The paper "**Optimizing earthquake phase association performance with semi-supervised learning**" is currently under **internal review**.

It will be published on this website soon. In the meantime, the document below provides a detailed report of the work conducted during this internship.

Summer 2025 Research Project (Unaite Challenge)

Academic year: 2024–2025

Optimizing earthquake phase association GNN model performance through bayesian optimization of a synthetic training data generator for semi-supervised learning

Author: Gabriel Dupuis
Stanford Supervisors: Ian W. McBrearty, Gregory C. Beroza

Github Repo:
https://github.com/imcbrearty/GENIE/tree/training_data_generation

Internship period: 26/05/2025 – 22/08/2025

Host organization: Stanford University, Department of Geophysics

Confidentiality: Non-confidential report

Abstract

Abstract

Earthquake phase associators are a critical component of the earthquake detection pipeline, responsible for determining when, where, and how many distinct seismic sources occurred, and for linking each seismic arrival to its corresponding source. Recent deep learning models such as GENIE have demonstrated promising performance [5], but their training relies on synthetic datasets that are not optimally tuned to real seismic network statistics. This project explores the development of a more physically inspired synthetic data generator and its calibration using Bayesian Optimization of key parameters controlling attenuation, missing-pick rates, inter-station correlations, and travel-time uncertainty. The calibrated generator is then used to retrain GENIE in a semi-supervised setting. Results show that this approach improves phase association performance and enhances catalog completeness for seismicity in Central California.

Résumé

Les algorithmes d'association de phases sismiques constituent un élément essentiel de la chaîne de détection des séismes. Ils sont responsables de déterminer quand, où et combien de sources sismiques distinctes se sont produites, ainsi que d'associer chaque arrivée sismique à sa source correspondante. Les modèles récents d'apprentissage profond, tels que **GENIE**, ont montré des performances prometteuses [5], mais leur entraînement repose sur des jeux de données synthétiques qui ne sont pas toujours ajustés de manière optimale aux statistiques réelles des réseaux sismiques.

Ce projet explore le développement d'un générateur de données synthétiques plus inspiré des lois physiques, ainsi que sa calibration par **optimisation bayésienne** de paramètres clés tels que l'atténuation, le taux de détections manquées, les corrélations inter-stations et l'incertitude sur les temps d'arrivée. Le générateur calibré est ensuite utilisé pour réentraîner GENIE dans un cadre semi-supervisé. Les résultats montrent que cette approche améliore les performances d'association de phases et augmente la complétude du catalogue pour la sismicité en Californie centrale.

Keywords: Seismology, Graph Neural Networks, Synthetic data, Bayesian Optimization, Semi-supervised learning

Acknowledgements

I sincerely thank my supervisors Ian W. McBrearty and Gregory C. Beroza for their guidance, scientific expertise, and availability throughout this internship. I am also grateful to ENSTA Paris for providing the opportunity to conduct this project at Stanford. Special thanks go to the members of the Seismology group for their support and kindness, and to the other interns for making this a very enriching experience.

Contents

List of Figures

1 Introduction

Earthquake catalogs are the fundamental records of when and where earthquakes occur. They are used to study faults, estimate seismic hazard, and inform early-warning systems. Building these catalogs from continuous ground-motion recordings involves two steps: (*i*) each seismic station marks possible changes in the signal, producing *picks*; (*ii*) these scattered picks must be grouped into coherent events with an origin time, location, and magnitude. This second step—called *phase association*—is difficult because events can overlap, some stations miss detections, others report false alarms, and network coverage is uneven. Successful association therefore requires combining information across many sensors in a consistent physical framework.

Recent machine-learning models, such as **GENIE**, use Graph Neural Networks (GNNs) to learn patterns across the station network and assemble picks into events. However, training such models requires large labeled datasets, which are not available in practice. To compensate, synthetic picks are generated where the true event is known. If these synthetic picks fail to reflect reality, the trained model underperforms when applied to real data.

This internship focused on reducing that gap between synthetic and real picks for Central California. The work consisted of designing a more physically realistic synthetic data generator and calibrating it to observed pick statistics before retraining GENIE. The generator exposes interpretable controls—attenuation with distance, correlated successes or failures across stations, and timing uncertainty—which were optimized using Bayesian methods. The calibrated generator was then integrated into a semi-supervised training pipeline where GENIE benefits from both synthetic and real picks.

Contributions.

- A new generator design that accounts for anisotropy and inter-station correlations.
- A calibration procedure matching synthetic to real picks across magnitude ranges.
- Integration of the calibrated generator into GENIE's semi-supervised training workflow.

Report outline. **Chapter 2** reviews catalog building and learning-based association. **Chapter 3** presents the generator, calibration, and training pipeline. **Chapter 4** reports calibration results and model behavior before and after retraining. **Chapter 5** concludes with perspectives, followed by a personal assessment.

2 State of the Art

2.1 The earthquake phase association problem

When an earthquake happens, it sends out different kinds of waves that are recorded by many seismic stations. Each station marks the moment it thinks a wave has arrived, but these "picks" are just individual pieces of information. The challenge is to figure out which picks belong together and actually come from the same earthquake, while also determining when and where that earthquake occurred. This is called phase association.

It's not easy, because earthquakes can happen close together in time and space, stations sometimes miss arrivals or make mistakes, and the quality of the data varies across the network. A good association method needs to piece everything together in a way that makes physical sense, so that each event is located correctly and each wave arrival is linked to the right source.

2.1.1 Different characteristics of an earthquake

When studying an earthquake, we first need to look at what characteristics define an earthquake. There are a lot of characteristics but we will focus on 3 main characteristics here.

1. **The source of an earthquake**: Every given earthquake comes from a given source, which is characterized by its hypocenter (latitude, longitude, depth) and origin time, with the epicenter being the surface projection of the hypocenter. Sources can arise from tectonic fault slip, volcanic or geothermal processes, induced seismicity, or explosions. Studying the repartition of the sources can help us to identify ridges, faults, ... to delineate active structures and deformation zones, constrain fault geometry, and assess catalog resolution for subsequent analyses.

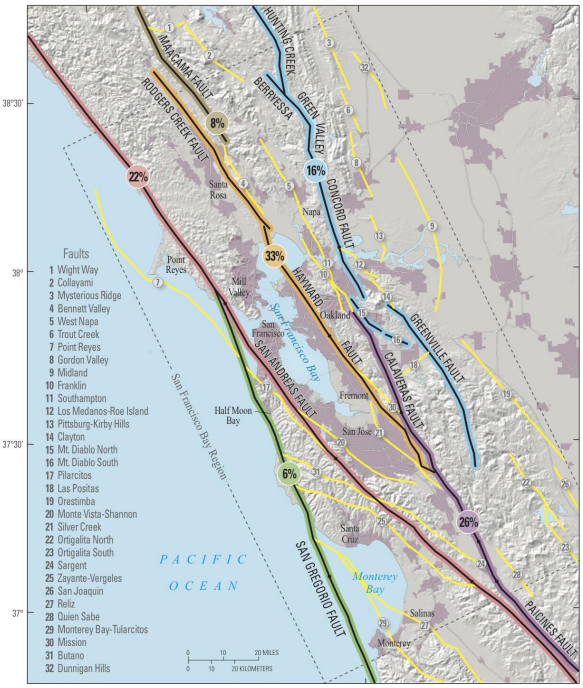

Figure 2.1: Known active geologic faults in the San Francisco Bay Region (source: USGS).

2. **The magnitude of an earthquake**: An earthquake is also defined by its magnitude, characterizing the amplitude and intensity of the earthquake. The magnitude of an earthquake is given by this formula $M_L = \log_{10}(A) - \log_{10}(A_0(\Delta))$, where A is the maximum ground-motion amplitude measured on a standard Wood–Anderson seismograph and $A_0(\Delta)$ is a distance-dependent reference amplitude (attenuation correction) as a function of epicentral distance Δ.

   The magnitude scale extends below zero, meaning negative magnitudes are also possible.

3. **The phase and velocity of the wave**: Each earthquake has different waves propagating in the ground.

   The first-one is the **p-wave** which is a compressional body wave with particle motion **parallel to the direction of propagation**; it is the fastest seismic phase and typically travels in the continental crust at about 5–8 km/s, propagating through both solids and fluids.

   The second one is the **s-wave** which is a **shear body wave** with particle motion perpendicular to the direction of propagation; it is slower than P-wave, with typical crustal speeds of about 3–4.5 km/s, and does not propagate in fluids. Something worth mentioning is that the velocity of the wave is given by its type but also by the **velocity model of the ground** that comes from regional 1D or 3D models derived from travel-time inversions and seismic tomography using observed picks and will be supposed already known in this study.

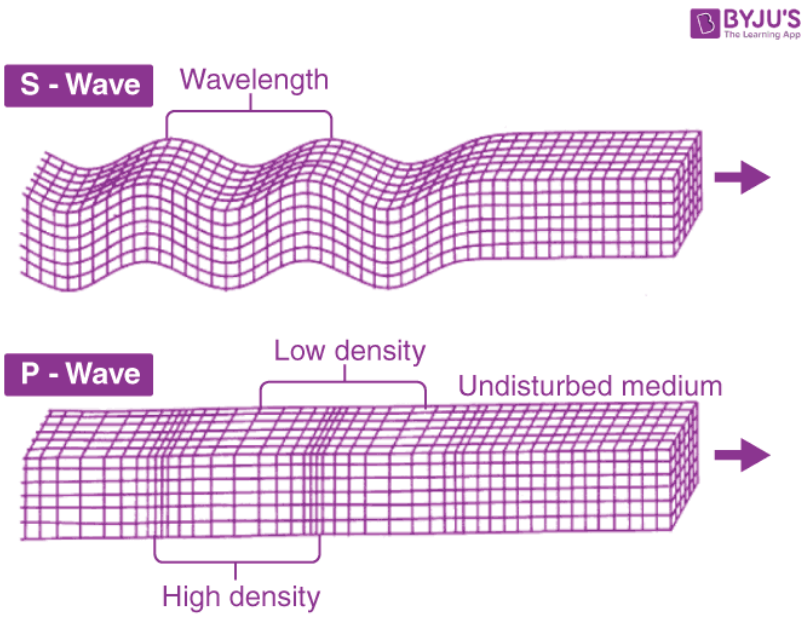

Figure 2.2: Schematic of P- and S-waves and their particle-motion directions (source: BYJU).

Now that we are aware of the different characteristics of an earthquake, the question is: how do we collect data on earthquake? More importantly, how can we use this data to create an earthquake **catalog** in a given region?

2.1.2 State-of-the-art to build a catalog of earthquake

Simplest response to the problem

To monitor the data, there is only one way: we use seismic stations placed around the globe that record ground motion with seismometers and digitizers, continuously sampling three-component waveforms and gives us waveform time series to analyze. We can then just look at the maximum of the signal to detect when an earthquake occurs. However, it is very hard from only the seismic stations signal to get informations about the phase of the earthquake, its source or how noisy is the signal. In practice, waveforms contain noise, site and instrument effects, and arrivals from multiple overlapping sources; manual detection is not scalable, and simple thresholding provides unreliable onsets [1, 2] and no phase labels. This motivates automatic pickers such as PhaseNet [12] that learn robust P/S onsets directly from continuous waveforms. That is why the first step will be to use a process called **phase picking**.

Phase picking and phase association

Phase Picking To extract information from the signal, the modern way to do it is to do a processus called phase picking, which consists of detecting the onset times of seismic phases in continuous waveforms and outputting a **list of timestamps** where earthquakes are observed. It is like a list of mark in time that tells us when an earthquake is happening at the station. Traditional methods relied on well-designed filters or mathematical operations applied to waveform arrays but modern methods such as PhaseNet now use deep learning to learn and extract the picks. [7, 12]

After doing phase picking, we now have a **extremely large amount of picks/marks** for each station giving information about when a wave of an earthquake was seen. Therefore, the

question is now, how do we use and aggregate the information to associate each point to a source, a magnitude, and a phase of an earthquake wave? And knowing that so many earthquakes are active simultaneously, how to distinguish that scrambled data to make a coherent catalog at the end? That is where the fun begins, this processus is called **phase association** and is the problem that the model I worked on, **GENIE**[5], tries to solve (see Figure 2.3).

Phase association

Traditional methods include back-projection, probabilistic associators, clustering, and Bayesian mixtures [13]. For example, backpropagation aligns moveout curves across stations to score candidate sources. A detailed formulation is provided in Appendix A.1.

But the GENIE model **differs from these approach**, and uses deep learning and graph neural networks to construct a catalog from picks given by PhaseNet, using the spatial information of the stations to aggregate information.

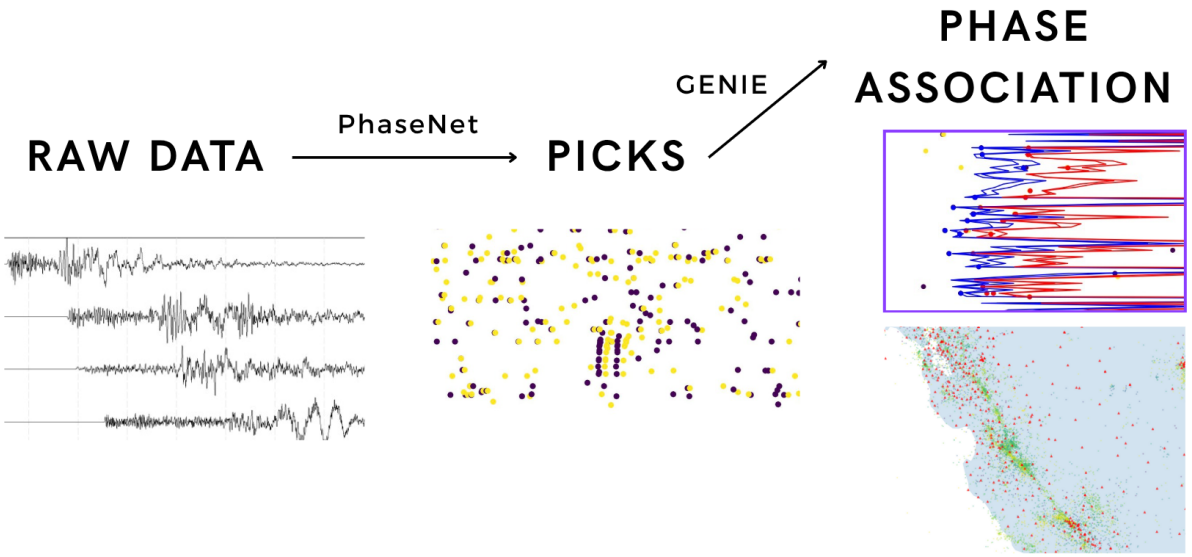

Figure 2.3: Steps required to build an earthquake catalog from continuous waveforms: picking, association, location, and catalog generation.

2.2 Deep learning approaches explanation

In practice we train a parametric model f_θ to map inputs to outputs and update the parameters θ to reduce a loss. Concretely,

$$\theta^* = \arg\min_\theta \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\big(f_\theta(\mathbf{x}_i),\, \mathbf{y}_i\big).$$

For GENIE, $\mathbf{x}$ contains PhaseNet picks [5, 12] with the station and spatial graphs; $\mathbf{y}$ encodes a space–time source likelihood and pick–source association probabilities. Training minimizes a supervised loss on labeled data and, when available, adds a consistency or pseudo-label term on unlabeled real picks.

A **graph neural network** (GNN) is a neural network where features live on the nodes of a graph and, at each layer, nodes exchange information only with their immediate neighbors [3]. One round of exchanging information is called **message passing**; stacking several rounds lets information travel farther across the graph (Figure 2.4).

Consider a social network where each node (person) stores their number of comments. To estimate the global mean, each person repeatedly replaces their value with the average of their neighbors' values; this iterative averaging converges to the network mean.

The general message–passing update can be written as

$$\mathbf{h}_i^{(\ell+1)} \;=\; \psi\Big(\mathbf{h}_i^{(\ell)},\; \mathrm{AGG}_{j\in\mathcal{N}(i)}\,\phi\Big(\mathbf{h}_i^{(\ell)},\, \mathbf{h}_j^{(\ell)},\, \mathbf{e}_{ij}\Big)\Big)\,,$$

where $\mathbf{h}_i^{(\ell)}$ is node i's feature at layer ℓ, $\mathcal{N}(i)$ are neighbors, $\mathbf{e}_{ij}$ are optional edge features, and ϕ, ψ are small shared networks; AGG is a permutation-invariant aggregator (e.g., sum or mean).

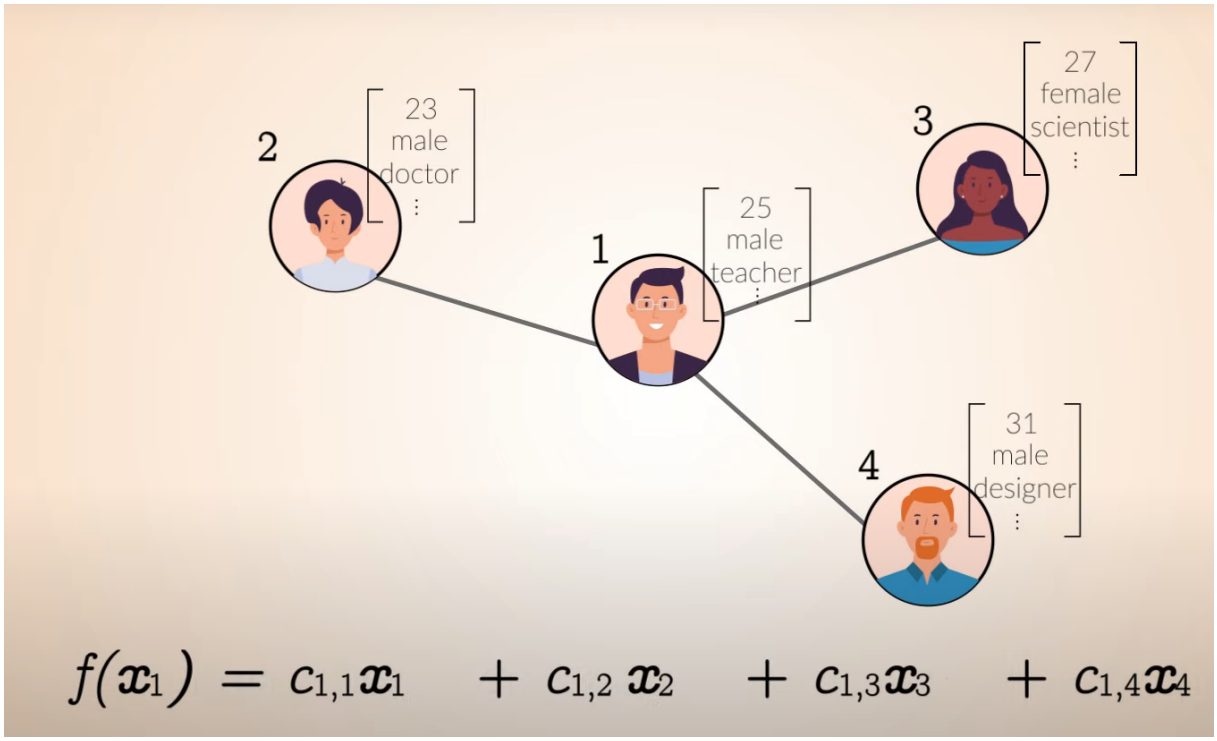

Figure 2.4: Toy example of message passing on a social graph used to illustrate neighborhood aggregation.

This mechanism **aggregates information over the network**. In GENIE, we build a station graph S and a spatial graph X, then operate on their Cartesian product $S \times X$. Each node represents a (station, candidate source) pair, and messages flow along nearby stations and nearby sources to favor moveout-consistent patterns that indicate an event.

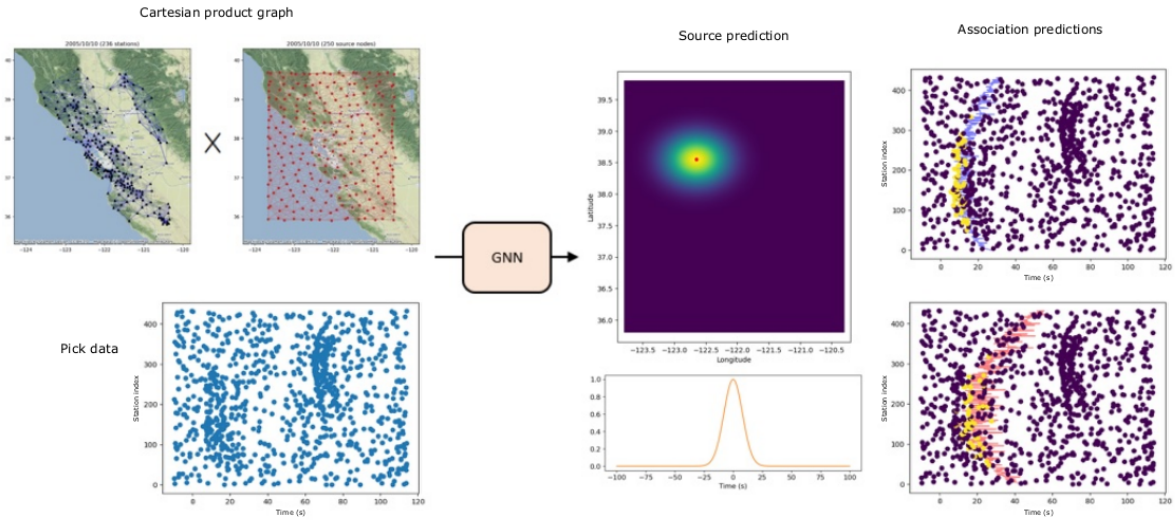

Figure 2.5: Schematic of GENIE: message passing on the product graph $S \times X$ to infer source likelihoods and pick–source associations.

Semi-supervised learning Semi-supervised learning is a machine learning paradigm that combines a small amount of labeled data with a large amount of unlabeled data to improve generalization. In practice, this is useful when high-quality labels are scarce or costly to obtain, as in seismology where manual phase association is time-consuming and incomplete. A common strategy is *pseudo-labeling*, where confident predictions on unlabeled samples are treated as temporary labels and used to retrain the model, gradually transferring knowledge from synthetic or labeled data to real, unlabeled data [4]. In earthquake monitoring, this approach has been shown to improve both phase picking and association by leveraging abundant unlabeled picks while retaining consistency with physical constraints [10].

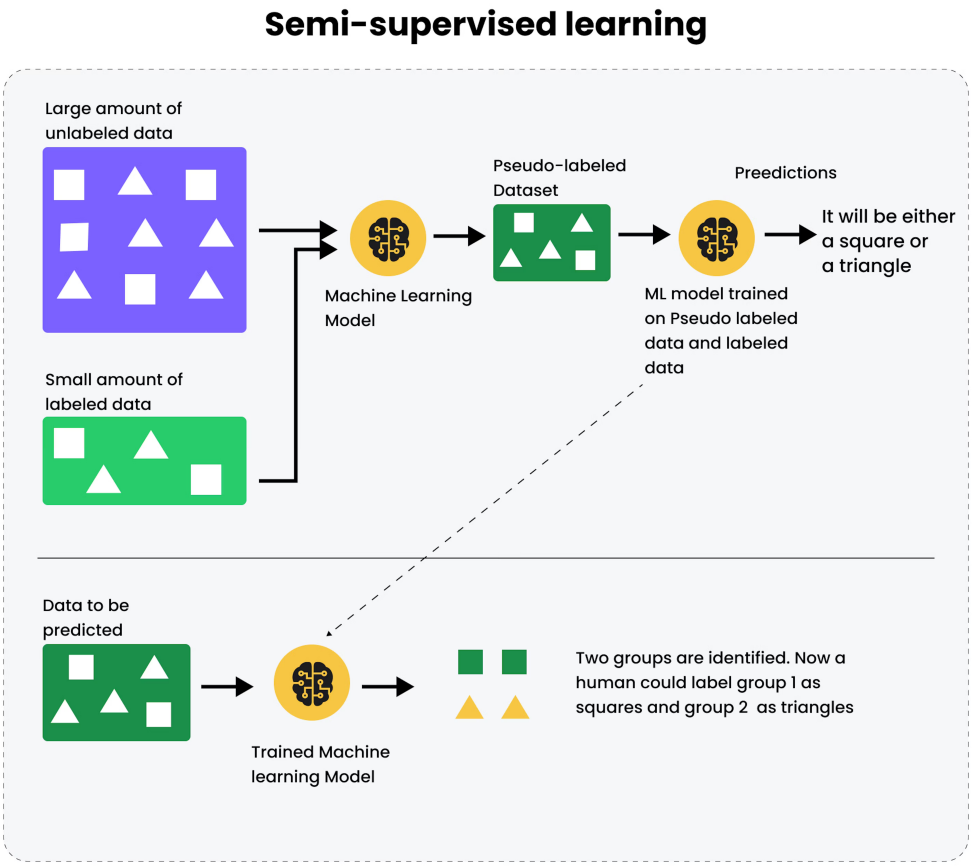

Figure 2.6: Illustration of semi-supervised learning with pseudo-labeling: (1) train on labeled synthetic data, (2) predict on real unlabeled data, (3) select high-confidence predictions as pseudo-labels, and (4) retrain the model jointly on both. Source: MadData

2.3 How GENIE works and explanation of the synthetic data

GENIE ingests per-station pick sets and the station/spatial graphs, performs message passing on $S \times X$, and outputs (i) a source likelihood field and (ii) pick–source association probabilities. Large-scale fully labeled real datasets are not available (catalogs are incomplete and phase labels can be inconsistent), so we rely on synthetic data where ground truth is known and then adapt the model with semi-supervised fine-tuning.

We generate synthetic data in three steps:

1. spatial generation — sample event locations and origin times within the network footprint;
2. pick generation — compute theoretical travel times with a fixed velocity model and simulate detections per station using distance–magnitude attenuation and coverage, with small timing jitter;
3. noise addition — add false positives and drop a fraction of true picks with phase-dependent uncertainty and mild inter-station correlation.

The part we will focus on in this paper is the spatial generation. The old method was using an arbitrary d_max from which all the stations inside d_max circle to the source

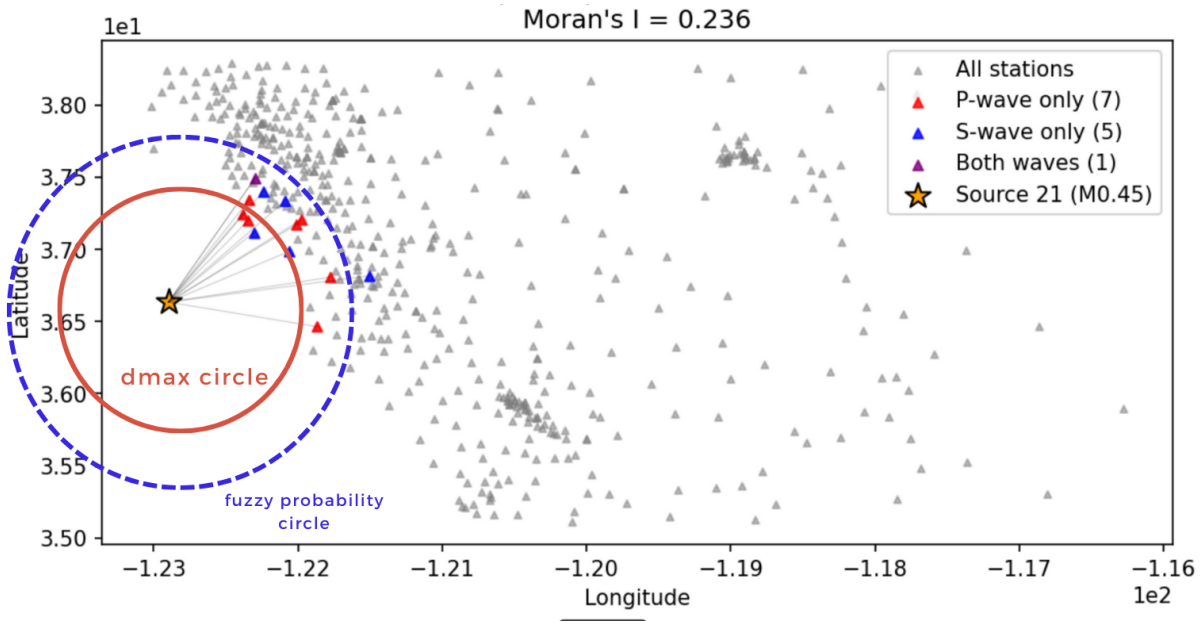

Figure 2.7: Old method of synthetic data picks generation

are selected. The stations outside were selected stochastically if they were part of the "fuzzy probability threshold".

Old spatial sampling and independence assumptions produce unrealistically uniform source distributions and station coverages, weak clustering along known faults, and insufficient correlation in miss/false patterns across nearby stations. This mismatch reduces transfer to real picks. In summary, this method was not flexible or realistic at all and that's what I tried to improve.

To reduce this gap, I implemented:

1. a more realistic, tunable synthetic generator that can be adapted to different regions,
2. Bayesian optimization to fit generator hyperparameters to real pick statistics,
3. a semi-supervised training loop that re-trains GENIE with the calibrated generator so the model improves over time.

3 Methods and Materials

3.1 Synthetic data generator

3.1.1 Radial function

Problem to solve

We seek a probability function that assigns, for each station in a fixed network, the probability of being selected as observing a phase given the event location and magnitude.

The function should:

1. Approach 1 near the source and decay after a distance σ_{radial} to mimic the impact of the noise compared to the magnitude of an earthquake
2. Have $\sigma_{radial} = g(magnitude)$
3. Select a number of stations roughly proportional to event size

Formally, station selections are Bernoulli with $X_k \sim \text{Bernoulli}(p_k)$ and $p_k = f(\text{radius}, \text{magnitude})$.

Solution

We adopt a radial probability function $f(r)$ that depends on the source–station distance r and a magnitude-dependent shape parameter p. To keep the 3σ scale comparable across p, we use

$$f_{\sigma_r}^{(p)}(r) \;=\; \exp\!\left(-\frac{r^{2p}}{2\,9^{\,p-1}\,\sigma_r^{\,2p}}\right),$$

which maintains $f_{\sigma_r}^{(p)}(3\sigma_r) \approx 0.01$ for all integer $p \geq 1$. H

Optionally, a scaling factor $\alpha \geq 0.9$ is applied to prevent f from saturating at 1 in the center and allow for tunable miss_pick: $p_r = \alpha\, f_{\sigma_r}^{(p)}(r)$.

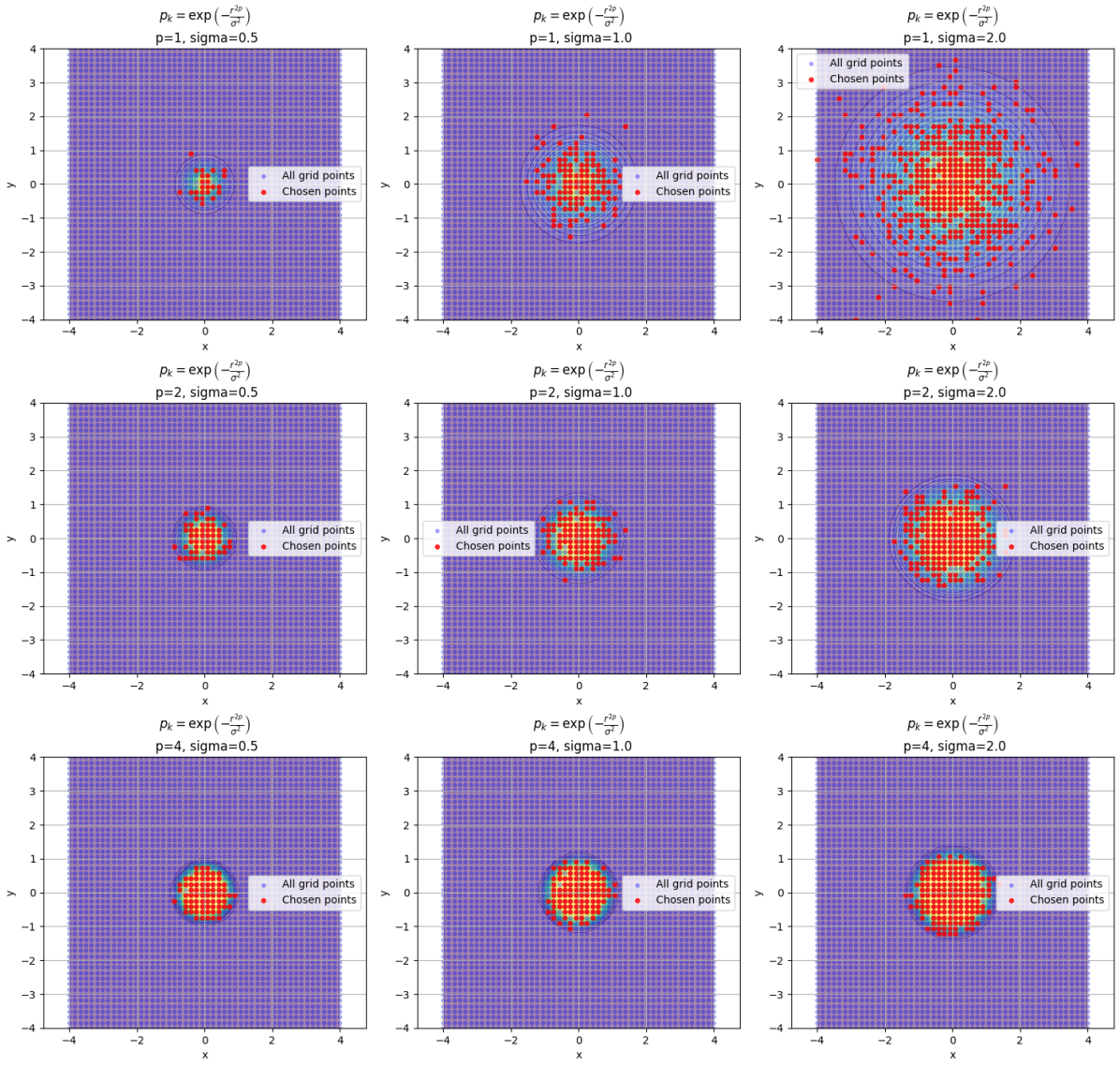

Figure 3.1: Radial selection probability as a function of source–station distance r for a given magnitude-dependent shape p.

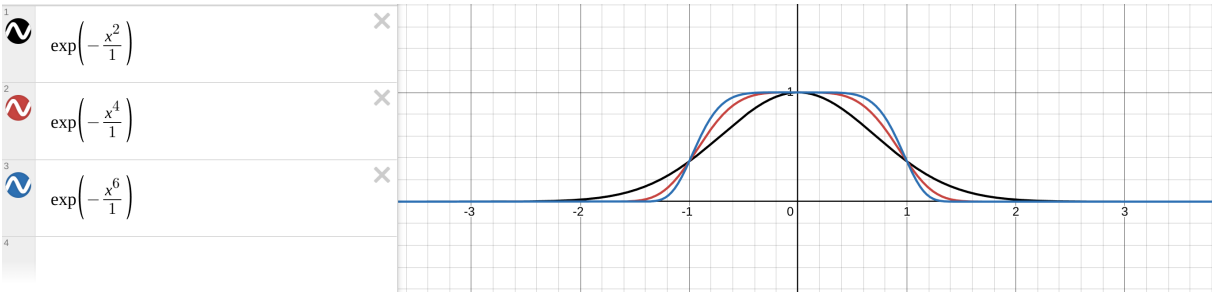

Figure 3.2: One-dimensional slices of $f_{\sigma_r}^{(p)}(r)$ for several exponents p, showing sharper decay for larger p.

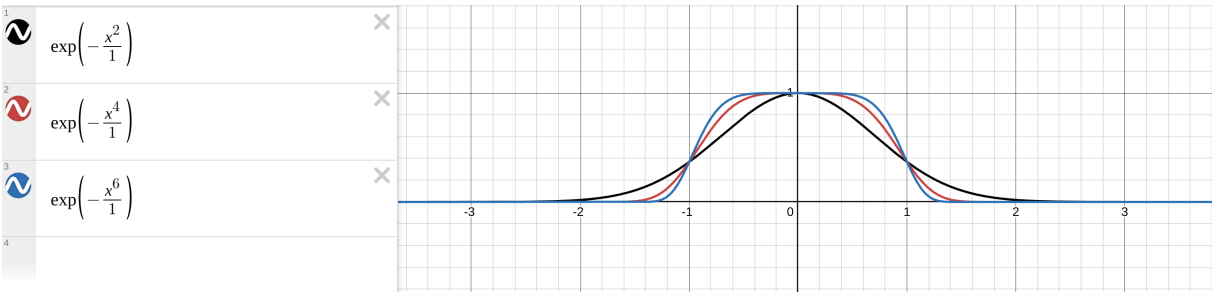

Figure 3.3: Same as Figure 3.2, highlighting the common 3σ scale across different p.

Elliptical part

To account for anisotropy and radiation pattern, we replace the Euclidean distance by a Mahalanobis distance defined by a positive-definite covariance Σ and center c:

$$r^2 \;=\; (x-c)^\top \Sigma^{-1}(x-c), \qquad p(x) \;=\; \exp\!\left(-\frac{r^2}{\sigma^2}\right).$$

The ellipse $(x-c)^\top\Sigma^{-1}(x-c) \le 1$ has semi-axes $\sqrt{\lambda_1}, \sqrt{\lambda_2}$ and orientation given by the eigenvectors of Σ. In practice one can sample a rotation θ and semi-axes ℓ_1, ℓ_2, form $R(\theta)$, $D = \mathrm{diag}(\ell_1, \ell_2)$, then set $\Sigma = RD^2R^\top$. We can basically choose the orientation and eccentricity of the ellipse

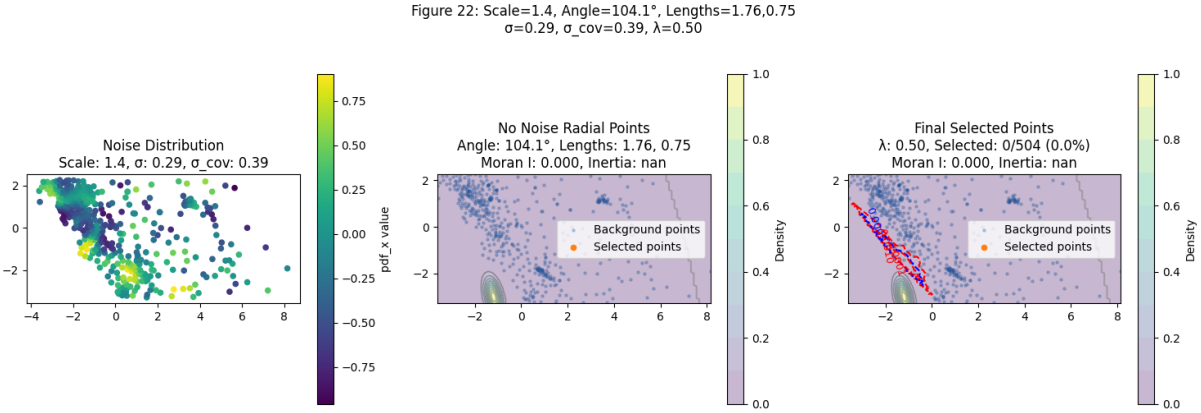

Figure 3.4: Example of elliptical anisotropy: station-selection probability elongated along a preferred direction.

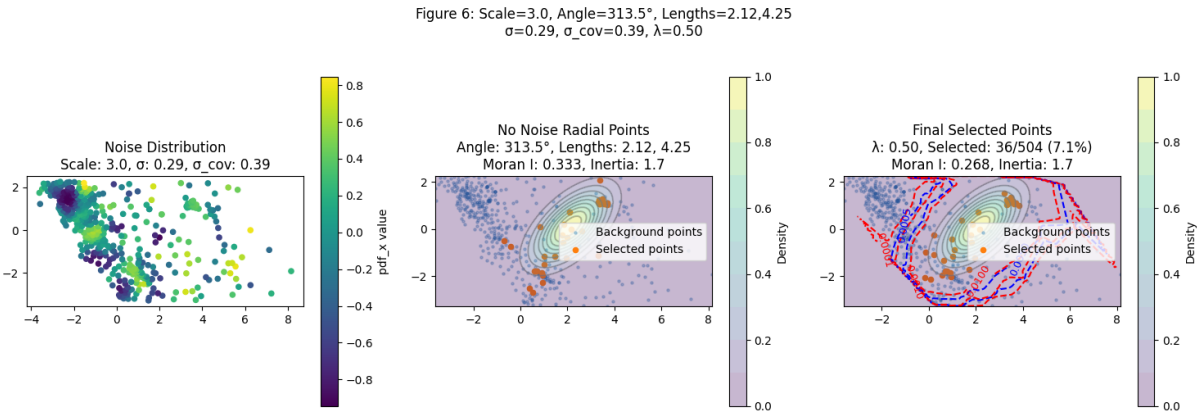

Figure 3.5: Example with a different ellipse orientation and eccentricity illustrating tunable anisotropy.

3.1.2 Noise

First iteration

Problem Most of the time, when stations don't detect an earthquake, it is because of the unknown distribution of the ground, so there is a high likelihood that they are close to each other. Therefore, to reproduce spatial correlations in missed and false picks, we introduce spatially correlated noise fields instead of i.i.d. perturbations.

Solution

Cholesky sampling We define a "covariance distance matrix" between stations i, j by a radial kernel

$$\Sigma_{ij} \;=\; \exp\!\left(-\frac{d(i,j)^2}{\sigma_{\text{cov}}^2}\right),$$

compute its Cholesky factor L ($\Sigma = LL^\top$), draw $z \sim \mathcal{N}(0, I)$, and set $\varepsilon = Lz \sim \mathcal{N}(0, \Sigma)$.

Logistic mapping To transform this noise between [-$3\sigma_{noise}$, $3\sigma_{noise}$] to a probability function we map ε to $[0, 1]$ via a logistic transform that defines the **"harshness"** of the noise with $\sigma_{logistic}$.

$$p_\varepsilon(x, y) \;=\; \frac{1}{1 + \exp\!\big(-\varepsilon(x, y)/\sigma_{\text{logistic}}\big)},$$

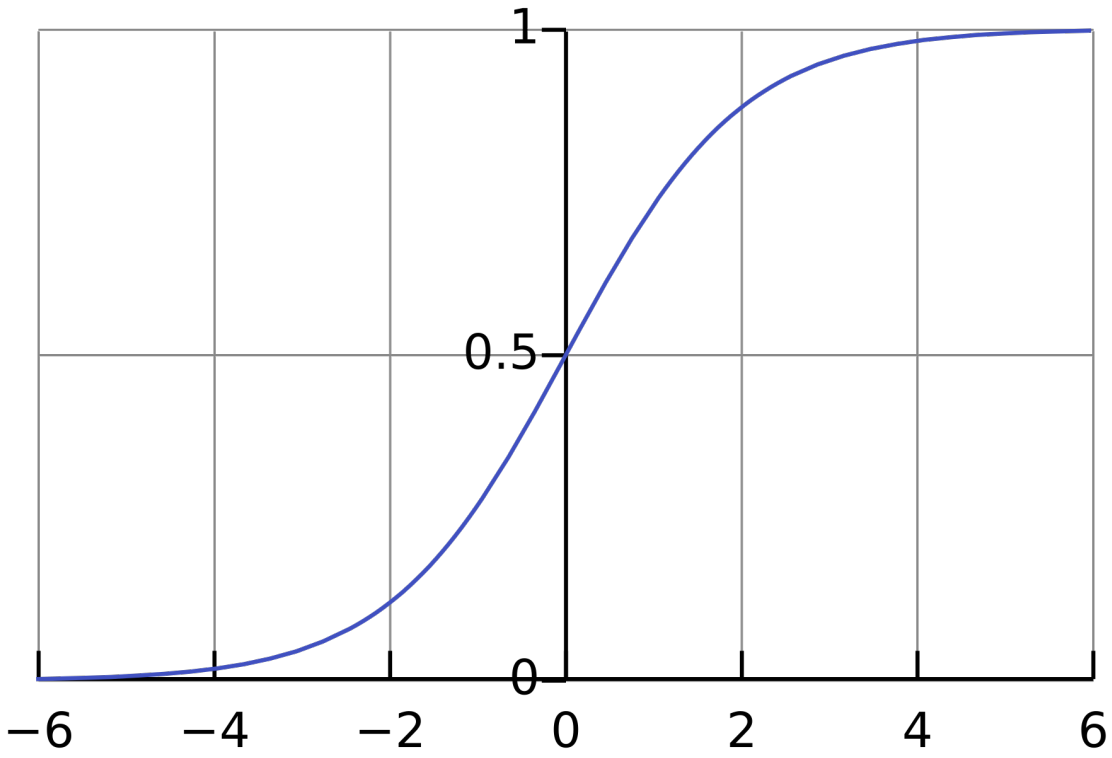

Figure 3.6: Logistic mapping p_ε used to convert correlated Gaussian noise into station-selection probabilities (example with $\sigma_{\text{logistic}} = 1$).

Sum Finally, we combine with the radial probability p_r using a clipped sum of the noise probability and the radial probability.

$$p_{\text{tot}} \;=\; p_r \;+\; [p_r > 0.02]\lambda_{noise}\, 2\big(p_\varepsilon - 0.5\big)$$

ensuring $p_{\text{tot}} \in [0, 1]$ and controlling the noise influence by λ_{noise}.

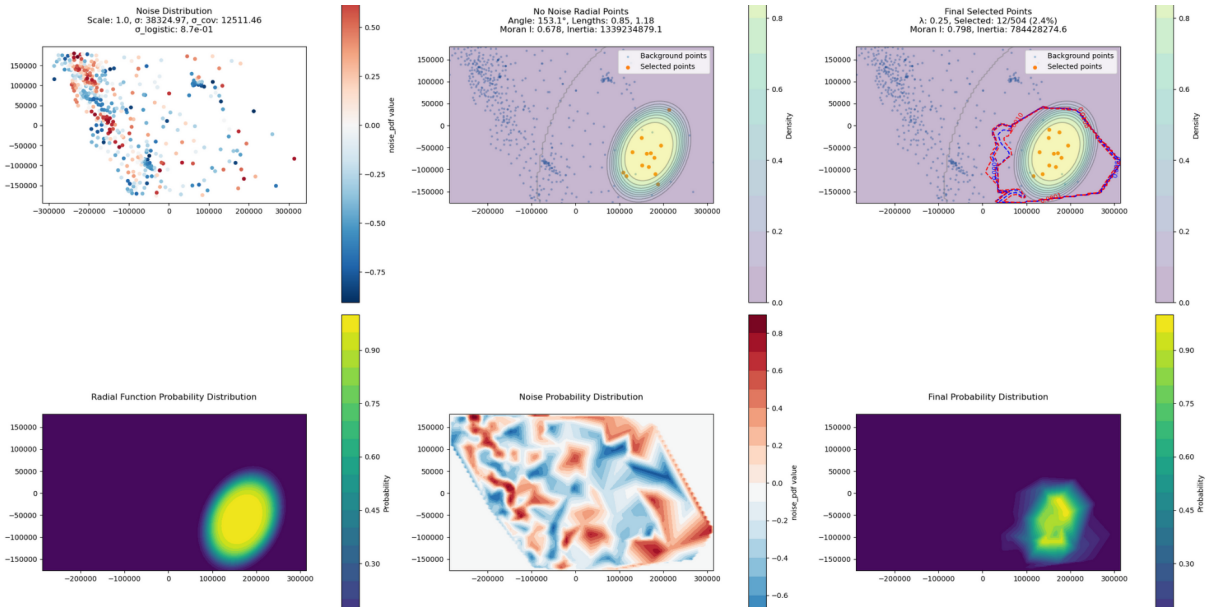

Figure 3.7: Plot of the noise, radial function, summed probabilities and picked stations on top

Second iteration

Issue with last method The logistic function is not interpretable and we want to simplify the problem to make it more flexible and avoid tradeoff. The sum function is also really arbitrary and we want to limit this choice.

New method Now instead of using the correlated noise on the data as a way to perturb the **probability** we want instead to perturb **the considered distance from the points to the source** which means that we pass perturbed points to the radial function. On top of that, we normalize the distance when passing to the Mahalanobis distance so the noise doesn't have to be adapted in any way.

We still use λ_{noise} to scale the noise.

This is what we obtain (Figure 3.8).

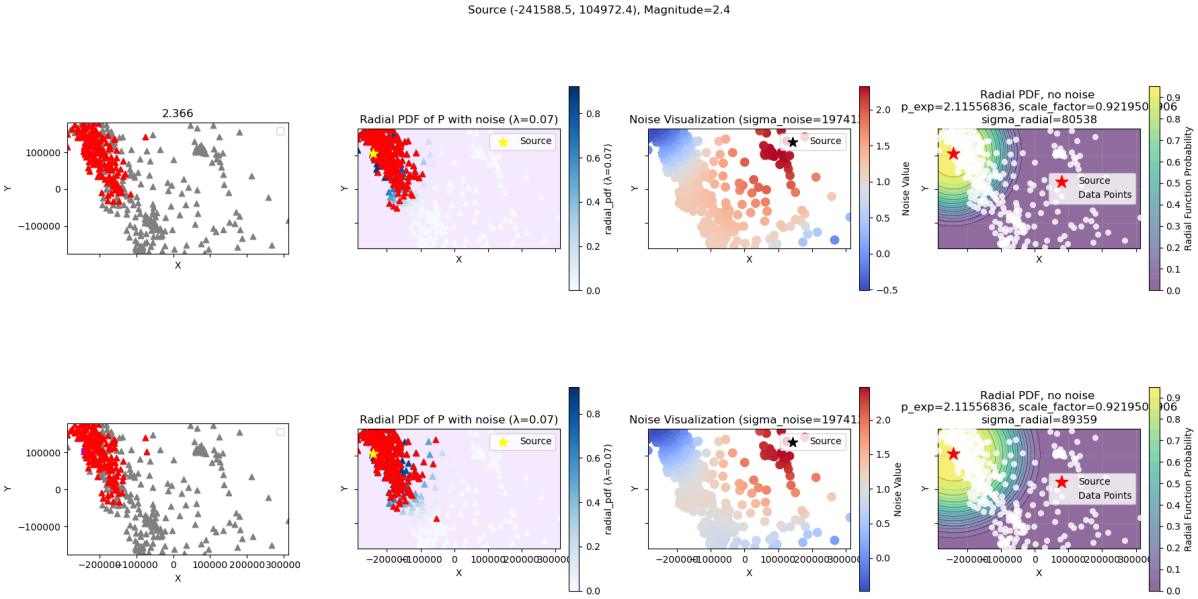

Figure 3.8: Comparison between real and synthetic spatial station selections (real on the very left, synthetic after that with p-wave on top and s-wave below) after switching to distance perturbations before evaluating the Mahalanobis-based probability.

3.2 Calibration via Bayesian Optimization

3.2.1 How does the Bayesian Optimization works?

Instead of manually tuning generator parameters, we use Bayesian Optimization (BO) to search for settings that make synthetic picks better match real ones. BO is well-suited to low-dimensional, expensive objectives. A simplified description is given here; full details of the acquisition function, evaluation metrics (inertia, Moran's I, overlaps), and objective are provided in Appendix A.2.

We use BO to optimize the parameters of an objective function that quantifies the distance between synthetic data and the best generated catalog so far. The goal is to compare two spatial point distributions and measure how close they are.

3.2.2 Evaluation metrics for the objective function

To define the objective, we first specify metrics that characterize the distribution of 2D points (selected stations) in space.

Inertia

Let $\mathcal{S} = \{s_i\}_{i=1}^{n} \subset R^2$ be selected station coordinates and $\bar{s} = \frac{1}{n}\sum_i s_i$ their centroid.

The (unnormalized) spatial inertia is

$$\mathcal{I}(\mathcal{S}) \;=\; \sum_{i=1}^{n} \left\| s_i - \bar{s} \right\|_2^2,$$

and the mean inertia is $\mathcal{I}/n$. We compute this for P picks and S picks separately, and optionally for their union, to capture dispersion relative to the epicentral region.

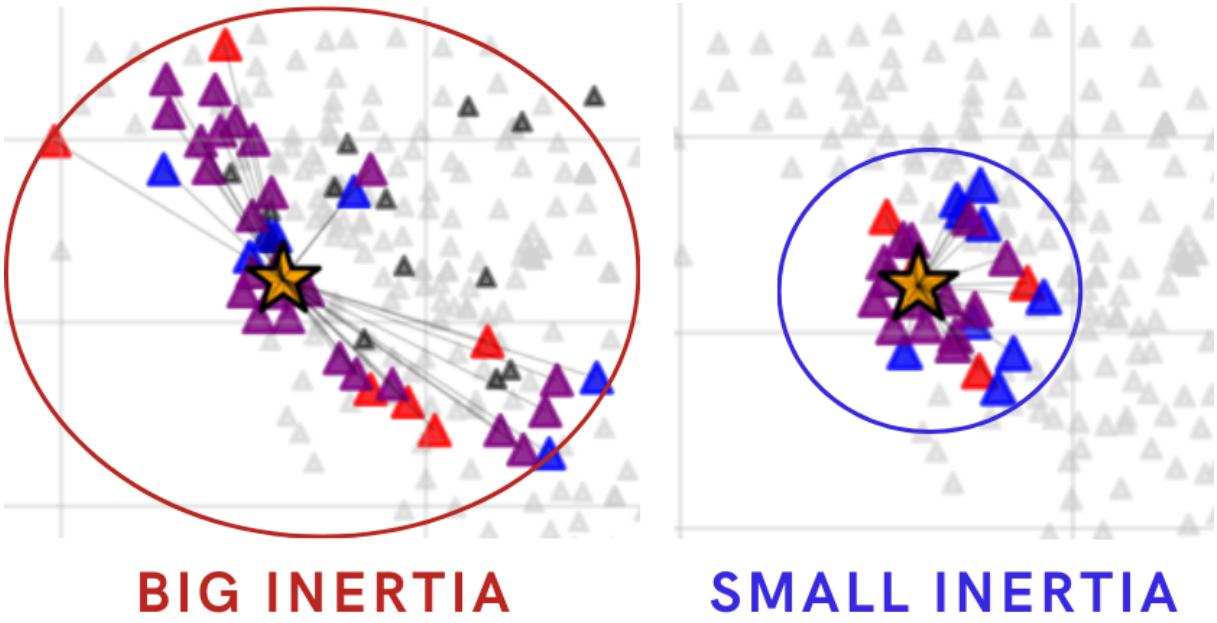

Figure 3.9: Examples of small and large spatial inertia for selected-station sets relative to their centroid.

Moran's I

Given binary indicators $x_i \in \{0, 1\}$ of station selection on a fixed station set and a spatial weight matrix $W = [w_{ij}]$ (e.g., $w_{ij} = \exp[-d(i, j)/\ell]$ with $w_{ii} = 0$), Moran's I measures spatial autocorrelation:

$$I = \frac{n}{\sum_{i \neq j} w_{ij}} \frac{\sum_{i \neq j} w_{ij} (x_i - \bar{x})(x_j - \bar{x})}{\sum_i (x_i - \bar{x})^2}, \quad \bar{x} = \frac{1}{n} \sum_i x_i.$$

Values $I > 0$ indicate clustering (nearby stations tend to co-select), $I < 0$ indicates dispersion.

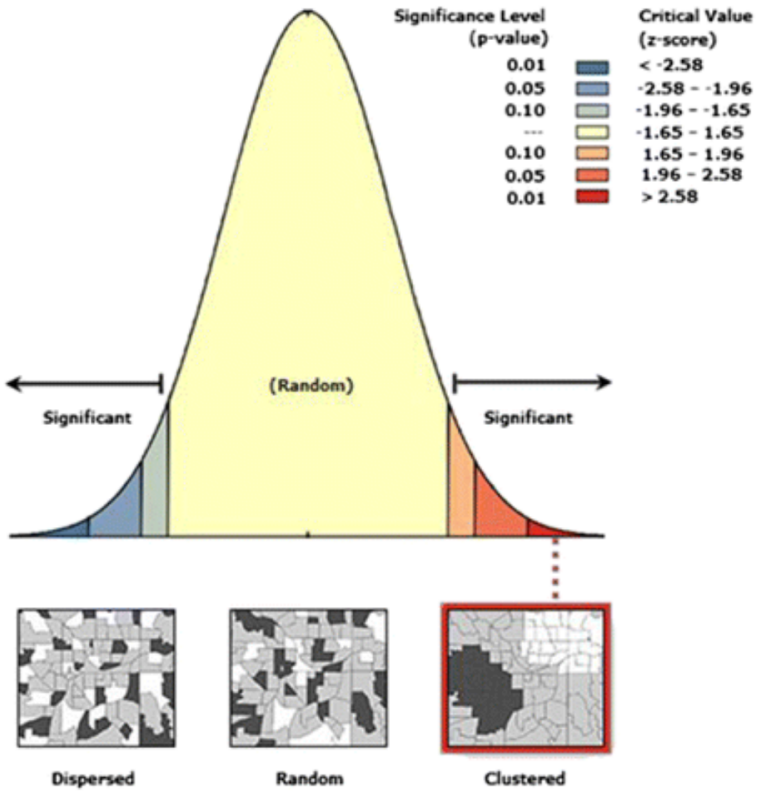

Figure 3.10: Example distributions illustrating positive and negative spatial autocorrelation as measured by Moran's I.

Statistics metrics

On top of inertia and Moran's I, we also track:

- **number of P-wave picks**
- **number of S-wave picks**,
- **size of the intersection between P and S selected stations**.

To compare a real event R and a synthetic event S, we apply set operations on selected station sets $A_P(R), A_S(R)$ and $A_P(S), A_S(S)$:

- intersection size $|A(R) \cap A(S)|$,
- symmetric difference size $|A(R) \,\Delta\, A(S)|$.

These can be computed on P only, S only, or the union, enabling 1:1 comparisons for events with the same source and magnitude.

3.2.3 Magnitude bins

We compare events from the real catalog to synthetic events generated at the same source and magnitude. Because magnitudes are not uniformly distributed in the real catalog, we sample real events using a magnitude-stratified scheme (uniform over predefined bins). Since station selection patterns depend strongly on magnitude, we evaluate metrics within **magnitude bins** to avoid bias and local optima.

3.2.4 Evaluation function

The evaluation function is the sum oof mean metrics over each magnitude.

$$f(\text{params}) \;=\; \sum_{i=1}^{N_{\text{metrics}}} \text{mean}_{m\in\text{bins}} \left(\frac{\left|\text{metric}_i^{\text{real}}(m) - \text{metric}_i^{\text{syn}}(m;\,\text{params})\right|}{\text{metric}_i^{\text{real}}(m) + \varepsilon} \right),$$

with a small $\varepsilon > 0$ for numerical stability. BO seeks parameters that minimize f.
The parameters to optimize are:

```
## Set Cholesky parameters
chol_params = {}
chol_params['p_exp'] = 2.087                    # Radial function exponent (fixed integer)
chol_params['miss_pick_rate'] = 0.07              # Miss pick rate (scale_factor = 1 - miss_pick_rate)
chol_params['sigma_noise'] = 98127                # Sigma noise for cluster spreading (in meters)
chol_params['lambda_noise'] = 0.168                # Correlation between radial function and noise
chol_params['radial_factor_p'] = 1.0     # P-wave detection radius factor (before division)
chol_params['radial_factor_s'] = 0.75      # S-wave detection radius factor (before division)
chol_params['radial_perturb_factor'] = 0.03    # Perturbation factor for ellipse parameters
chol_params['angle_perturb'] = 0.17    # Perturbation factor for ellipse parameters
chol_params['axis_perturb_factor'] =  0.06    # Perturbation factor for ellipse parameters
```

Figure 3.11: Hyperparameters tuned by the Bayesian optimizer for the synthetic generator.

This shows the distributions obtained before and after optimization.
Before:

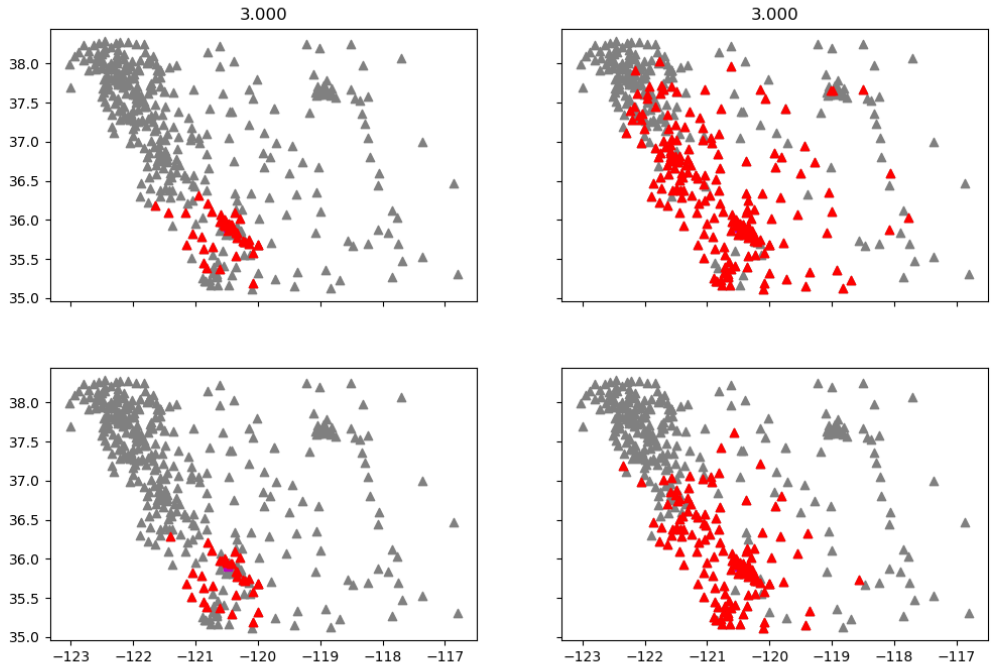

Figure 3.12: Real vs synthetic event comparison **before** calibration: P (top-left), S (bottom-left) for real; P (top-right), S (bottom-right) for synthetic with matched source and magnitude.

After:

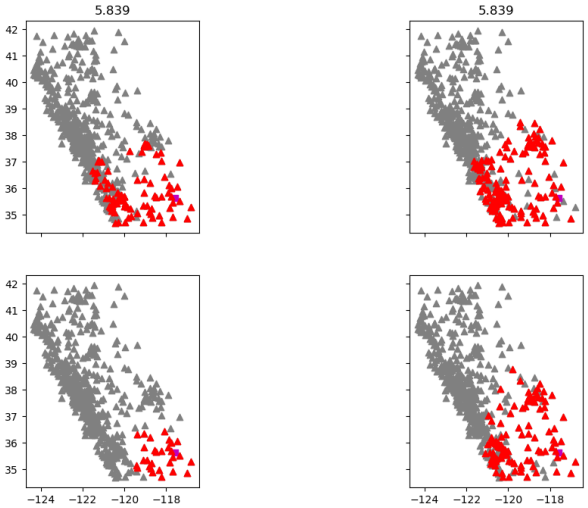

Figure 3.13: Real vs synthetic event comparison **after** calibration (example 1) with matched source and magnitude for P and S selections.

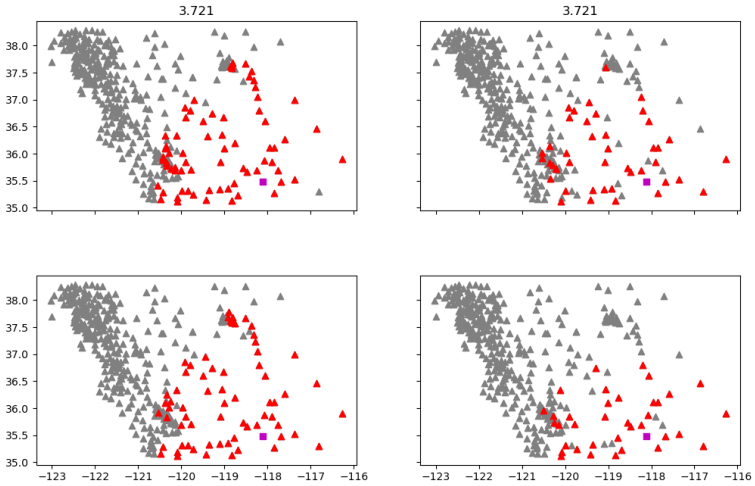

Figure 3.14: Real vs synthetic event comparison **after** calibration (example 2) with matched source and magnitude for P and S selections.

The next step is to do a **3-round optimization procedure** that also optimizes additive time noise applied to picks produced from the 2D spatial selection.

3.3 Semi-supervised training of GENIE

Once we have a calibrated synthetic generator and a way to align it with the best catalog, we build a pipeline that keeps improving the model.

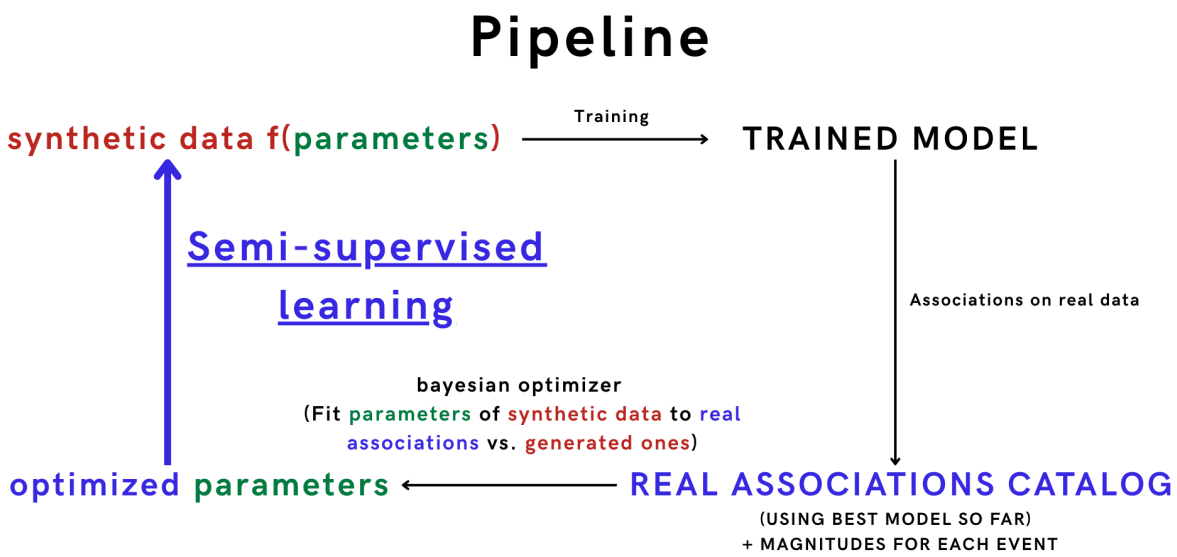

Figure 3.15: Semi-supervised training pipeline: calibrate generator → retrain GENIE on calibrated synthetic plus high-confidence real associations → re-infer and harvest new labels.

This pipeline is **semi-supervised**: we use the best catalog generated so far to calibrate the synthetic generator and then retrain GENIE, i.e., **pseudo-labeling** [4]. Predictions on real data provide structure that reflects actual pick statistics; calibrated synthetic data narrows the domain gap. For now, we focus on improving the generator via optimization before closing the loop. After these upgrades, we expect the closed loop to further improve performance, consistent with prior self-training approaches in phase picking (e.g., Tonga [10]).

4 Results and Discussion

4.1 Generator calibration

When training a model, the first thing we want to check is the quality of the training data. For that, we plot for each event the picks generated (both temporal and spatial but my work focused only on the spatial aspect) and look at three things.

- Is there any picks isolated from the rest of the picks?
- Are the generated picks coherent with the given magnitude?
- Is the difference between p and s-wave coherent with the radiation pattern we know (Cf. Chapter 1)?

When looking at the figures below, we see that those 3 criteria are respected (Figures 4.1 and 4.2).

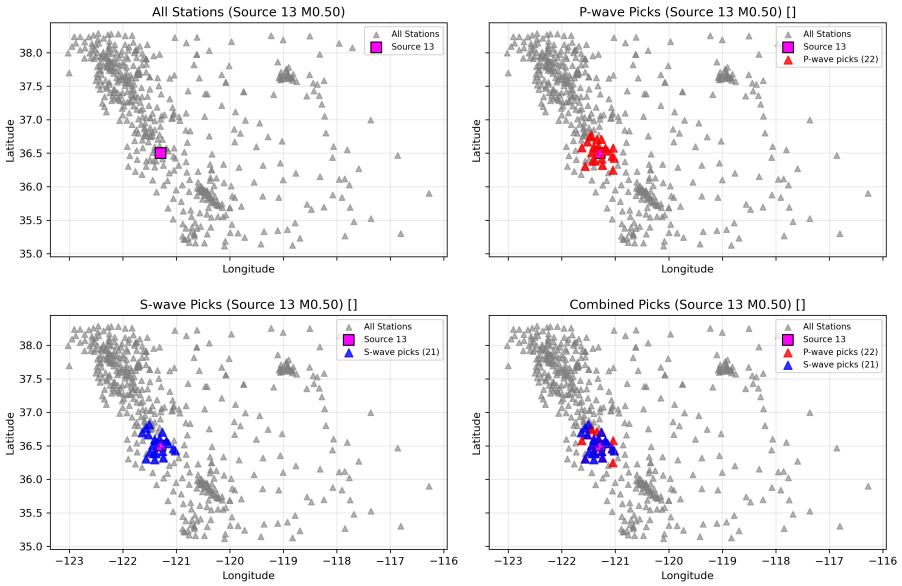

Figure 4.1: Example training event showing station selections and P/S differentiation used for qualitative checks.

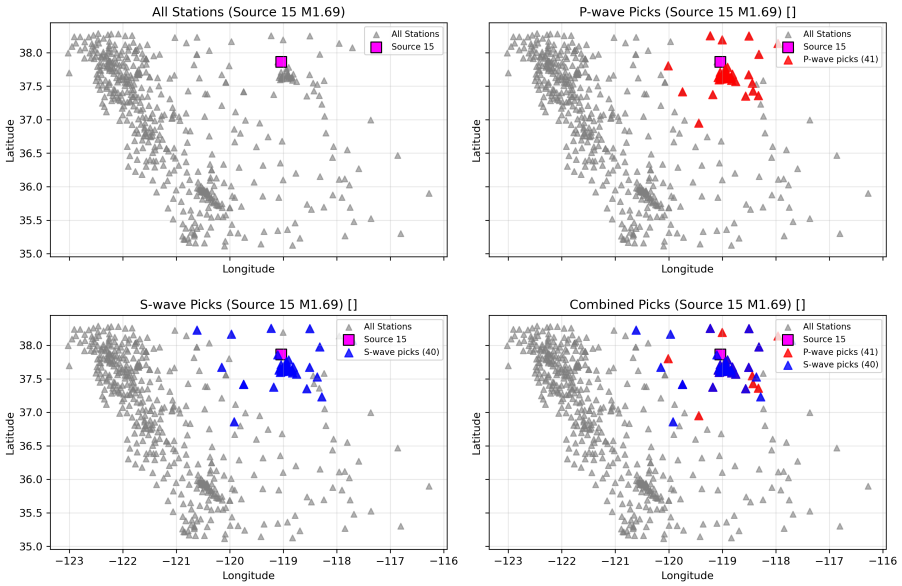

Figure 4.2: Another training example illustrating coherence between magnitude, coverage, and P/S patterns.

4.2 Results of the new model compared to the best one

4.2.1 GR Curves and number of detected earthquakes

One important way to assess the quality of the model is to compare the catalog to the USGS Catalog [8, 9], the U.S. Geological Survey's national earthquake catalog, which compiles analyst-reviewed origin times, locations, and magnitudes reported by regional and national seismic networks, and serves as a reference dataset for the United States. Using that Catalog, we can compare the number of events in Central California we detect by magnitude to the strongest existing Catalog in the region. This is what we obtain (Figures 4.3 and 4.4).

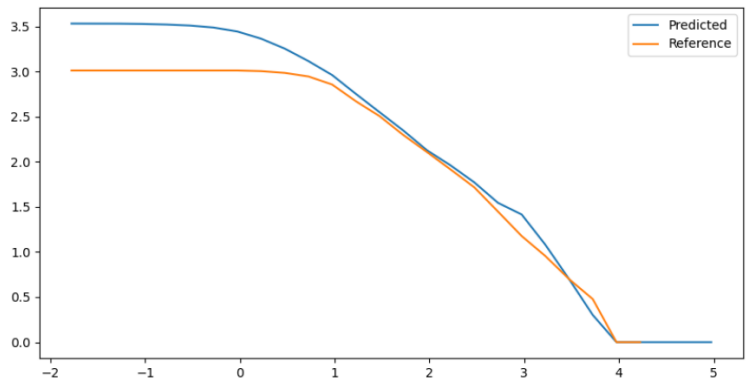

Figure 4.3: Cumulative number of detected events versus magnitude in Central California: comparison between GENIE-derived catalog and USGS catalog (panel 1).

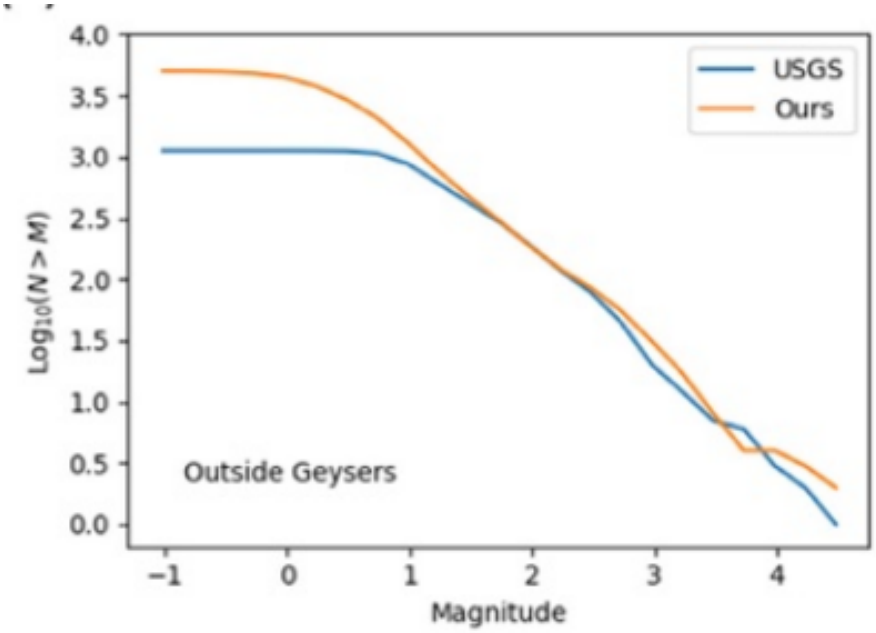

Figure 4.4: Cumulative number of detected events versus magnitude in Central California: comparison between GENIE-derived catalog and USGS catalog (panel 2).

The important thing to look at is how close is the number of big magnitude detected events. Because these are "easy" to detect, we know USGS catalog have everything and we don't want to have any offset in high magnitudes. We see here that the GENIE model curve match the USGS curve in high magnitude so it is very good.

On top of that, we see that the GENIE model detects way more small events, compared to the USGS catalog and in fact improves it.

Finally, we have better results with our last model since we compute only one year of data while outperforming the USGS catalog.

4.2.2 Source location

Another important figure for assessing the quality of our model is the placement of all the sources detected over a year on a 2D space. This allows us to compare with a real map of the faults and see if our source are localized in the right place (Figure 4.5).

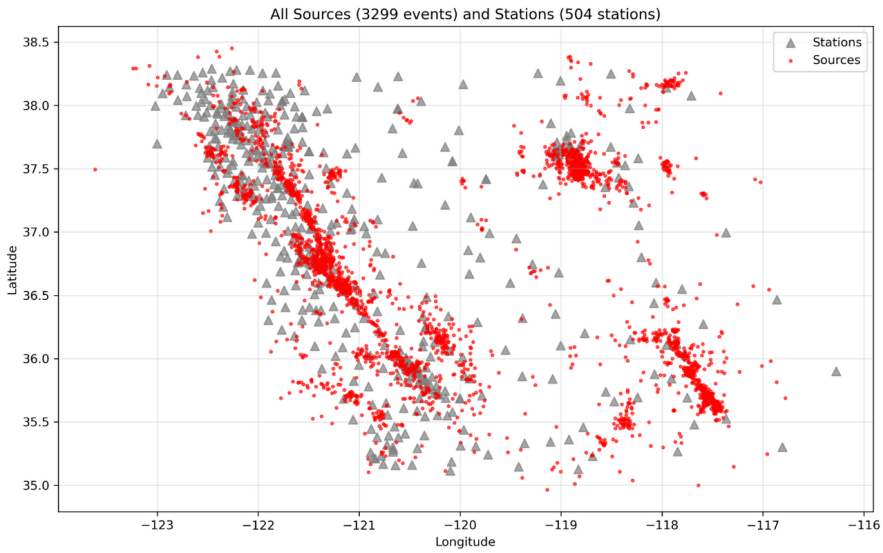

Figure 4.5: Spatial distribution of detected sources over a one-year period in the study region.

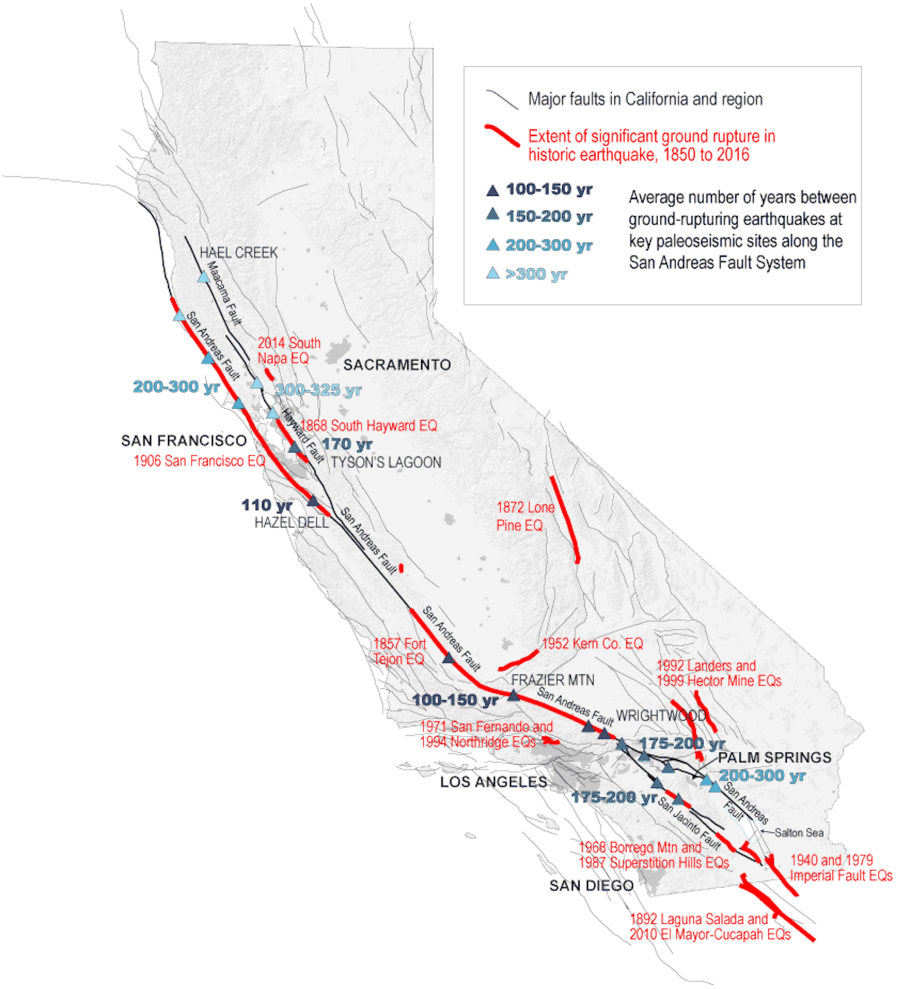

Figure 4.6: Reference map of known faults in the region used for qualitative comparison with detected source locations.

Looking at this figure, we see that the sources closely match with real faults, and is less clustered compared to the old model. That is a good sign that the model is realistic and making good predictions. A lot of the sparse data will then be relocated by the GenieDD process.

4.3 Discussion

4.3.1 Challenges, errors, and adaptations

Challenges The main challenge was bridging two learning curves at once: (i) enough geophysics to judge whether synthetic pick patterns were physically plausible, and (ii) ML know-how to interpret changes in the generator and thresholds. A second challenge was the high interaction between parameters (attenuation, correlations, miss/false rates, thresholds), which made non-linear behaviors. To cope, I relied on systematic visualizations and small ablations to make each effect observable.

Errors The first radial-only generator was difficult to interpret, as several controls were entangled. This led to unstable evaluations, with magnitude skew and thresholds varying unpredictably across stations. It became clear that generator design and thresholding needed to be co-developed.

Adaptations I addressed these issues by redesigning the generator with explicit, low-dimensional controls for attenuation, miss/false rates, and correlations. I re-sampled data to balance magnitude bins, added correlated noise and elliptical geometry, and retuned thresholds to maintain consistency across the pipeline.

4.3.2 Next steps

Future work should:

1. Extend correlated noise to the temporal domain to better reproduce inter-station timing residuals.
2. Add depth and full 3D ellipsoids to model attenuation and coverage realistically.
3. Complete the semi-supervised loop by alternating calibration, retraining, and re-inference, while monitoring recall and completeness on fixed validation windows.

5 Conclusion and Perspectives

5.1 Summary of contributions

This internship tackled the gap between simplistic synthetic generators and real pick statistics, a key limitation for training phase associators. My contributions were:

- **Generator design:** developed a physically motivated generator with elliptical geometry, explicit controls for attenuation and noise, and correlated station behavior.
- **Calibration:** defined a stable, magnitude-aware evaluation objective (inertia, Moran's I, pick counts, overlaps).
- **Optimization:** implemented Bayesian Optimization to tune generator parameters against this objective.
- **Integration:** combined the calibrated generator with GENIE's training pipeline, showing better alignment with real pick structure and improved small-event coverage.

The result is a more realistic and tunable generator, a principled calibration method, and an integration path toward semi-supervised retraining. Remaining limitations are the lack of time-correlated noise, the 2D assumption, and the absence of a fully closed semi-supervised loop.

5.2 Future directions

Promising extensions include: (i) spatio-temporal correlated noise, (ii) depth and 3D ellipsoids for more realism, (iii) a closed-loop semi-supervised pipeline, (iv) robustness tests on other networks, and (v) evaluation of downstream impact on relocation, magnitude estimation, and catalog analysis.

Bibliography

[1] Richard V. Allen. Automatic earthquake recognition and timing from single traces. *Bulletin of the Seismological Society of America*, 68(5):1521–1532, 1978. URL https://link.springer.com/10.1007/978-3-642-35344-4_185.

[2] M. Baer and U. Kradolfer. An automatic phase picker for local and teleseismic events. *Bulletin of the Seismological Society of America*, 77(4):1437–1445, 1987. URL https://pubs.geoscienceworld.org/ssa/bssa/article/77/4/1437/119016.

[3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. URL https://arxiv.org/abs/1609.02907.

[4] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML 2013 Workshop on Challenges in Representation Learning*, 2013. URL http://deeplearning.net/wp-content/uploads/2013/03/pseudo_label_final.pdf.

[5] Ian W. McBrearty and Gregory C. Beroza. Earthquake phase association with graph neural networks. *Bulletin of the Seismological Society of America*, 113(2):524–547, 2023. doi: 10.1785/0120220182. URL https://pubs.geoscienceworld.org/ssa/bssa/article/113/2/524/619845/Earthquake-Phase-Association-with-Graph-Neural.

[6] Ian W. McBrearty and Gregory C. Beroza. Double difference earthquake location with graph neural networks. *Earth, Planets and Space*, 77(1):??, 2025. doi: 10.1186/s40623-025-02251-4. URL https://earth-planets-space.springeropen.com/articles/10.1186/s40623-025-02251-4. Graph Double Difference (GraphDD). Also available as arXiv:2410.19323.

[7] S. Mostafa Mousavi, William L. Ellsworth, Weiqiang Zhu, Lindsey Y. Chuang, and Gregory C. Beroza. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking. *Nature Communications*, 11(3952):1–12, 2020. doi: 10.1038/s41467-020-17591-w. URL https://www.nature.com/articles/s41467-020-17591-w.

[8] Daniel T. Trugman and Peter M. Shearer. Growclust: A hierarchical clustering algorithm for relative earthquake relocation, with application to the spanish springs and sheldon, nevada, earthquake sequences. *Seismological Research Letters*, 88(2A):379–391, 2017. doi: 10.1785/0220160188. URL https://igppweb.ucsd.edu/~shearer/mahi/PDF/2017/Trugman_growclust_2017.pdf.

[9] Felix Waldhauser and William L. Ellsworth. A double-difference earthquake location algorithm: Method and application to the northern hayward fault, california. Open-File Report 01-113, U.S. Geological Survey, 2000. URL https://www.ldeo.columbia.edu/~felixw/papers/Waldhauser_OFR2001.pdf.

[10] Zhuoxiao Xi, Weiqiang Zhu, Gregory C. Beroza, and William L. Ellsworth. Deep learning for deep earthquakes: insights from obs observations of the tonga subduction zone. *Geophysical Journal International*, 238(2):1073–1088, 2024. doi: 10.1093/gji/ggae200. URL https://academic.oup.com/gji/article/238/2/1073/7689220.

[11] Y. Zhong, J. Tytell, J. A. Power, M. West, X. Zhu, and R. Hansen. Deep-learning-based phase picking for volcano and tectonic earthquakes in alaska: Evaluation of phasenet and eqtransformer. *Bulletin of the Seismological Society of America*, 114(5):2457–2477, 2024. doi: 10.1785/0120240096. URL https://pubs.geoscienceworld.org/ssa/bssa/article/114/5/2457/644473/Performance-of-AI-Based-Phase-Picking-and-Event.

[12] Weiqiang Zhu and Gregory C. Beroza. Phasenet: A deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International*, 216(1):261–273, 2019. doi: 10.1093/gji/ggy423. URL https://academic.oup.com/gji/article/216/1/261/5129142.

[13] Weiqiang Zhu, Ian W. McBrearty, S. Mostafa Mousavi, William L. Ellsworth, and Gregory C. Beroza. Earthquake phase association using a bayesian gaussian mixture model. *Seismological Research Letters*, 93(5):2661–2675, 2022. doi: 10.1785/0220210218. URL https://www.osti.gov/pages/biblio/1978539.

A Appendices

A.1 Backpropagation formulation

Traditional methods to do phase association are back-projection (time-reversal stacking), maximum-likelihood and probabilistic associators [13], graph-based optimization, RANSAC-style clustering, and Bayesian mixture-model clustering. One of the simplest is backpropagation, consists in aligning theoretical moveout curves across stations for a candidate source and stacking station-wise evidence to score space–time hypotheses. A common formulation is

$$\mathrm{BP}(\mathbf{x}, t_0) = \sum_{s \in S} w_s \max_{\tau \in D_s} \exp\left(-\frac{\left[\tau - \left(t_0 + T_k(s, \mathbf{x})\right)\right]^2}{2\sigma^2} \right),$$

where S is the station set, D_s the picks on station s, $T_k(s, \mathbf{x})$ the theoretical travel time for phase k from candidate source $\mathbf{x}$ to station s, w_s station weights, and σ a kernel bandwidth.

A.2 Bayesian Optimization details

Bayesian optimization (BO) fits a probabilistic surrogate to an unknown objective $f(\boldsymbol{\theta})$ and uses an acquisition function to choose the next parameters to evaluate. We place a Gaussian process prior on f, yielding at iteration t a posterior mean $\mu_t(\boldsymbol{\theta})$ and variance $\sigma_t^2(\boldsymbol{\theta})$ after observing $\{(\boldsymbol{\theta}_k, f(\boldsymbol{\theta}_k))\}_{k=1}^{t}$.

A common acquisition is Expected Improvement (EI):

$$\mathrm{EI}_t(\boldsymbol{\theta}) \;=\; E\big[\max(0,\, f^\star - f(\boldsymbol{\theta}))\big] \;=\; (f^\star - \mu_t)\,\Phi\left(\frac{f^\star - \mu_t}{\sigma_t}\right) \;+\; \sigma_t\,\phi\left(\frac{f^\star - \mu_t}{\sigma_t}\right),$$

where $f^\star$ is the best observed objective, and ϕ, Φ are the standard normal pdf and cdf. At each step BO selects $\boldsymbol{\theta}_{t+1} = \arg\max_{\boldsymbol{\theta}} \mathrm{EI}_t(\boldsymbol{\theta})$, evaluates $f(\boldsymbol{\theta}_{t+1})$, and updates the surrogate. This is well-suited to low-dimensional, expensive objectives.

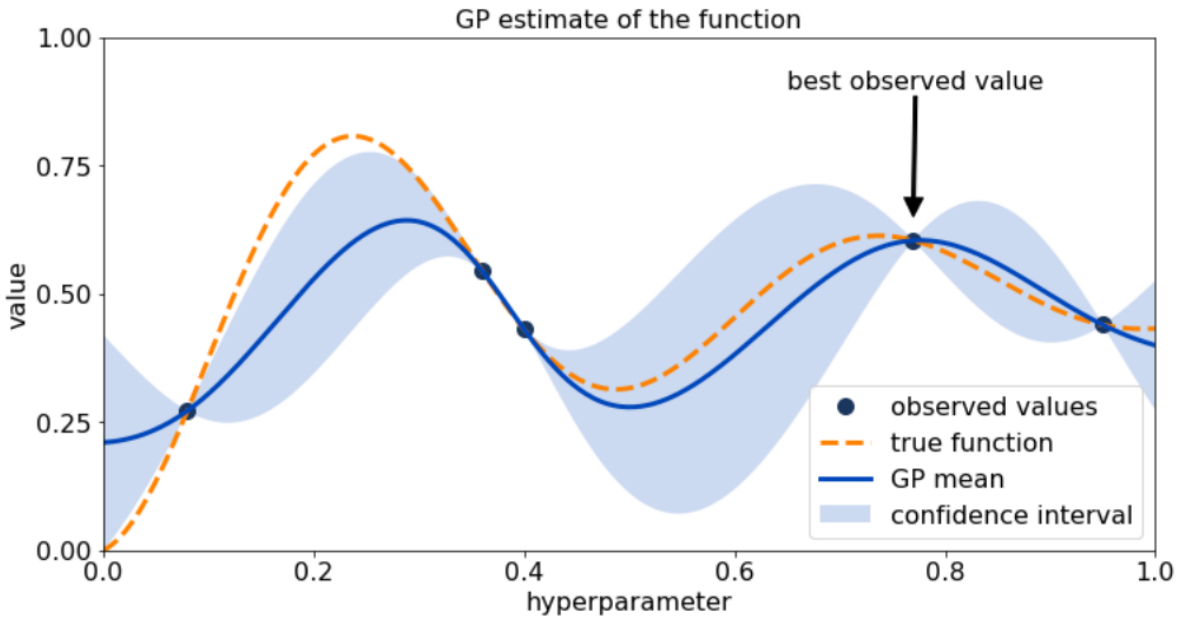

Figure A.1: Illustration of a function optimized by Bayesian optimization with an acquisition function (source: CERN).

We use BO to optimize the parameters of an objective function that quantifies the distance between synthetic data and the best generated catalog so far. The goal is to compare two spatial point distributions and measure how close they are.

Glossary

2

P-wave: Primary (compressional) seismic wave, fastest type of body wave, travels through solids and fluids.

S-wave: Secondary (shear) seismic wave, slower than P-wave, propagates only through solids.

Phase association: Process of grouping seismic picks from multiple stations into coherent earthquake events with estimated source parameters.

PhaseNet: Deep learning model for automatic seismic phase picking from continuous waveforms.

GENIE: Graph Neural Network (GNN) model for phase association, trained on synthetic seismic data.

Graph Neural Network (GNN): Neural network architecture operating on graphs, useful for data with relational structure such as seismic station networks.

Synthetic data: Artificially generated data used to train models when labeled real data are scarce or incomplete.

Bayesian Optimization: Global optimization strategy that builds a probabilistic model of the objective function and selects samples based on uncertainty.

Mahalanobis distance: Generalized distance measure that accounts for correlations between variables, used here for anisotropic calibration.

Radial Basis Function (RBF): Kernel function used to model spatial correlations in synthetic seismic noise generation.

Semi-supervised learning: Machine learning approach that combines limited labeled data with abundant unlabeled data during training.

Seismic catalog: Database of earthquake events including location, origin time, magnitude, and phase picks.