CS772 Project Report

Team Members
Kshitij Bagga
Aadi Singh
Meet Lalwani
Sethu Prasad Senthil

Abstract

We study the impact of different stochasticity levels in Bayesian LSTM models for sequence modeling and uncertainty estimation. Specifically, we compare four variants: a fully deterministic LSTM, a partially stochastic LSTM with a stochastic final layer trained via mean-field variational inference, a fully stochastic LSTM using the same inference method, and a final-layer stochastic LSTM trained with Stochastic Gradient Langevin Dynamics (SGLD). Despite the common assumption that full stochasticity is more expressive or principled, our experiments reveal no consistent performance advantage. In fact, partially stochastic models—especially those with stochasticity limited to the output layer—achieve comparable or superior results in both predictive accuracy and uncertainty quantification, while offering significant gains in efficiency. These findings challenge the default preference for fully stochastic recurrent architectures and highlight the practical benefits of targeted stochasticity.

1 Introduction

Bayesian neural networks (BNNs) provide a principled framework for modeling uncertainty in deep learning by treating model parameters as distributions rather than point estimates. This approach has shown particular promise in sequence modeling tasks, where capturing uncertainty is often as important as making accurate predictions. Bayesian Long Short-Term Memory (LSTM) networks, in particular, aim to extend this paradigm to recurrent architectures, enabling uncertainty-aware modeling of temporal data.

Traditionally, BNNs assume full stochasticity—that is, distributions are placed over all weights and biases in the network, and inference is performed jointly across them. However, this standard construction poses serious practical challenges. Exact inference is intractable, and approximate methods such as mean-field variational inference or SGLD are often computationally expensive, sensitive to initialization, and may fail to represent the true posterior accurately.

Recent work questions whether full stochasticity is necessary for good predictive performance or uncertainty estimation. In this project, we extend this investigation to Bayesian LSTM models, comparing four variants: a fully deterministic LSTM, a partially stochastic LSTM with a stochastic output layer trained via mean-field variational inference, a fully stochastic LSTM using the same inference method, and a final-layer stochastic LSTM trained with Stochastic Gradient Langevin Dynamics (SGLD).

Our aim is to evaluate whether introducing stochasticity selectively—particularly in the final layer—can yield the benefits of Bayesian modeling without the full computational burden of full stochastic inference. Through empirical analysis, we find that partially stochastic models often perform on par with, or better than, fully stochastic counterparts, both in terms of prediction quality and uncertainty calibration. These findings challenge the conventional emphasis on full stochasticity in Bayesian RNNs and suggest that carefully chosen partial stochasticity may offer a better trade-off between performance and efficiency in practice.

1.1 Problem Description

In this project, we aim to systematically compare four variants of Bayesian LSTMs—(1) fully deterministic, (2) partially stochastic with a variational output layer, (3) fully stochastic using mean-field variational inference, and (4) partially stochastic with an SGLD-trained output layer—on the task of predicting the evolution of Euler angles in synthetic microstructure data under incremental strain. The objective is to evaluate whether full parameter stochasticity is necessary or if selective stochasticity can offer comparable predictive performance and uncertainty estimates with reduced computational overhead. This comparison will inform the design of efficient, uncertainty-aware models for time-dependent scientific prediction tasks.

1.2 Literature Review and Prior Work

Sharma et al. (2023) in Do Bayesian Neural Networks Need To Be Fully Stochastic? challenge this assumption by both theoretically and empirically demonstrating that full stochasticity is not a prerequisite for expressive predictive distributions. Their findings show that partially stochastic models—those with stochasticity limited to select layers—can act as universal conditional distribution approximators and often match or outperform fully stochastic models across various inference schemes and datasets. This work provides a strong foundation for rethinking the design of Bayesian deep learning models with an emphasis on practicality and efficiency.

In parallel, prior undergraduate research (UGP) applied a deterministic Recurrent Neural Network (RNN) architecture—specifically, an LSTM model—to simulate the evolution of Euler angles in a synthetic microstructure subjected to incremental strain. The model demonstrated strong extrapolation performance by learning temporal dependencies from simulation data and predicting deformation behavior beyond the trained strain range. While effective in performance, this approach lacked mechanisms for uncertainty estimation, which are critical for scientific and safety-critical predictions.

1.3 Novelty

The novelty of the current work lies in unifying these two strands: we introduce controlled stochasticity into the LSTM architecture to bridge high extrapolation performance with uncertainty. This approach enables a systematic exploration of the trade-offs between accuracy, uncertainty quantification, and computational cost in scientific simulations, addressing a critical gap between theoretical expressiveness and practical deployment.

2 Team Members

This section details the contributions made by each member of the team.

- Aadi Singh : [worked on Last Layer Stochastic VI and Report]
- Kshitij Bagga : [worked on the Deterministic Model, Error Graphs and Report]
- Meet Lawani : [worked on SGLD model and Report]
- Sethu Prasad : [worked on Fully Stochastic VI and Report]

3 Description of Tools/Software Used

- **Python 3.9:**

  Python served as the core programming language, enabling efficient data processing, model development, and experimentation.
- **PyTorch:**

  Used for deep learning model development, specifically for implementing ***neural networks***, including ***LSTMs***.

- **Pyro:**
  A probabilistic programming library built on PyTorch, Pyro was utilized to model and implement ***variational inference*** techniques, such as ***Stochastic Gradient Langevin Dynamics***, to incorporate uncertainty in the model predictions.
- **NumPy:**
  NumPy provided essential numerical operations, enabling fast and efficient handling of large datasets and array manipulations for data preprocessing.
- **Pandas:**
  Facilitated the *manipulation* and *cleaning* of datasets, providing efficient tools for data organization and transformation into a usable format for model training.
- **Scikit-learn:**
  Used for model evaluation, particularly for calculating performance metrics such as **mean squared error (MSE)**.
- **Glob:**
  Used for file ***management***, enabling the loading and processing of ***multiple*** dataset files based on specific patterns.
- **CUDA:**
  Utilized for ***GPU*** acceleration, allowing faster computations during model training and enabling efficient use of hardware resources.
- **Kaggle GPUs (T4x2):**
  The project leveraged ***Kaggle's T4x2 GPUs*** for high-performance model training, significantly speeding up training time
- **ATEX**:
  Used for generating ***microstructures*** from ***Euler angles*** based on both model predictions and experimental data, enabling the visualization and analysis of ***material microstructures*** in the project.

4 Dataset Description

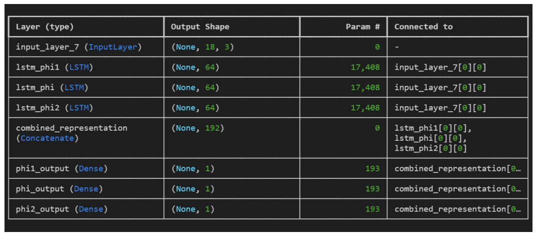
The dataset used in this project consists of **20 CSV files**, with **19 files used for training** and **1 file reserved for testing** the predictive model. Each file represents material data obtained under specific strain conditions, with each strain condition corresponding to a **unique strain percentage** (e.g., *1%, 2%*, ... *20%*.). These files contain data captured during various phases of material testing, where the properties of the material evolve under increasing strain.

The dataset includes the following key features:

- **Phi1, Phi, Phi2**: These columns represent angular or phase-related measurements, which are likely related to the crystallographic or microstructural orientation of the material. The values in these columns help to describe the material's internal structure as it undergoes strain.
- **KAM (*Kernel Average Misorientation*)**: This feature quantifies the misorientation between adjacent grains in the material, often used to assess the degree of deformation and dislocation structure in materials science.
- **X, Y**: These columns represent coordinates associated with sample positions or data points during testing.

Each file corresponds to a **strain condition**, with the strain levels denoted by variations in the Phi1, Phi and Phi2 columns, indicating how the material's properties change under applied stress. The files are structured as sequential data, which allows the application of time-series analysis or machine learning models to predict material behavior under various strain conditions.

The **training files (1-19)** are used to train predictive models that aim to understand how the material behaves under different strain levels. The **testing file (file 20)** is used to validate the model's performance and evaluate its ability to generalize to unseen strain conditions.



Layer (type)	Output Shape	Param #	Connected to
input_layer_7 (InputLayer)	(None, 18, 3)	0	-
lstm_phi1 (LSTM)	(None, 64)	17,408	input_layer_7[0][0]
lstm_phi (LSTM)	(None, 64)	17,408	input_layer_7[0][0]
lstm_phi2 (LSTM)	(None, 64)	17,408	input_layer_7[0][0]
combined_representation (Concatenate)	(None, 192)	0	lstm_phi1[0][0], lstm_phi[0][0], lstm_phi2[0][0]
phi1_output (Dense)	(None, 1)	193	combined_representation[0…
phi_output (Dense)	(None, 1)	193	combined_representation[0…
phi2_output (Dense)	(None, 1)	193	combined_representation[0…

Figure 1: LSTM Backbone

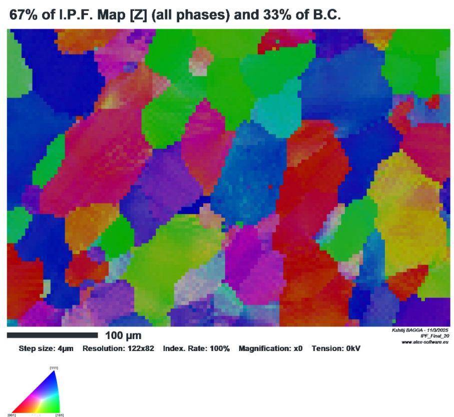


Figure 2: Orginal 20% Strain Microstructure

This dataset is particularly useful for modeling material behavior as a function of strain, which is crucial for understanding the mechanical properties of materials under stress, such as yield strength, ductility, and hardness.

5 Experimental Results

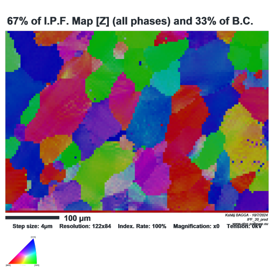


Figure 3: Deterministic LSTM - IPF

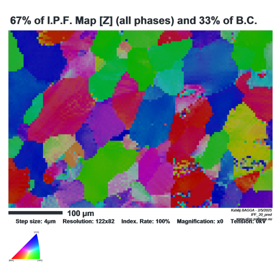


Figure 4: Partially Stochastic VI - IPF

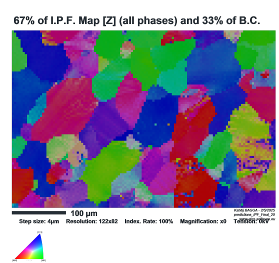


Figure 5: Fully Stochastic VI - IPF

Figure 6: Comparison of IPFs

Figure 6 Shows IPF Maps for deterministic, partially stochastic, and fully stochastic LSTM models. While the deterministic model produces the most accurate and visually clean output, both stochastic models introduce uncertainty-aware variations.

Model	Deterministic	last layer SVI	Fully SVI	last layer SGLD
RMSE	5.4855	13.3633	19.3111	14.7996

Table 1: cross examination of RMSE (root mean squared error) across different models

Table 1 confirms this trade-off: the deterministic model achieves the lowest RMSE, while partial stochasticity (especially with SGLD) balances uncertainty modeling with reasonable accuracy.

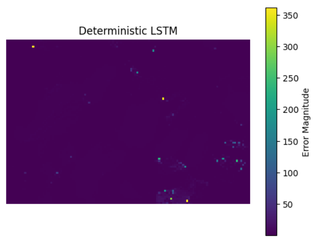


Figure 7: Deterministic LSTM - Error Map

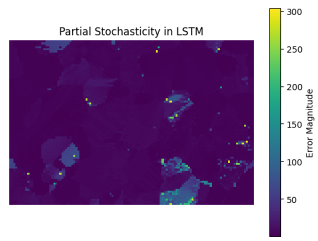


Figure 8: Partially Stochastic VI - Error Map

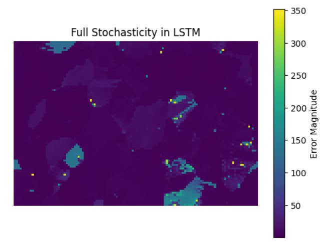


Figure 9: Fully Stochastic VI - Error Map

Figure 10: Comparison of Error Maps

The error maps in fig:9 compare three LSTM model variants—deterministic, partially stochastic (variational inference), and fully stochastic (variational inference)—in predicting Euler angles under strain. Both the partially stochastic and fully stochastic models successfully capture uncertainty, as evidenced by the more structured and dispersed error patterns in their maps. In contrast, the deterministic model shows higher concentration of error in specific regions, indicating limited capacity to account for uncertainty. Notably, the partially stochastic model achieves comparable visual performance to the fully stochastic one, highlighting that full stochasticity may not be necessary for effective uncertainty modeling.

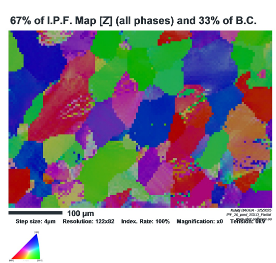


Figure 11: IPF MAP SGLD

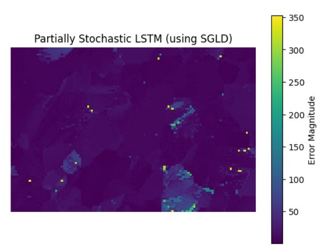


Figure 12: Error Map for SGLD

6 Lessons Learned

Throughout the course of this project, several important lessons were learned that significantly shaped our approach to Bayesian LSTM models and their application in sequence modeling tasks:

1. Selective Stochasticity vs. Full Stochasticity:

One of the key takeaways was the realization that full stochasticity (applying distributions over all model parameters) is not always necessary for effective predictive modeling and uncertainty estimation. By selectively introducing stochasticity, particularly in the output layer, we can maintain performance while achieving significant computational efficiency. This challenges the common assumption in Bayesian deep learning that fully stochastic models always outperform their partially stochastic counterparts.

2. Importance of Uncertainty Estimation in Predictive Models:

While LSTM models are excellent at learning temporal dependencies and making predictions, uncertainty estimation is crucial, especially in scientific applications like microstructure prediction under strain. The project highlighted the importance of uncertainty quantification in model predictions, especially in safety-critical and materials science tasks. We learned that incorporating Bayesian methods, even selectively, can significantly improve the model's ability to convey the uncertainty of its predictions.

3. Practical Applications of Bayesian Deep Learning:

The project also deepened our understanding of how Bayesian deep learning techniques can be applied effectively to scientific prediction tasks. We learned that targeted stochasticity in the output layer allows for efficient uncertainty-aware predictions, making Bayesian models more practical for real-world applications, particularly when dealing with material behavior predictions under external conditions like strain.

4. Challenges with Hamiltonian Monte Carlo (HMC):

While we initially attempted to use Hamiltonian Monte Carlo (HMC) for inference, we encountered issues with vanishing step sizes and low acceptance probabilities, which made it impractical for our model. Due to these challenges, we switched to alternative methods such as mean-field variational inference (VI) and Stochastic Gradient Langevin Dynamics (SGLD). These methods provided more stable and feasible alternatives for approximating the posterior distribution, particularly in the context of our large-scale sequence modeling task.

7 Future Work

We can explore several avenues to further enhance the effectiveness of Bayesian LSTM models. One direction would be to increase the number of stochastic parameters by extending stochasticity beyond the output layer, incorporating hidden layers into the model. This would allow the network to capture uncertainty in intermediate states, potentially improving the model's ability to generalize and handle complex data patterns.

Additionally, it would be valuable to apply stochasticity to the inner hidden layers, rather than limiting it to just the final output layer. By introducing stochastic behavior into the recurrent layers, the model could better represent uncertainty at every level of the sequence, offering a more comprehensive uncertainty estimation.

Finally, to build on our findings, we plan to extend the work to other models discussed in the seed paper, and perform a thorough comparison of the Evidence Lower Bound (ELBO) scores. This will help identify the most suitable model for this problem, balancing predictive accuracy, uncertainty quantification, and computational efficiency. This comparison will provide a clearer path toward optimizing the model architecture for complex time-dependent prediction tasks

8 Conclusion

This project explored the use of Bayesian LSTM models for sequence modeling and uncertainty estimation, focusing on the prediction of Euler angles in synthetic microstructure data under incremental strain. We compared four variants of Bayesian LSTMs with varying levels of stochasticity:

Fully Deterministic LSTM

Partially Stochastic LSTM with a Stochastic Output Layer (Variational Inference)

Fully Stochastic LSTM (Stochastic Variational Inference)

Partially Stochastic LSTM with a Stochastic Output Layer (SGLD)

The goal was to assess whether full parameter stochasticity is necessary for high predictive accuracy and uncertainty estimation, or if selective stochasticity in the output layer can achieve comparable results with lower computational cost.

Our findings challenge the assumption that full stochasticity is always required, showing that partially stochastic models, especially those with stochasticity limited to the output layer, can provide similar or better performance while being more computationally efficient. The accuracy of the models, from highest to lowest, is: Deterministic model, Partially stochastic last layer using VI, Partially stochastic last layer using SGLD, and Fully stochastic VI, with the deterministic model yielding the best results. This approach offers a balanced trade-off between accuracy and efficiency, making it suitable for real-world scientific tasks that require both predictive accuracy and uncertainty estimation.

References

Seed Paper: https://arxiv.org/abs/2211.06291

Additional results and experiment details

Github Repository: https://github.com/kshitijbagga/CS772