Training SBI for Manning's Value Estimation from Observed Streamflow Data

immediate

1. Objective

Training SBI to learn the relationship between **Manning's values** and **observed streamflow data** from HydroData.

2. HUC Domain:02070001

The analysis is conducted on the **HUC domain 02070001**, which contains **nine different Manning's values**. The spatial distribution of these Manning's values across the domain is illustrated in Figure 1.

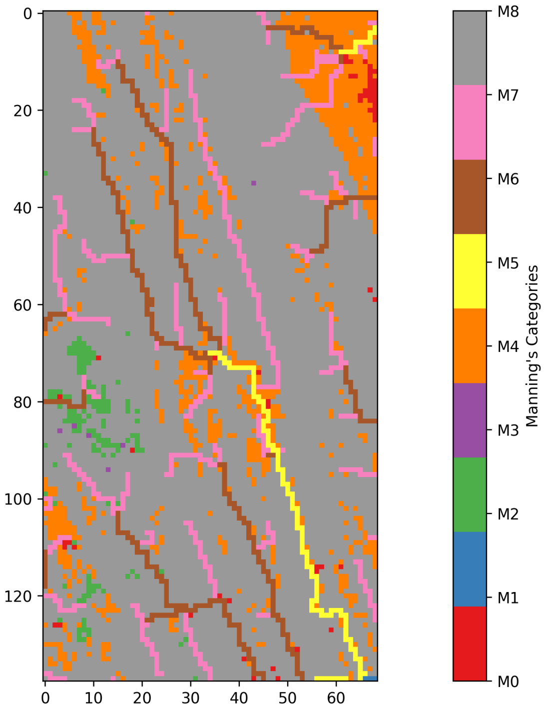


Figure 1. Manning's value distribution.

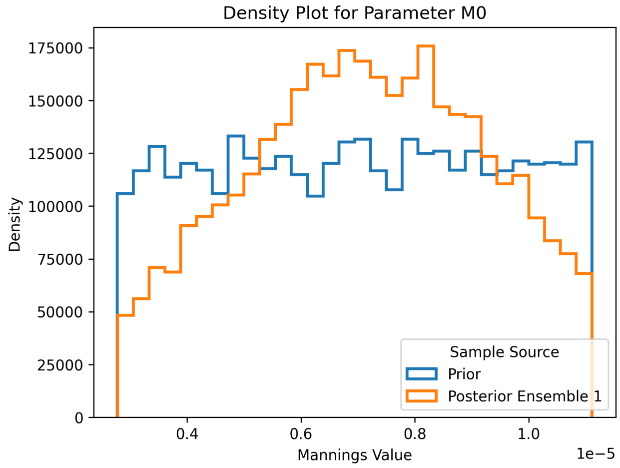
3. Manning's Value and Land Cover

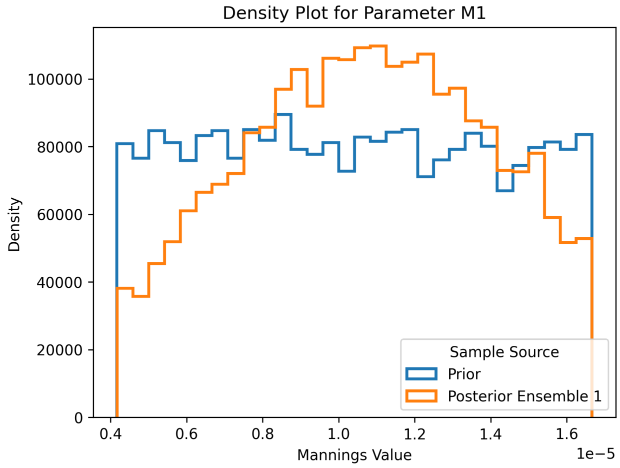
Manning's Value	Land Cover
M0 = 5.56e-06	Urban and built-up lands
M1 = 8.33e-06	Stream order 5
M2 = 9.72e-06	Cropland / natural vegetation mosaics
M3 = 1.39e-05	Shrublands
M4 = 1.53e-05	Savannas
M5 = 1.67e-05	Stream order 3
M6 = 1.94e-05	Stream order 2
M7 = 2.50e-05	Stream order 1
M8 = 2.78e-05	Forests

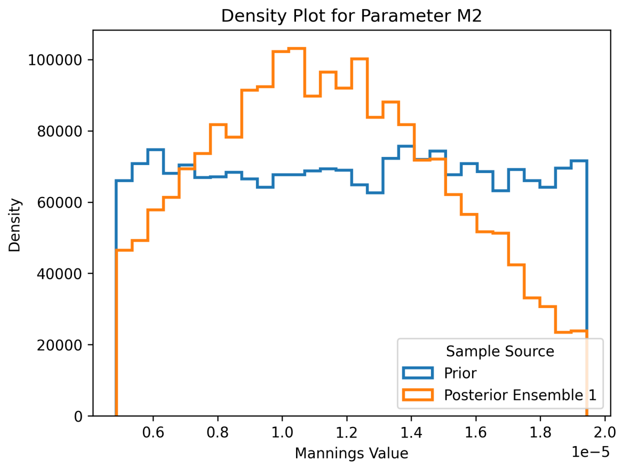
Table 1. Manning's values and corresponding land cover classifications.

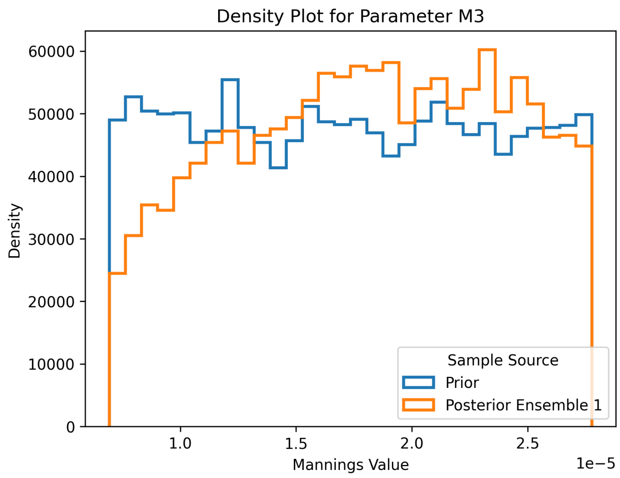
4. First Iteration to Train SBI

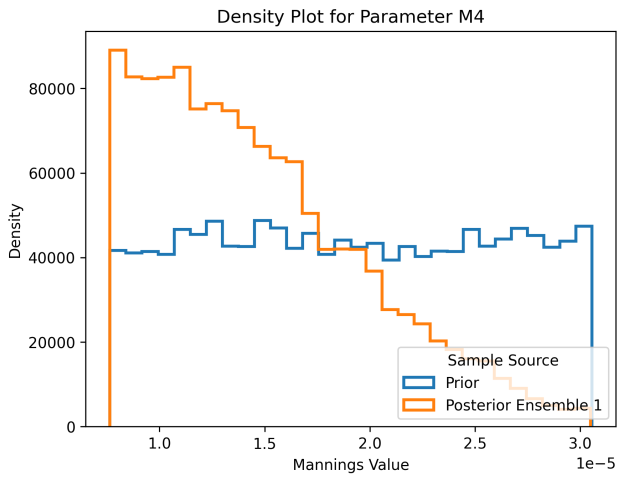
In the first iteration, **800 samples** were drawn from the prior distribution. Each sampled parameter set was used to run ParFlow, generating a total of 800 simulated streamflow outputs.

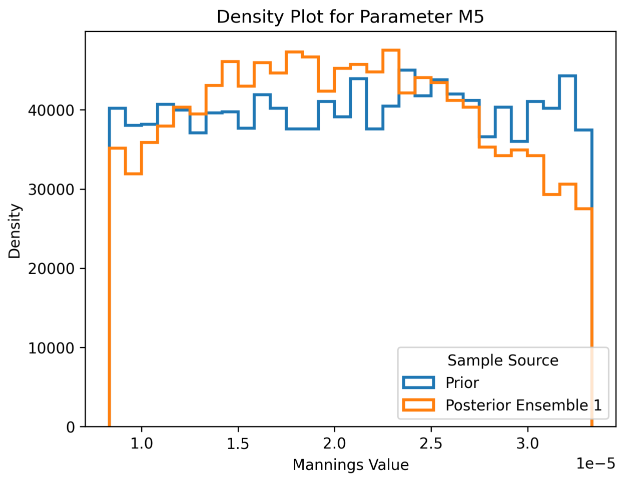

Density Plot for Parameter M0
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1

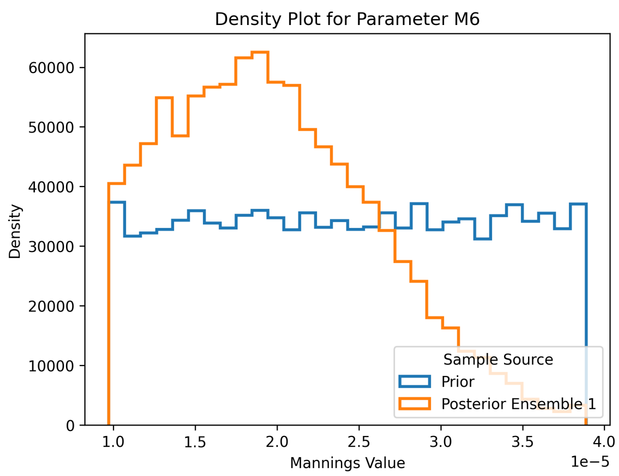

Density Plot for Parameter M1
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1

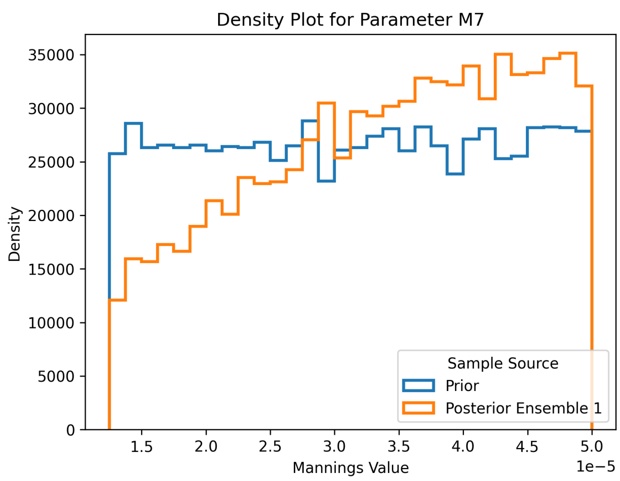

Density Plot for Parameter M2
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1

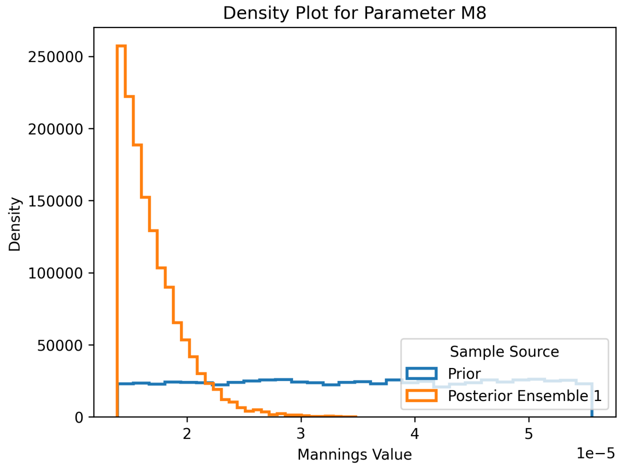

Density Plot for Parameter M3
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1


Density Plot for Parameter M4
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1


Density Plot for Parameter M5
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1


Density Plot for Parameter M6
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1


Density Plot for Parameter M7
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1



5. Discussion of Prior and Posterior Distributions

The prior distribution appears **appropriate** for the following Manning's values

- **M0, M1, M2, M3,M5. and M6**: The highest densities of the posterior distributions fall well within the prior range, suggesting adequate parameter coverage.

For certain Manning's values, the posterior densities indicate that the prior distribution may require a**adjustment**:

- **M4**: The highest posterior densities are concentrated near the *left boundary* of the prior range. This suggests that the prior range should be **extended on the left side** to better capture relevant parameter values.
- **M7**: The posterior distribution shows a clustering of highest density estimates on the *right side*. This suggests that the **right boundary should be extended** to ensure sufficient parameter coverage.
- **M8**: The posterior distribution is **poorly aligned** with the prior, as the highest density is concentrated on the *left side*, while the middle-to-right range has nearly zero density. This misalignment suggests that the prior should be adjusted by **expanding the parameter range on the left side**.

6. Calibration of Prior Distribution

The prior distribution is initially defined as a **uniform** distribution, constructed based on Manning's values derived from the static input data. Its range is determined as follows:

- The **lower bound** is set as the Manning's value divided by a scalar of **2**.
- The **upper bound** is set as the Manning's value multiplied by **2**.

To expand the boundaries and facilitate broader exploration of the parameter space, the **scalar has now been increased to 4**. This adjustment effectively **doubles the prior range**, ensuring a more comprehensive coverage of possible Manning's values and enhancing the flexibility of the model.

7. Manning's Value Ranges For THe Prior Ditribution

Table 2. Original Manning's Range

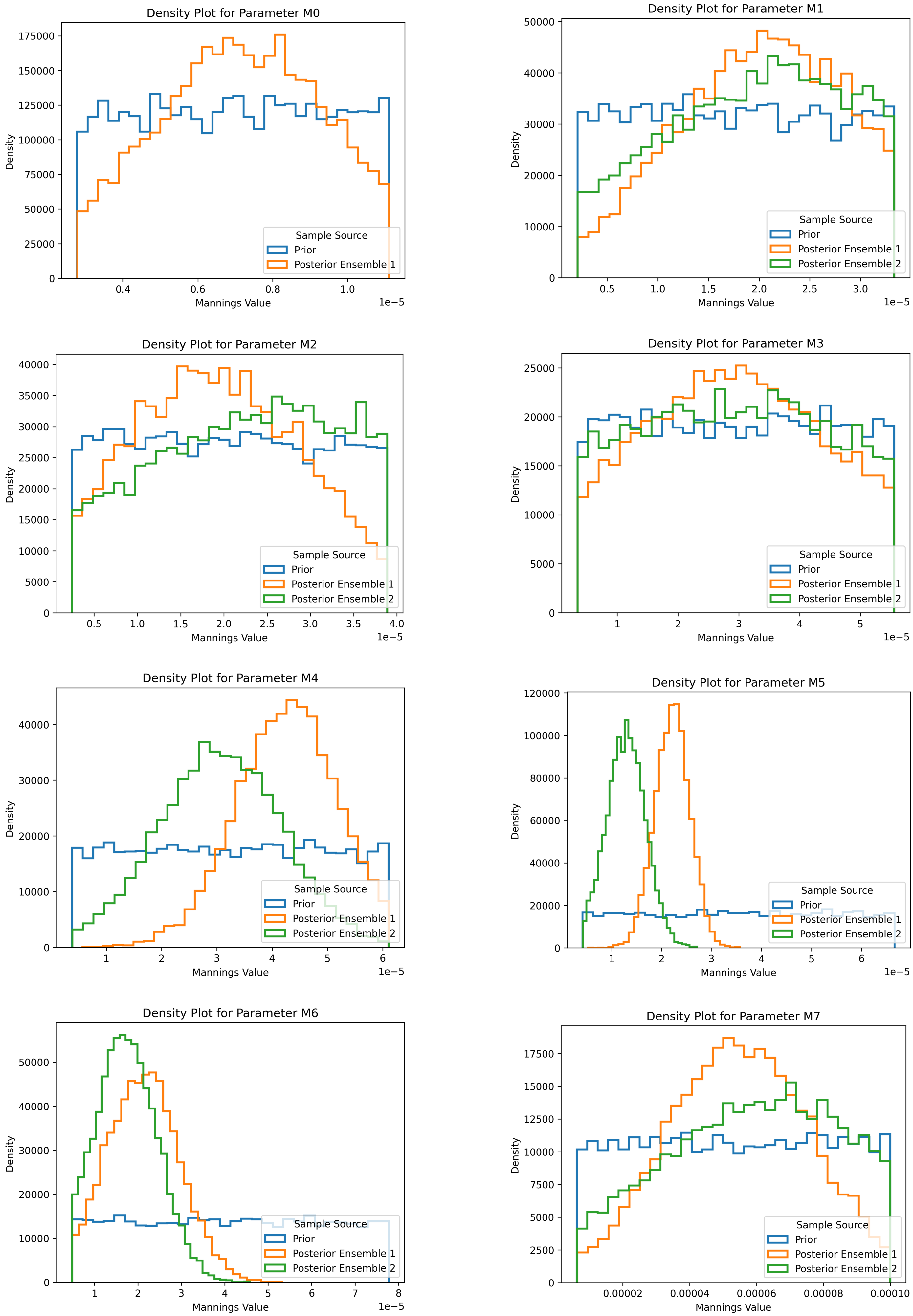
Manning's Index	Lower Bound	Upper Bound
M0	2.78×10^{-6}	1.11×10^{-5}
M1	4.17×10^{-6}	1.67×10^{-5}
M2	4.86×10^{-6}	1.94×10^{-5}
M3	6.95×10^{-6}	2.78×10^{-5}
M4	7.65×10^{-6}	3.06×10^{-5}
M5	8.35×10^{-6}	3.34×10^{-5}
M6	9.70×10^{-6}	3.88×10^{-5}
M7	1.25×10^{-5}	5.00×10^{-5}
M8	1.39×10^{-5}	5.56×10^{-5}

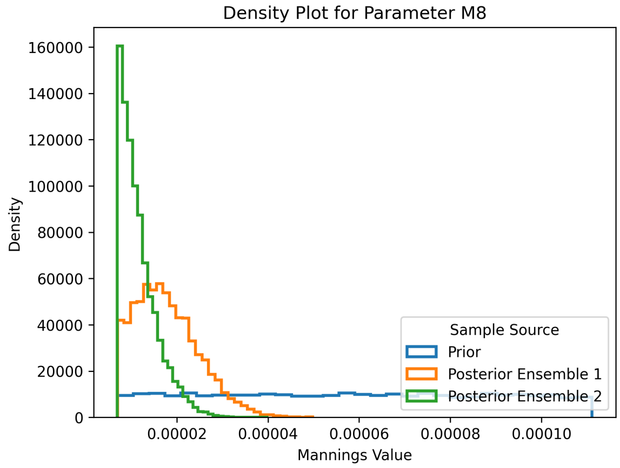
Table 3. Expanded Manning's Values range

Manning's Index	Lower Bound	Upper Bound
M0	1.39×10^{-6}	2.22×10^{-5}
M1	2.08×10^{-6}	3.33×10^{-5}
M2	2.43×10^{-6}	3.89×10^{-5}
M3	3.48×10^{-6}	5.56×10^{-5}
M4	3.82×10^{-6}	6.12×10^{-5}
M5	4.17×10^{-6}	6.68×10^{-5}
M6	4.85×10^{-6}	7.76×10^{-5}
M7	6.25×10^{-6}	1.00×10^{-4}
M8	6.95×10^{-6}	1.11×10^{-4}

8. Training SBI Following Prior Distribution Calibration

Here are the posterior density estimates obtained after calibrating the prior distribution.


Density Plot for Parameter M0
Density Plot for Parameter M1
Density Plot for Parameter M2
Density Plot for Parameter M3
Density Plot for Parameter M4
Density Plot for Parameter M5
Density Plot for Parameter M6
Density Plot for Parameter M7
Density
Mannings Value
1e−5
Sample Source
Prior
Posterior Ensemble 1
Posterior Ensemble 2



9. Discussion of Posterior Distribution

The posterior density estimations from the first ensemble are better than those from the second, as they more effectively capture the pattern. In contrast, the second ensemble shows a reduced ability to learn the pattern.

10. Selecting minimum and maximum noise

These values control how much perturbation is introduced into the data to ensure smooth training. In real-world cases, simulators are imperfect, meaning that:

$$x_{\text{obs}} \notin p(x \mid \theta)$$

This means the posterior estimator needs to account for differences between observed and simulated data.
The noise parameter can be determined by computing **the discrepancy** between **the observed data** and **the simulated outputs** from ParFlow.
The noise is assumed to follow a **Gaussian distribution**.

11. Training and Validation Phase

Manning's value is independent of time and depends solely on spatial characteristics. Thus, once the model is trained, the Manning's value should remain constant regardless of the streamflow at any given time. Streamflow is dependent on various environmental factors, including precipitation, temperature, evapotranspiration, and cloud cover. The importance of dealing with the same dates lies in the ability to compare the output streamflow, given the Manning's value obtained from the posterior distribution, with the observed data from `hydr_data`.

Questions: Can Simulation-Based Inference (SBI) learn the relationship such that, when provided with x_{obs} from different months or years, it consistently predicts the same Manning's value (M), given that M is **independent of time** but **depends on spatial characteristics**?

If SBI successfully captures this relationship, we can replace x_{obs} with data from different years, the same month, or even different months, and the estimated Manning's value should remain consistent.

12. Calibrating The Hyperparameters

Hyperparameter calibration is essential in SBI, particularly in **Sequential Neural Posterior Estimation (SNPE)**, to ensure an accurate and well-calibrated posterior distribution.

The hyperparameters in SBI that could be calibrated to improve te learning and the convergence include:

- Learning rate (η)
- Number of hidden layers and neurons
- Batch size
- Type of density estimator (e.g., MAF, NSF)
- `num_simulations`: Number of simulations per round
- Proposal distribution: Controls how new parameter samples are chosen
- `noise_min` and `noise_max`: Handle uncertainty from simulation mismatch

QUESTIONS: How could be calibrated these hyperparameters?

13. Filtering The Posterior Distribution

The first step involves filtering the posterior samples based on their density:

- Only the highest-density samples (e.g., top 4%) are retained.
- For each retained θ (Manning's value), ParFlow generates simulated streamflow data x_{sim}.

- RMSE to standard deviation is calculated to measure how well the simulated streamflow (x_{sim}) approximates the observed streamflow (x_{obs}).
- To further refine our parameter selection, we introduce a **thresholding**

This ensures that only high-probability Manning's values are used in inference, making it more reliable. Visualizing the Difference: Observed vs. Simulated Streamflow To further validate the approach, we plot the observed streamflow x_{obs} and the simulated streamflow x_{sim} since it allows us to visually inspect the differences.

QUESTIONS: Is this approach computationally efficient?

14. SNPE and NLE

In **Simulation-Based Inference (SBI)**, two common methods for estimating parameter distributions are **Neural Posterior Estimation (NPE)** and **Neural Likelihood Estimation (NLE)**. Both approaches aim to approximate the posterior distribution $p(\theta|x)$, where θ represents Manning's value and x represents the observed streamflow. **Neural Posterior Estimation (NPE)** directly learns the posterior distribution:

$$q_\phi(\theta|x) \approx p(\theta|x)$$

where q_ϕ is a neural network-based approximation.

Sample Efficiency:

- If the number of simulations is limited, **NLE is more efficient**, as it uses each simulation more effectively.
- However, if enough simulations are available, **NPE is preferable due to its direct posterior estimation**.

15. Optimization

15.1. Sampling Efficiency

Sampling efficiency aims to reduce unnecessary simulations while ensuring that high-confidence Manning's values are used for inference. The optimization strategy involves:

- **Step 1: Define a Posterior Density Threshold**
  Instead of using all posterior samples, we retain only the highest-density samples from the posterior:

  $$q_\phi(\theta|x) > d_{\text{thresh}}$$

  where d_{thresh} is dynamically adjusted based on past inference quality.
- **Step 2: Run ParFlow Simulations Using Filtered Samples**
  Using only the high-density Manning's values, ParFlow is executed to simulate streamflow.
- **Step 3: Compare Simulated and Observed Values**
  The simulated x_{sim} and observed x_{obs} values are compared using an error metric, such as the Root Mean Square Error (RMSE):

  $$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{\text{sim},i} - x_{\text{obs},i})^2}$$

- **Step 4: Adaptive Thresholding for Optimization**
  If RMSE is too high, the posterior density threshold is adjusted to further restrict sampling to more confident Manning's values. This avoids running simulations on low-confidence values, reducing computational cost.

Optimization Outcome: This method optimizes SBI by reducing the number of unnecessary simulations, ensuring that only high-quality Manning's values are used for inference.

15.2. Cost Function Optimization

A well-defined cost function can help balance computational efficiency and accuracy in SBI. The optimization strategy involves:

1. Defining a Cost Function
   The total computational cost C is expressed as:

   $$C = \alpha \cdot T_{\text{sim}} + \beta \cdot \text{Bias}(\theta)$$

   where:

   - T_{sim} is the total simulation time.
   - $\text{Bias}(\theta)$ measures how far the estimated θ (Manning's value) is from the true value.
   - α, β are weighting factors that balance computational cost and accuracy.
2. Bias-Variance Tradeoff Optimization
   Since SBI involves uncertainty, we adjust the cost function to minimize bias while controlling variance:

   $$\text{Bias-Variance Cost} = C + \gamma \cdot \text{Var}(\theta)$$

   where:

- $\text{Var}(\theta)$ is the variance of posterior estimates.
- γ controls the penalty for high variance.

3. Optimization Through Gradient-Based Updates
If C is differentiable, gradient-based optimization (e.g., SGD or Adam) can be applied:

$$\theta_{t+1} = \theta_t - \eta \frac{\partial C}{\partial \theta}$$

where η is the learning rate.

Optimization Outcome: By incorporating computational cost into the optimization process, we ensure that SBI remains efficient while maintaining high inference accuracy.

15.3. Informative Prior Distribution

Using an informative prior can significantly reduce computational cost and improve convergence. The optimization strategy involves:

1. Restricting the Sampling Space
Instead of sampling from a broad prior, we use an informative prior based on prior knowledge:

$$p(\theta) \propto \exp\left(-\frac{(\theta-\mu)^2}{2\sigma^2}\right)$$

where:

- μ is the expected Manning's value based on prior studies.
- σ^2 controls the prior's spread.

2. Reducing Total Simulations
By restricting sampling to high-likelihood regions, we reduce the number of necessary simulations, leading to faster convergence.
3. Adaptive Prior Updating
As new observations are incorporated, the prior is updated using Bayesian updating:

$$p(\theta|x) \propto p(x|\theta)p(\theta)$$

This ensures that the prior remains **informative and adaptive**.

16. Joint and Marginal Probability

The joint probability of two random variables X and Y represents the probability that both events occur simultaneously. It is denoted as:

$$P(X=x, Y=y) \quad \text{or} \quad P(X,Y)$$

For discrete random variables, the joint probability mass function (PMF) is:

$$P(X=x, Y=y) = P(X,Y)$$

For continuous random variables, the joint probability density function (PDF) is:

$$f(X,Y) = f(x,y)$$

The marginal probability of a single variable X is obtained by summing (discrete case) or integrating (continuous case) the joint probability over the other variable Y. It describes the probability of X occurring, regardless of Y.

16.1. Discrete Case

For a discrete random variable:

$$P(X=x) = \sum_y P(X=x, Y=y)$$

16.2. Continuous Case

For a continuous random variable:

$$f(X=x) = \int_{-\infty}^{\infty} f(X=x, Y=y)\, dy$$

16.3. Relationship Between Joint and Marginal Probability

The marginal probability is derived from the joint probability. In a two-variable system:

$$P(X) = \sum_Y P(X,Y)$$

or for continuous distributions:

$$f(X) = \int_{-\infty}^{\infty} f(X,Y)\, dY$$

17. Sampling with MCMC

If SNPE struggles with convergence due to strict filtering e.g., only retaining 0.56% of samples, MCMC can help explore the full posterior more effectively. **QUESTIONS:** Why in this case MCMC works better ?

18. Equifinality problem

The equifinality problem refers to the situation where multiple different parameter sets produce similar or identical model outputs, making it difficult to uniquely determine the "true" parameter values. It is commonly encountered in hydrological modeling, where different parameter combinations lead to nearly indistinguishable results.
QUESTIONS: How can I avoid Equifinality?

19. Convergence Issue