We thank all reviewers for taking the time to provide detailed feedback and valuable suggestions for our work. We address the reviewers' detailed comments below:

• **Comparison with Dasgupta et al. (2017)**. We thank Reviewer # 1 for pointing out this interesting work. Both our model and Dasgupta et al. (2017) use flexible transformations to augment a parametric model, but the two approaches differ in (1) the exact type of transformation used and (2) the inference method for the ensemble parameters ω. Specifically, the transformation in our method $G_{\mathbf{x}} : \Phi(.|\mathbf{x}) \to F^*(.|\mathbf{x})$ is an operator between distribution functions, while the transformation in Dasgupta et al. (2017) is a diffeomorphism $\Gamma_{\mathbf{x}} : y \to y'$ acting on the response space $y, y' \in \mathcal{Y}$. As a result, the PDFs for the two models are $f(y|\mathbf{x}, \mu) = \frac{\partial}{\partial y}\Phi\big(\mathbf{\Gamma}_{\mathbf{x}}(y)|\mathbf{x}, \hat{\boldsymbol{\omega}}\big) = \phi\big(\mathbf{\Gamma}_{\mathbf{x}}(y)|\mathbf{x}, \hat{\boldsymbol{\omega}}\big) * \boldsymbol{\gamma}_{\mathbf{x}}(y)$ (Dasgupta et al), and $f(y|\mathbf{x}, \mu) = \frac{\partial}{\partial y}\mathbf{G}_{\mathbf{x}}\big(\Phi(y|\mathbf{x}, \boldsymbol{\omega})\big) = \boldsymbol{g}_{\mathbf{x}}\big(\Phi(y|\mathbf{x}, \boldsymbol{\omega})\big) * \phi(y|\mathbf{x}, \boldsymbol{\omega})$ (this work), where $g_{\mathbf{x}} = \frac{\partial}{\partial \Phi} G_{\mathbf{x}}$ and $\gamma_{\mathbf{x}} = \frac{\partial}{\partial y}\Gamma_{\mathbf{x}}$. As shown, the two models share a similarity in that their PDFs are the product of ϕ (a Gaussian PDF) and the derivative of a transformation function. However, their PDFs' exact expressions are in fact different. Comparing the two models, both models are flexible in learning the data-generating F^*. However the construction in this work is better posed for uncertainty quantification in ensemble learning for two reasons: (1) Interpretability: $G_{\mathbf{x}}$ is a transformation that directly "fixes" the original likelihood Φ, which allow us to diagnose the misspecification in original likelihood by inspecting $G_{\mathbf{x}}[\Phi] - \Phi$. This is not easy to do under the formulation of Dasgupta et al; (2) Estimation quality for the ensemble weights $\boldsymbol{\omega}$. Since Dasgupta et al. (2017) estimates $\boldsymbol{\omega}$ under the original parametric model, the resulting estimate $\hat{\boldsymbol{\omega}}$ can be biased if the original likelihood is misspecified. In comparison, our model estimates $\boldsymbol{\omega}$ under the flexible BNE likelihood to avoid biased inference. In the final manuscript, we will include above discussion in the Related Work section (at the end of Section A.1).

• **Implementation difficulty of CGP and HMC.** Since CGP is simply a GP model with probit-based likelihood penalties, implementing CGP is in fact not difficult. Any GP model can be converted to a CGP model by adding log probit terms to the GP log likelihood (e.g., in TensorFlow Probability, this can be done in one line as `gp_likelihood + tfd.Normal.log_cdf(g)`) and then apply MCMC as before. The HMC method proposed in this work in fact does not require hand tuning. It uses an automated adaptive step size scheme that is readily available in TensorFlow Probability (see Appendix Section C.3). Alternatively, one can also use the *No U-Turn Sampler* (NUTS) implemented in Stan.

• **Flexibility in the mean function.** As shown in equation (5) of the main text, BNE's mean function consists of the original ensemble, the residual process $\boldsymbol{\delta}$ and a bias correction term due to $\mathbf{G}$. In addition to the base predictors, the flexibility of BNE's mean function is mainly driven by the residual process $\boldsymbol{\delta}$, and domain experts can select a flexible kernel for $\boldsymbol{\delta}$ to best approximate the data-generating function of interest. (e.g., a RBF kernel to approximate arbitrary continuous functions over a compact support (Micchelli et al., 2006)). In the manuscript, we will include this discussion when first introducing the residual process (line 103-106 of the main text)

• **Extension to moderately high dimensional predictors.** The BNE framework can be naturally extended to high dimension by choosing kernel functions for $\boldsymbol{\delta}$ and $\mathbf{G}$ that are suitable for high-dimensional problems. Example choices include the additive kernel (Durrande et al., 2011) or (deep) neural network kernel (Bach, 2014; Lee et al., 2017). Alternatively, one could also build variable selection into the model using shrinkage priors such as the Automatic Relevance Determination (ARD), spike-and-slab, or Horseshoe (Bobb et al., 2015; Vo and Pati, 2017). In the final manuscript, we will include the above discussion in the Conclusion and Future Work section.

• **In the experiment, BNE is by construction more expressive than BAE.** We thank Reviewer # 3 for highlighting an important part of our experiment design. Indeed, BAE is an abalated version of BNE (i.e. without $\mathbf{G}$). The goals to include BAE are to see (1) if $\mathbf{G}$ leads to significant improvement in large sample sizes, and (2) if $\mathbf{G}$ severely overfits data and leads to worse performance in small sample sizes. Figures 4-5 suggest that the former is true, but not the latter.

• **In Appendix H2, the low RMSEs of BNE and BME are within one another's standard deviations.** We thank Reviewer # 3 for making this observation. Indeed, this result is expected and is consistent with what we observed in the simulation experiment (Figures 4-5): In a small sample (where uncertainty is high), BNE is competitive with BME in prediction performance, while providing several advantages when the goal is uncertainty quantification (i.e., uncertainty decomposition in Figure 5, and model diagnosis in Figure H.3).

• **Extend empirical comparison.** Following suggestions by Review # 1 and # 3, in the final manuscript we will extend the empirical comparisons in Figure 4-5 to include two more models: the popular Deep Ensemble (Lakshminarayanan et al., 2017) and the classic `npcdist` from the `np` package. Deep Ensemble and `npcdist` fits a finite Gaussian mixture and a kernel smoother to the data, respectively. In terms of performance, Deep Ensemble is expected to perform similarly to BME which also uses a Gaussian mixture. `npcdist` is expected to perform similarly to BNE in this 1D experiment, however generalizing kernel density estimators to higher dimensions is usually more difficult (Scott, 2015). We feel the two scalable GP approaches cannot be directly compared to BNE since they still rely on a parametric model likelihood from a known distribution family. However, we do note that BNE can be made scalable by estimating $\mathbf{G}$ and $\boldsymbol{\delta}$ using the variational inducing point method in Hensman et al. (2015). However, the disadvantage of this approach is that the uncertainty estimate may be inaccurate since the variational family usually does not fully capture the posterior distribution. Research into scalable inference method that provides good uncertainty quantification is an important future direction of this work. We will include the above discussion in the Conclusion and Future Work section.