

How to Use the SPACE Framework in a Quantitative Spatial Model

Greg Howard

June 15, 2026

This note gives a practical guide to using the SPACE (Spatially and Persistently Auto-correlated Epsilons) framework as the migration or location-choice block in a quantitative spatial model (QSM). The model follows Howard and Shao (2026). The purpose here is not to re-derive the theory. It is to show what block to replace, how to calibrate the replacement, and how to use the calibrated block in counterfactual exercises.

SPACE can be used in both dynamic and static spatial models. In dynamic settings, it gives a tractable alternative to moving-cost-based migration models. In static settings, it gives a disciplined substitution structure because population cross-elasticities are tied directly to observed migration flows. The framework is therefore useful whenever migration data are available, even if the surrounding model is otherwise static.

1 Incorporating SPACE into a QSM

Consider a standard QSM with locations $i = 1, \dots, I$, flow utility u_i coming from the productivity, trade, housing, or amenity blocks, and a migration or location-choice block. SPACE replaces only that last block. The rest of the model can usually be left unchanged.

The block being replaced is typically a discrete-choice problem over locations. In a static model, it is often written as

$$\max_i \left\{ u_i + \frac{\epsilon_i}{\sigma} \right\},$$

where ϵ_i is an idiosyncratic taste shock and σ controls the scale of that shock. In a dynamic moving-cost model, the corresponding block is often written as

$$V_{jt} = \max_i \left\{ u_{it} - \delta_{ji} + \frac{\epsilon_{it}}{\sigma} + \beta \mathbb{E}[V_{i,t+1}] \right\},$$

where V_{jt} is the value of currently being in location j , u_{it} is flow utility in destination i at date t , δ_{ji} is the cost of moving from current location j to destination i , and β is the discount factor.

1.1 Agent Optimization Problem

The SPACE replacement is

$$V(\epsilon_t) = \max_{i \in \{1, \dots, I\}} \left\{ u_{it} + \frac{\epsilon_{it}}{\nu} \right\} + \beta \mathbb{E} [V(\epsilon_{t+1}) \mid \epsilon_t]$$

where $V(\epsilon_t)$ is the value function, but which now depends on ϵ_t and not j , ϵ_{it} is an idiosyncratic utility component but which is now correlated across locations and persistent over time, and ν is an inverse scale parameter.¹ In a static model, the same expression can be used with the forward-looking term omitted.

The vector of shocks $\epsilon_t = (\epsilon_{1t}, \dots, \epsilon_{It})$ follows a generalized extreme value (GEV) distribution. Let $z_i = e^{\epsilon_{it}}$. Then the joint distribution is characterized by a generating function $G(z_1, \dots, z_I)$ such that, for thresholds x_1, \dots, x_I ,

$$\Pr(\epsilon_{1t} \leq x_1, \dots, \epsilon_{It} \leq x_I) = \exp(-G(e^{-x_1}, \dots, e^{-x_I})).$$

The function $G(\cdot)$ is homogeneous of degree one and determines the pattern of correlation across locations. It must also satisfy the regularity conditions in McFadden (1978)—which the following calibration does.

Persistence over time is introduced by specifying that the joint distribution of $(\epsilon_t, \epsilon_{t+1})$ is also GEV, with a parameter $\rho \in [0, 1)$ governing the degree of persistence. In particular, the joint generating function takes the form

$$G(e^{\epsilon_t}, e^{\epsilon_{t+1}}) = G\left(\left(\left(e^{\epsilon_{1t}}\right)^{\frac{1}{1-\rho}} + \left(e^{\epsilon_{1,t+1}}\right)^{\frac{1}{1-\rho}}\right)^{1-\rho}, \dots, \left(\left(e^{\epsilon_{It}}\right)^{\frac{1}{1-\rho}} + \left(e^{\epsilon_{I,t+1}}\right)^{\frac{1}{1-\rho}}\right)^{1-\rho}\right)$$

This construction preserves the marginal GEV distributions while inducing correlation across periods. Intuitively, higher ρ means that realizations of ϵ_{it} are more strongly correlated over time. That persistence generates inertia in location choices without explicit moving costs.

The key departure from standard models is that ϵ_{it} is jointly distributed across locations and time. There are no moving costs or switching frictions; all dynamics come from the evolution of ϵ_{it} .

¹I use a different notation because this one cannot be interpreted as a population elasticity, as σ can in the static model.

Under SPACE, population shares have a closed form:

$$p_i = \frac{e^{\nu u_i} G_i(e^{\nu u_1}, \dots, e^{\nu u_I})}{G(e^{\nu u_1}, \dots, e^{\nu u_I})}$$

where G_i is the partial derivative of G with respect to its i th term.

Migration has a corresponding closed form when $u_t = u_{t+1}$:

$$m_{i \rightarrow j} = (1 - \rho) e^{\nu(u_i + u_j)} \left(\frac{G_i(e^{\nu u_1}, \dots, e^{\nu u_I}) G_j(e^{\nu u_1}, \dots, e^{\nu u_I})}{G(e^{\nu u_1}, \dots, e^{\nu u_I})^2} - \frac{G_{ij}(e^{\nu u_1}, \dots, e^{\nu u_I})}{G(e^{\nu u_1}, \dots, e^{\nu u_I})} \right)$$

where $m_{i \rightarrow j}$ is the share of people moving from i to j , and G_{ij} is the cross-partial derivative of G with respect to its i th and j th terms.

The key implementation implication is

$$\frac{\partial p_i}{\partial u_j} = -\frac{\nu}{1 - \rho} m_{i \rightarrow j}$$

so migration flows are a sufficient statistic for population cross-elasticities.

2 Calibration: Overview

For implementation, the goal is not to search over all possible GEV structures. The goal is to choose a structure that is easy to calibrate, easy to simulate, and disciplined by observed migration data. Following Howard and Shao (2026), a convenient choice is a cross-nested GEV system in which each nest contains exactly two locations:

$$G(z_1, \dots, z_I) = \sum_{i < j} \left((w_{ij} z_i)^{\frac{1}{1-\gamma}} + (w_{ji} z_j)^{\frac{1}{1-\gamma}} \right)^{1-\gamma}.$$

Here, each pair $\{i, j\}$ is a two-location nest, w_{ij} and w_{ji} are nonnegative nest weights, and γ is the within-nest elasticity of substitution. This two-location version of the Howard-Shao calibration is useful because the relevant objects can be recovered from bilateral migration data.

The next step is to reparameterize the model in a way that is convenient for calibration. Howard and Shao define a reduced-form persistence object $\tilde{\rho}$ and a reduced-form inverse scale parameter $\tilde{\nu}$ through

$$1 - \gamma = \frac{1 - \rho}{1 - \tilde{\rho}} \quad \text{and} \quad \nu = (1 - \rho) \tilde{\nu}.$$

These reparameterizations keep the formulas well behaved as $\rho \rightarrow 1$, which is helpful for simple algebra and in cases where it is helpful to simulate the micro-behavior (which is not covered in this note). Without the ν normalization, the cross-elasticity formula would diverge in precisely the high-persistence case that makes SPACE attractive. With this normalization,

$$\frac{\partial p_i}{\partial u_j} = -\tilde{\nu} m_{i \rightarrow j}$$

in the $\rho \rightarrow 1$ limit. The parameter ρ remains the underlying persistence parameter. The objects that matter for calibration and counterfactual work are $\tilde{\rho}$ and $\tilde{\nu}$.

Rather than calibrating the raw nest weights w_{ij} , it is more convenient to work with the transformed weights

$$\tilde{w}_{ij} \equiv (w_{ij} e^{\nu u_i})^{\frac{1}{1-\gamma}} \left[(w_{ij} e^{\nu u_i})^{\frac{1}{1-\gamma}} + (w_{ji} e^{\nu u_j})^{\frac{1}{1-\gamma}} \right]^{-\gamma},$$

for each two-location nest $\{i, j\}$. These transformed weights are the practical objects. They enter the formulas for populations, migration rates, and counterfactual responses.

All calibration objects in this section can be measured as counts or as shares, as long as the same normalization is used throughout. Let p_i denote baseline population in location i , and let $m_{i \rightarrow j}$ denote baseline migration from i to j . Under this parameterization, the calibration targets become:

$$p_i = \sum_{j \neq i} \tilde{w}_{ij}, \quad m_{i \rightarrow j} = (1 - \tilde{\rho}) \frac{\tilde{w}_{ij} \tilde{w}_{ji}}{\tilde{w}_{ij} + \tilde{w}_{ji}}.$$

The first equation says that all of location i 's population must be accounted for by the pairwise nests that include i . The second says that migration between i and j is generated by the nest linking those two locations. Thus the calibration problem is simple: choose $\tilde{\rho}$ and \tilde{w}_{ij} to match observed populations and bilateral migration rates. The steps below solve this matching problem in a useful order: construct symmetric migration moments, choose the highest feasible persistence, and then recover the weights that make the population and migration equations hold.

Step 1: Construct Migration Objects

Let $m_{i \rightarrow j}$ denote the observed one-year migration flow from origin i to destination j , including the non-mover count $m_{i \rightarrow i}$. In the data, bilateral flows are not exactly symmetric, whereas the SPACE calibration uses symmetric steady-state flows. To construct the symmetric migration object used by the model for each pair $i \neq j$, take the logarithmic mean:

$$\tilde{m}_{ij} = \begin{cases} \frac{m_{i \rightarrow j} - m_{j \rightarrow i}}{\log(m_{i \rightarrow j}) - \log(m_{j \rightarrow i})}, & m_{i \rightarrow j} \neq m_{j \rightarrow i}, m_{i \rightarrow j} > 0, m_{j \rightarrow i} > 0, \\ m_{i \rightarrow j}, & m_{i \rightarrow j} = m_{j \rightarrow i} > 0, \\ 0, & m_{i \rightarrow j} = 0 \text{ or } m_{j \rightarrow i} = 0. \end{cases}$$

The second and third cases handle equal flows and pairs with a zero flow. This logarithmic mean is the object that matches the model's steady-state migration flows (Howard and Shao, 2026). In practice, using the arithmetic average is unlikely to make much of a difference anyway, as migration is fairly symmetric. For non-movers, set $\tilde{m}_{ii} = m_{i \rightarrow i}$.

Step 2: Recover Persistence via Eigenvalues

Next, define total migration out of location i as $m_i = \sum_{j \neq i} \tilde{m}_{ij}$, and define baseline population as $p_i = \sum_j \tilde{m}_{ij}$, where the sum includes non-movers. Collect these objects in a matrix M whose largest eigenvalue recovers $1 - \tilde{\rho}$.² Define

$$M_{ij} = \begin{cases} \frac{\tilde{m}_{ij}}{p_i} & i \neq j \\ \frac{m_i}{p_i} & i = j \end{cases}$$

This is *not* a transition matrix. The off-diagonal entries are symmetric migration flows normalized by origin population. The diagonal entry is different from what one might expect in a Markov matrix: here $M_{ii} = m_i/p_i$, not $1 - m_i/p_i$. In other words, the diagonal contains the total migration rate out of i , not the non-mover share.

Then compute the largest eigenvalue of M , denoted $\lambda_{\max}(M)$. In the two-location-nest calibration, Howard and Shao (2026) show that this eigenvalue equals $1 - \tilde{\rho}$, so

$$\tilde{\rho} = 1 - \lambda_{\max}(M)$$

In practice, this is a simple linear algebra step. Build M , compute its largest eigenvalue, and set $\tilde{\rho}$ using the formula above. The computation is fast even with many locations.

Step 3: Recover Weights

Let ℓ denote the eigenvector associated with λ_{\max} . The calibrated transformed weights are

²The eigenvalue construction is the Howard-Shao way of choosing the two-location-nest calibration that maximizes feasible $\tilde{\rho}$, subject to matching the symmetric migration flows and baseline populations. Once this maximal persistence is fixed, the associated eigenvector gives the weights in Step 3.

$$\tilde{w}_{ij} = \frac{1}{1 - \tilde{\rho}} \tilde{m}_{ij} \left(1 + \frac{\ell_j}{\ell_i} \right)$$

These weights satisfy the population accounting equation and, by construction, reproduce the migration moments targeted in the calibration.

Once calibration is complete, the key objects are $\tilde{\rho}$, $\tilde{\nu}$, the baseline population shares p_i , and the calibrated weights \tilde{w}_{ij} . These are enough to compute population responses to utility changes. Primitive weights w_{ij} are needed only if the population block is written in levels.

Step 4: Calibrate $\tilde{\nu}$

Unlike $\tilde{\rho}$ and the pairwise weights, $\tilde{\nu}$ is not pinned down by migration shares alone. It must come from additional discipline, such as the utility scale in the surrounding QSM or an external elasticity target.

The literature usually reports own-population elasticities rather than bilateral cross-elasticities. Under SPACE, summing the cross-effects gives

$$\frac{\partial p_i}{\partial u_i} = \tilde{\nu} m_i,$$

so

$$\frac{\partial \log p_i}{\partial u_i} = \tilde{\nu} \frac{m_i}{p_i}.$$

If an existing static-logit calibration targets an average own-population elasticity σ with u measured in log real wages, then a reasonable calibration is

$$\tilde{\nu} = \frac{\partial \log p_i / \partial u_i}{m_i / p_i} \approx \frac{\sigma}{\mathbb{E}_i[m_i / p_i]}.$$

For example, suppose an existing calibration implies an own-population elasticity to real wages of 3, and the average migration rate is $m_i / p_i \approx 0.05$. Then a reasonable calibration is $\tilde{\nu} = 3 / 0.05 = 60$.

3 Using the Model in a QSM

Once calibrated, the SPACE block can be inserted into a QSM much like a standard migration or location-choice block. The inputs are the utilities u_i implied by the rest of the model. The output is the population vector p_i .

The substitution is easiest to see by comparing the population equations. In a static logit model, one often closes the model with

$$p_i = \frac{e^{u_i/\sigma}}{\sum_k e^{u_k/\sigma}},$$

where k indexes possible destinations and σ is the logit scale parameter. In a dynamic moving-cost model, one often uses

$$m_{j \rightarrow i, t} = \frac{e^{(u_{it} - \delta_{ji} + \beta \mathbb{E}[V_{i, t+1}])/\sigma}}{\sum_k e^{(u_{kt} - \delta_{jk} + \beta \mathbb{E}[V_{k, t+1}])/\sigma}}, \quad p_{it+1} = \sum_j p_{jt} m_{j \rightarrow i, t},$$

where $m_{j \rightarrow i, t}$ is the share of people in origin j who move to destination i at date t , p_{jt} is the population share in origin j , δ_{ji} is the moving cost, and $V_{i, t+1}$ is next period's value of being in destination i . With SPACE, these closures are replaced by the population mapping below. The wage, rent, housing, and trade blocks can stay the same. What changes is the mapping from utilities into populations and migration flows.

There are two useful ways to write this mapping. Use the level formula when solving a full equilibrium model from utilities to populations. Use hat algebra when studying counterfactual changes around a calibrated baseline. Both formulas below use the two-location-nest calibration described above.

Exact Formula for Population

For the implementation formulas, work with the $\rho \rightarrow 1$ representation while holding $\tilde{\rho}$ and $\tilde{\nu}$ fixed. The first step is to recover the primitive pairwise weights w_{ij} from the calibrated objects. Let u_i^0 denote baseline utility. Then

$$w_{ij} = \tilde{w}_{ij} e^{-(1-\tilde{\rho})\tilde{\nu}u_i^0}.$$

The analogous expression recovers w_{ji} . The \tilde{w} 's describe the baseline mass attached to each pairwise nest. The w 's are the fixed primitives used to evaluate the model at any utility vector u .

Once these weights have been recovered, population can be written directly as a function of utilities:

$$p_i(u) = \sum_{j \neq i} (\tilde{w}_{ij} + \tilde{w}_{ji}) \frac{w_{ij} e^{(1-\tilde{\rho})\tilde{\nu}u_i}}{w_{ij} e^{(1-\tilde{\rho})\tilde{\nu}u_i} + w_{ji} e^{(1-\tilde{\rho})\tilde{\nu}u_j}}.$$

By construction, plugging in the baseline utility vector u^0 reproduces the baseline shares

p_i . Plugging in a counterfactual utility vector u' gives the counterfactual population vector. For equilibrium computation, this is the main object. Feed in utilities from the rest of the model, compute $p_i(u)$, and iterate until markets clear.

Hat Algebra

In many counterfactual exercises, the most convenient approach is not to resolve the full level problem. It is to work with exact hat algebra. Let u_i denote baseline utility and u'_i denote counterfactual utility. Define

$$\hat{u}_i = \exp(u'_i - u_i)$$

as the multiplicative utility change. Let $\hat{p}_i = p'_i/p_i$ denote the proportional change in population, where p'_i is counterfactual population. Under the two-location-nest calibration, population changes satisfy

$$\hat{p}_i = \hat{u}_i^{(1-\tilde{\rho})\tilde{\nu}} \sum_{j \neq i} \frac{\tilde{w}_{ij}}{p_i} \frac{\tilde{w}_{ij} + \tilde{w}_{ji}}{\tilde{w}_{ij}\hat{u}_i^{(1-\tilde{\rho})\tilde{\nu}} + \tilde{w}_{ji}\hat{u}_j^{(1-\tilde{\rho})\tilde{\nu}}}.$$

This is the same object as the level formula above, written relative to the baseline equilibrium. It uses exactly the objects produced by the calibration: the baseline shares p_i , the transformed pairwise weights \tilde{w}_{ij} , and the reduced-form parameters $(\tilde{\rho}, \tilde{\nu})$.

The term $\hat{u}_i^{(1-\tilde{\rho})\tilde{\nu}}$ captures how a utility increase raises demand for location i . The summation term aggregates that effect across the pairwise nests in which i participates. Stronger migration linkages appear as larger calibrated \tilde{w}_{ij} values and therefore larger cross-effects.

In practice, this is the easiest way to use SPACE in a counterfactual. Calibrate $\tilde{\rho}$ and \tilde{w}_{ij} from migration data, solve the baseline equilibrium, and then map counterfactual utility changes into population changes with the formula above.

4 Conclusion

SPACE provides a tractable alternative to moving-cost-based migration models. Its main advantages are calibration from migration data, simple eigenvalue-based estimation, closed-form population responses, and cross-elasticities disciplined by observed migration flows.

One final practical point is important. Switching a dynamic migration model to SPACE changes not only the migration block, but also the macro dynamics of adjustment. In the baseline SPACE formulation, population adjusts much more quickly to permanent shocks

than it does in a moving-cost model. If an application needs slower transition dynamics, that persistence will usually have to come from another realistic feature of the model, such as housing or local capital adjustment.

We conclude with a checklist for users to verify they have followed all the steps of implementing the SPACE model calibration:

1. Start with bilateral migration flows and baseline population.
2. Symmetrize flows using the logarithmic mean \tilde{m}_{ij} .
3. Compute total migration m_i , baseline population p_i , and the matrix M .
4. Recover $\tilde{\rho}$ from the largest eigenvalue of M .
5. Recover the transformed pairwise weights \tilde{w}_{ij} from the associated eigenvector.
6. Choose $\tilde{\nu}$ using an elasticity target or other discipline from the surrounding model.
7. Recover primitive weights w_{ij} only if solving the population block in levels.
8. Use either the level formula or the hat-algebra formula in the QSM.

References

- Howard, Greg, and Hansen Shao.** 2026. “The Dynamics of Internal Migration: A New Fact and its Implications.” *The Review of Economic Studies*.
- McFadden, Daniel.** 1978. “Modelling the Choice of Residential Location.” In *Spatial Interaction Theory and Planning Models*, edited by Anders Karlqvist, Lars Lundqvist, Folke Snickars, and Jörgen W. Weibull, 75–96, Amsterdam: North-Holland.