We thank all the reviewers for the valuable comments and suggestions.

To Reviewer 1

#1. Regression experiments on UCI regression datasets.

We further evaluate our model on five **UCI** regression datasets and show the results in Table 1. We randomly sample 90% of each dataset for training and leave the rest for testing. We run 20 experiments for each setup with fixed random seeds and report the averaged error rate. Feature normalization is applied in the experiments. The model is a simple MLP with one hidden layer of 50 units. We set the batch size to 50, the training epoch to 200, the learning rate to 1e-4, the default L^2 to 0.003 and the initial inverse temperature τ to 300. For SGHMC-EM and SGHMC-SA, we apply the SSGL prior on the BNN weights (excluding biases) and fix $a, \nu, \lambda = 1$, $v_1, \sigma = 10$ and $\delta = 0.5$. We select b from $\{10, 100\}$, v_0 from $\{0.001, 0.01, 0.1\}$. As shown in Table 1, SGHMC-SA outperforms all baselines. Nevertheless, without smooth adaptive update, SGHMC-EM mostly performs worse than SGHMC. While with simulated annealing where $\tau^{(k)} = 300 \times r^k$, we observe further improved performance in most of the cases with the optimal rate r selected from $\{1.01, 1.015, 1.02\}$. We plan to include the distributional distance metrics and other results in the future revision.

To Reviewer 2

1. Writing suggestions.

We appreciate the suggestions on writing and are to fix them in the future revision.

2. Problem statement and solution.

This paper provides a systematic approach for conducting sparse deep learning with two innovations: (i) We propose to use the spike-and-slab prior to shrink and cluster the connection weights to two clusters, which facilitates the followed weight pruning procedure; (ii) We propose an adaptive SGMCMC algorithm to automatically tune the hyper-parameters of the spike-and-slab prior and prove the convergence of the SGMCMC algorithm rigorously. The adaptive SGMCMC algorithm is itself of interest, which can be used in many "big data" applications, for example, estimating parameters for a state-space model when the states are simulated using a SGMCMC algorithm.

3. Over-parameterization and how realistic are these assumptions.

We acknowledge over-parameterization may fit some real applications better under certain scenarios. Our assumptions are quite standard in the adaptive sampling literatures and we have already made efforts to loose the assumptions, such as Lemma 1 in the appendix. We leave the extension on weaker assumptions in the future.

To Reviewer 3

1. Use spike-and-slab to select the structure.

Thanks for the constructive comments. We include scalar-fashion pruning to strengthen the predictive power as Resnet is a complicated model. We run additional experiments on **UCI** datasets with standard BNNs, and observe iterative pruning based on suitable probability thresholds can obtain good performance. E.g., on the **Wine** dataset, when pruned with $\boldsymbol{\rho}$ lower than 0.3, the model ends up with 31% sparsity in the hidden layer and 20% sparsity in the output layer, while RMSE drops from 0.632 to 0.629. We would like to include more results and discuss the use of the spike-and-slab prior in the style of group-Lasso such that a whole pathway will be retained or pruned in the future revision.

2. Discussions on larger neural networks.

Extension of the proposed method to larger networks is straightforward. However, as implicitly assumed in our theory, the convergence of the SGMCMC algorithm is essential. For larger networks, to achieve this convergence, longer training time might be needed. Existing techniques, such as gradient noise control and temperature tuning, for accelerating SGMCMC simulations should also be helpful to this proposed method.

Dataset Hyperparameters	Boston 100/0.01/1.015	Yacht 10/0.1/1.015	Energy 10/0.001/1.01	Wine 10/0.001/1.015	Concrete 10/0.01/1.015
SGHMC	2.840±0.120	0.764±0.029	1.466±0.058	0.654±0.014	5.668±0.073
A-SGHMC	2.887±0.128	0.726±0.042	1.354±0.044	0.632±0.009	5.644±0.084
SGHMC-EM	2.872±0.125	0.748±0.048	1.412±0.028	0.770±0.011	5.632±0.057
A-SGHMC-EM	2.858±0.120	0.736±0.036	1.402±0.027	0.638±0.008	5.474±0.096
SGHMC-SA	**2.838±0.115**	**0.746±0.037**	**1.366±0.034**	**0.632±0.010**	**5.372±0.071**
A-SGHMC-SA	**2.780±0.108**	**0.716±0.036**	**1.270±0.029**	**0.628±0.008**	5.438±0.079

Table 1: Average testing performance and standard deviation of RMSE (Root Mean Square Error), with b in the Beta distribution, v_0 in the SSGL prior, and r in the simulated annealing (Hyperparameters $b/v_0/r$).