Bensemble: A Python Library for Bayesian Neural Networks

Sobolevsky Fedor, Nabiev Mukhammadsharif, Vasilenko Dmitry, Kasyuk Vadim

1 Introduction

`bensemble` is a Python library for building Bayesian neural networks (BNNs) with a unified interface for approximate posterior inference and uncertainty estimation. While deep neural networks have achieved remarkable success across various domains, their predictions are often overconfident, limiting reliability in safety-critical applications. `bensemble` addresses this gap by implementing multiple state-of-the-art Bayesian inference methods in a consistent PyTorch-compatible framework.

The library supports four approximation techniques:

- **Variational Inference (VI)** for scalable optimization-based posterior approximation
- **Laplace Approximation (LA)** for Hessian-based uncertainty estimation
- **Probabilistic Backpropagation (PBP)** for moment-propagation-based inference
- **Variational Rényi (VR)** for flexible divergence minimization

Designed for researchers and practitioners, `bensemble` enables seamless integration of Bayesian principles into PyTorch workflows while maintaining computational efficiency and usability. This document provides a technical overview of the library's architecture, implementation, and usage patterns.

2 Supported Methods

2.1 Variational Inference (VI)

VI approximates the posterior over weights with a tractable distribution. Key features:

- Mean-field Gaussian variational posterior.
- Reparameterization trick for stochastic gradient optimization.
- Optimizes the Evidence Lower Bound (ELBO).
- Monte Carlo sampling for predictive mean and uncertainty estimation.

2.2 Laplace Approximation (LA)

LA approximates the posterior as a Gaussian centered at the MAP estimate. Features:

- Hessian-based covariance computation using Kronecker-factored approximation (KFAC-LA).
- Hooks capture activations and pre-activation Hessians.

- Posterior sampling for uncertainty estimation.
- Compatible with any linear architecture.

2.3 Probabilistic Backpropagation (PBP)

PBP propagates first and second moments through the network. Features:

- Assumed Density Filtering updates weight and noise posteriors sequentially.
- Supports ReLU and linear layers with analytical moment propagation.
- Efficient for small- to medium-sized datasets.
- Produces predictive mean and variance for regression, and predictive probabilities for classification.

2.4 Variational Rényi (VR)

VR generalizes VI using the α-Rényi divergence. Features:

- Reparameterized Gaussian weights with μ and ρ.
- Minimizes Rényi divergence for flexible posterior approximation.
- Monte Carlo sampling for predictions and uncertainty estimation.
- Tunable α parameter for controlling divergence behavior.

3 Interface Overview

3.1 Model Initialization

- Accepts arbitrary `nn.Module` PyTorch models.
- Automatically converts linear layers to Bayesian layers for VI and VR.
- Stores a template model for posterior sampling.

3.2 Training and Fitting

- `fit` trains models using the selected Bayesian method.
- Supports standard optimizers (Adam) and gradient clipping.
- Allows specification of epochs, learning rate, and number of Monte Carlo samples.

3.3 Prediction and Sampling

- `predict` returns predictive mean and optionally posterior samples.
- `sample_models` generates fully sampled deterministic models from the posterior.
- Supports estimation of aleatoric and epistemic uncertainty.

3.4 State Management

- _get_ensemble_state and _set_ensemble_state for saving/loading model state.
- Preserves optimizer state, model parameters, and hyperparameters.

4 Experiments and Evaluation

4.1 Experimental Setup

We evaluate bensemble on the Boston Housing dataset (506 samples, 13 features) with an 80/20 train-test split.

4.2 Performance Metrics

- **RMSE**: Root Mean Squared Error, measures prediction accuracy.
- **NLL**: Negative Log-Likelihood, measures probabilistic calibration.
- **ECE**: Expected Calibration Error, quantifies the agreement between predicted probabilities and actual outcomes.
- **Brier Score**: Measures accuracy of probabilistic predictions.

4.3 Results on Boston Housing

Table 1: Performance comparison on Boston Housing test set (n=101)

Method	RMSE	NLL	ECE	Brier Score
Deterministic MLP	4.211	47.994	0.0329	0.0895
MC Dropout	3.835	3.676	0.0190	0.0546
HMC Linear	5.850	3.222	0.0063	0.0940
Variational Rényi	5.364	3.155	0.0057	0.1274
Probabilistic Backprop	5.420	3.170	0.0081	0.0913
Variational Inference (VI)	4.631	**2.882**	0.0080	0.0991
Laplace Approx	19.377	4.549	0.0108	0.2412

Key observations:

- MC Dropout achieves the best RMSE, while Deterministic MLP follows closely.
- VI achieves the best NLL, indicating superior probabilistic calibration.
- Bayesian methods generally have lower ECE than deterministic approaches.
- Laplace Approximation performs poorly, likely due to Hessian approximation issues.

4.4 Adversarial Robustness Analysis

We evaluate models under FGSM attacks with varying ϵ:

Table 2: RMSE under FGSM adversarial attacks

ϵ	Det. MLP	HMC Lin.	Laplace	MC Dropout	VI	PBP	VR
0.00	4.21	5.86	19.22	3.80	4.58	5.42	5.35
0.05	5.79	6.00	20.63	4.40	4.79	5.60	5.49
0.10	7.41	6.19	24.04	5.16	5.16	5.73	5.76
0.15	8.96	6.42	21.71	5.99	5.38	6.01	5.88
0.20	10.45	6.69	21.34	6.90	5.65	6.33	6.19
0.30	13.13	7.32	22.92	8.82	6.40	6.87	6.64
0.40	15.50	8.04	22.93	10.75	7.02	7.60	7.54
0.50	17.70	8.84	26.40	12.67	7.71	8.28	8.74
0.60	19.77	9.70	29.54	14.69	8.60	9.04	9.40
0.70	21.79	10.60	27.79	16.71	9.79	9.85	11.06
0.80	23.81	11.53	33.15	18.72	10.48	10.68	11.17
0.90	25.79	12.49	35.19	20.81	11.48	11.45	11.89
1.00	27.74	13.47	40.46	23.05	12.22	12.34	13.50

4.5 Ensemble Size Analysis

Key findings:

- Most Bayesian methods achieve optimal performance with 20-30 ensemble members.
- ECE consistently decreases with more samples across Bayesian methods.
- Bayesian models remain better calibrated than deterministic ones under input perturbations.

4.6 Practical Recommendations

- **Maximum accuracy**: MC Dropout or Deterministic MLP
- **Well-calibrated uncertainty**: Variational Inference (VI)
- **Adversarial robustness**: Any Bayesian method
- **Linear interpretability**: HMC Linear (if applicable)

5 Conclusion

`bensemble` provides a unified framework for Bayesian neural networks with multiple inference algorithms, allowing users to:

- Flexibly switch between VI, LA, PBP, and VR.
- Obtain predictive uncertainties for regression and classification.
- Sample ensembles of models from approximate posteriors.
- Integrate seamlessly with PyTorch workflows.

`bensemble` is suitable for both research and practical applications requiring principled uncertainty estimation.

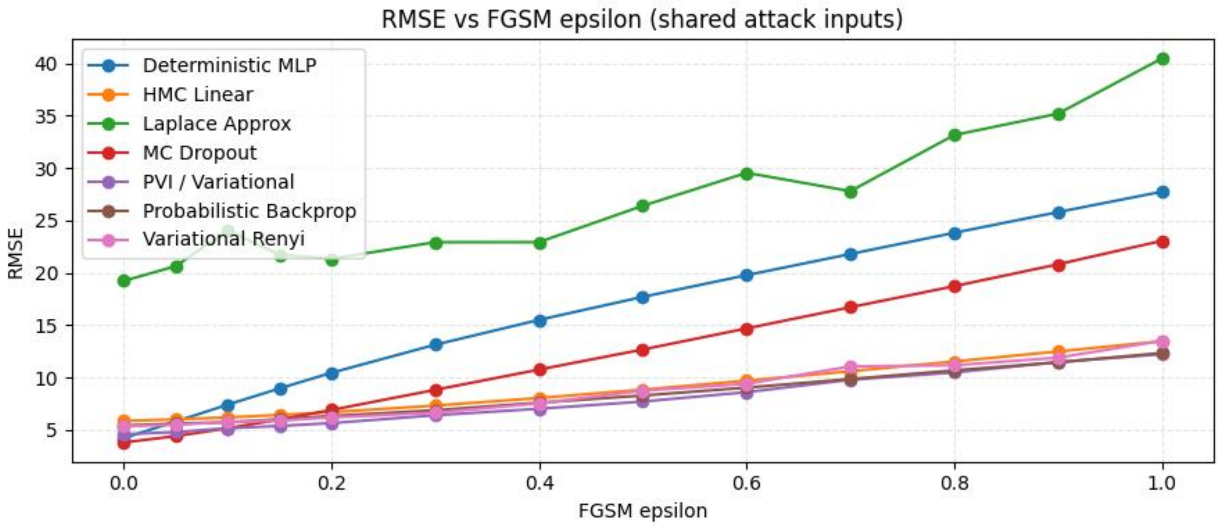

Figure 1: RMSE degradation under FGSM attacks. Bayesian methods (VI, PBP, VR) degrade more gradually than deterministic approaches.

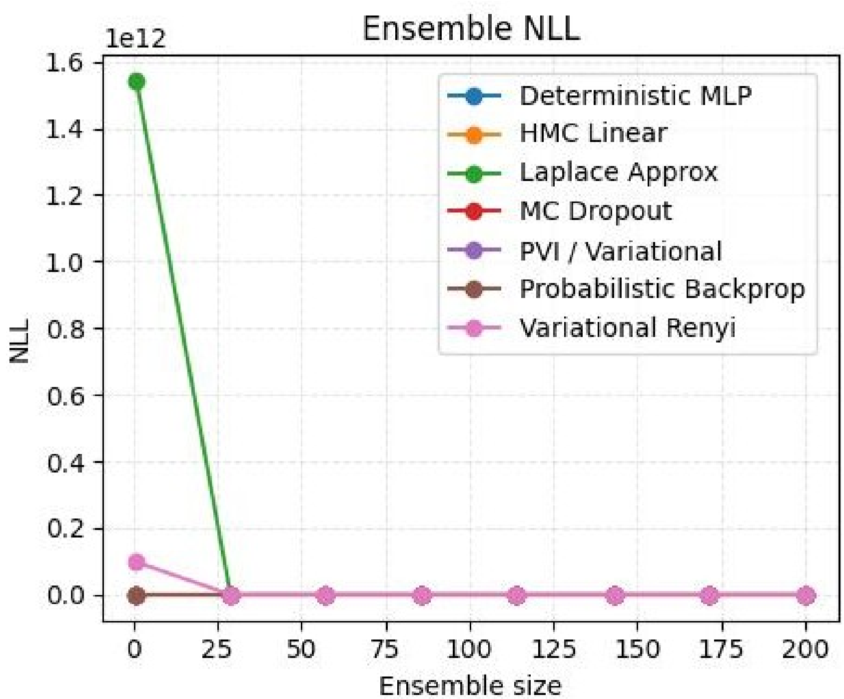

Figure 2: NLL vs Ensemble size

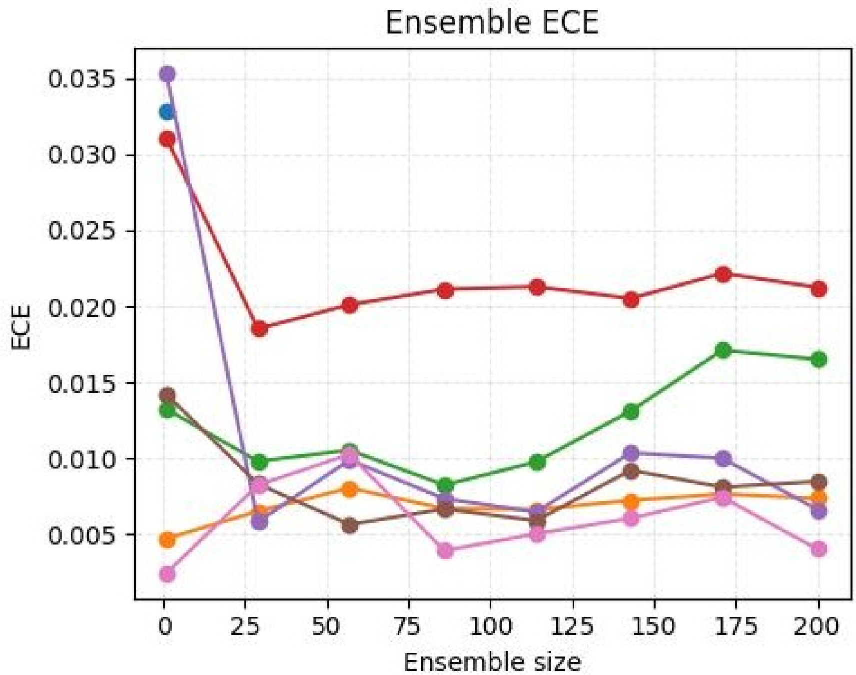

Figure 3: ECE vs Ensemble size

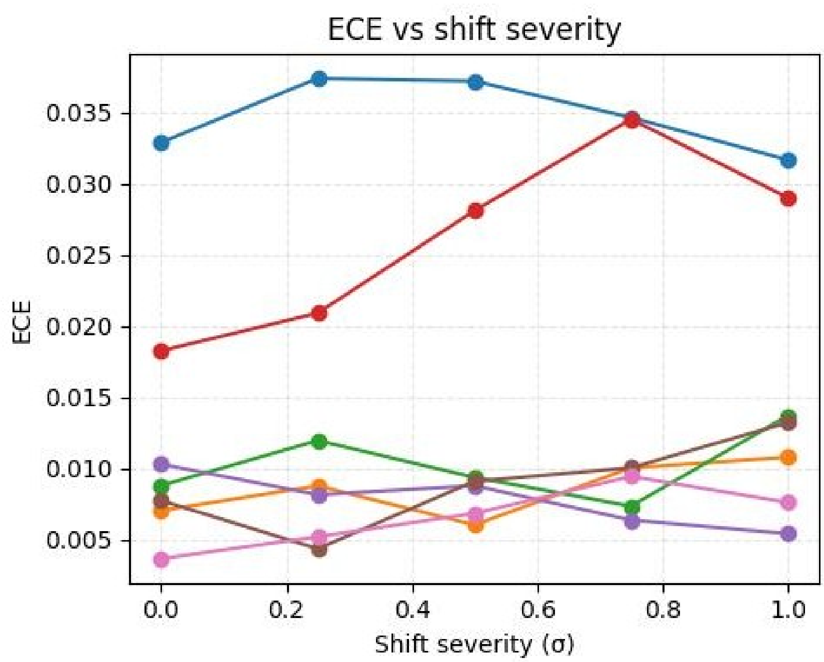

Figure 4: ECE vs Shift severity

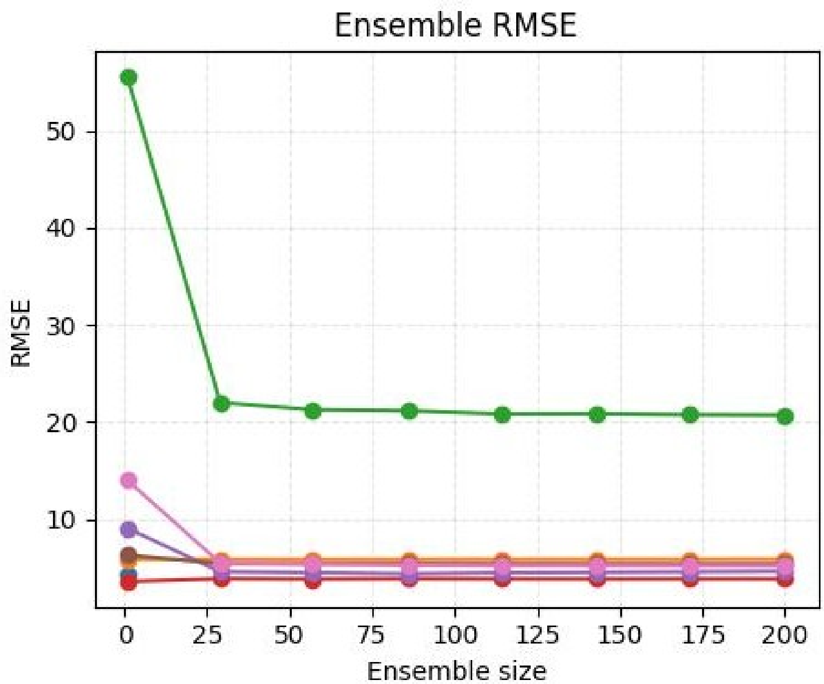

Figure 5: RMSE vs Ensemble size

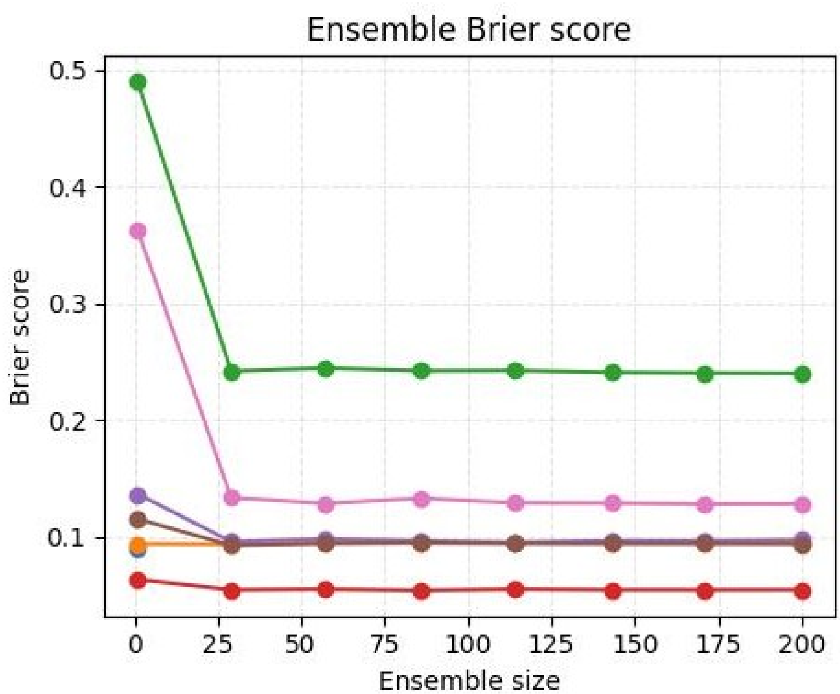

Figure 6: Brier score vs Ensemble size