they behave like the coordinate encoders. The AlphaEarth embeddings of three years perform almost the same as those of a single year.