Systems & Control Transactions

Original Research Article
Peer Reviewed Conference Proceeding
ESCAPE 36 - European Symposium on Computer Aided Process Engineering
Sheffield, UK. 21-24 June 2026

PSE PRESS

Variational Bayesian Neural Networks for Modelling and Uncertainty Quantification in Bioprocessing

George Spencer[a], Harini Narayanan[b], Claus Wirnsperger[b], Alessandro Butté[b], Cleo Kontoravdi[a], Maria M. Papathanasiou[a*]

[a] The Sargent Centre for Process Systems Engineering, Imperial College London, London, United Kingdom, SW7 2AZ
[b] DataHow AG, Hagenholzstrasse 111, 8050, Zürich, Switzerland
* Corresponding Author: maria.papathanasiou11@imperial.ac.uk.

ABSTRACT

Biopharmaceutical manufacturing requires systematic identification, understanding and control of critical process parameters (CPPs) and critical quality attributes (CQAs) While deterministic machine learning models have proven valuable for describing complex systems and providing insights into process behaviour, the extension to probabilistic frameworks allows for the capture of intrinsic biological and process variability, improving robustness, understanding and safety. This work introduces a framework for autoregressive variational Bayesian neural networks (BNNs) that integrate uncertainty quantification into data-driven modelling. The approach is demonstrated on a multicolumn countercurrent solvent gradient purification (MCSGP) process for monoclonal antibody purification. The BNN is trained autoregressively on high fidelity data generated using a detailed mechanistic process model data. A heteroscedastic variance head is included to model signal-dependent observation noise, while the epistemic uncertainty is captured through stochastic network parameters. Model performance is assessed using both deterministic accuracy metrics - MSE and probabilistic scores namely coverage and continuous ranked probability scores (CRPS). Results show that the proposed method accurately recovers noiseless concentration profiles while maintaining well calibrated predictive distributions. Additionally, the framework is tested under multiple noise regimes to test robustness of the proposed approach.

Keywords: Modelling and Simulations, Algorithms, Bayesian Neural Networks, Variational Inference, Uncertainty Quantification, Chromatography

INTRODUCTION

Biologics, such as monoclonal antibodies (mAbs), play a central role in the treatment of diseases, including cancers and autoimmune disorders. The commercial importance of mAbs is substantial, with six of the top ten best pharmaceutical products being mAb-based therapies [1]. As a result, there is an increasing demand for cost minimisation and high production efficiency, while regulators are pushing for transparency, robustness and explainability through initiatives such as Quality by Design (QbD) and the drive towards Biopharma 4.0. In this context, process modelling provides a valuable tool for supporting design, operation and decision making, by improving understanding and operability of complex bioprocesses.

Data-driven modelling allows engineers with access to process data and often limited mechanistic process knowledge to model unknown phenomena. This is pertinent in upstream bioprocessing where many underlying mechanisms remain partially characterised. In downstream processing, where process knowledge is richer, data-driven modelling can offer significant improvement in computational time for inference, making them attractive for online monitoring, optimisation, and control. However, in the pursuit of Biopharma 4.0, deterministic model inference alone is insufficient to support risk-aware decision making. Biological systems are often characterised by substantial inherent variability and therefore process decisions must account for uncertainty to satisfy stringent regulatory constraints and ensure consistent product quality.

In this setting, Bayesian Neural Networks (BNNs) offer a promising solution, by retaining the predictive

capabilities of artificial neural networks, while providing uncertainty estimates via probabilistic inference. In contrast to deterministic models, BNNs provide predictive distributions rather than point estimates, enabling quantification of uncertainty. This includes aleatoric uncertainty arising from measurement noise or process variability, and epistemic uncertainty associated with limited data and uncertainty in model parameters. Gaussian Processes (GP) are commonly utilised for this purpose, however lack scalability in dynamic systems such as bioprocessing [2].

Many algorithms exist for the training of BNNs, with Markov Chain Monte Carlo (MCMC) methods often being regarded as the gold standard [3]. Despite their wide application, these exact methods can be slow to converge and may become computationally prohibitive for large networks. Approximate inference methods offer a scalable alternative that is compatible with gradient based optimisation. In this work, the applicability of variational inference for autoregressive BNNs in bioprocess modelling is assessed using ion-exchange chromatography as a use case. The proposed framework is designed to enable probabilistic prediction and decompose aleatoric and epistemic uncertainties. This distinction is important, as only epistemic uncertainty can be reduced by more data and targeted experimentation, whereas the aleatoric uncertainty represents intrinsic process variability that must instead be managed through robust operation and control.

THE USE CASE OF MULTICOLUMN COUNTERCURRENT SOLVENT GRADIENT PURIFICATION (MCSGP) PROCESS

The case study of interest is the twin-column Multicolumn Countercurrent Solvent Gradient Purification (MCSGP) process presented by Aumann and Morbidelli [4]. The process is an ion exchange chromatography used as a purification step in mAb production. The system separates three components, namely the product, the weak impurities, and the strong impurities.

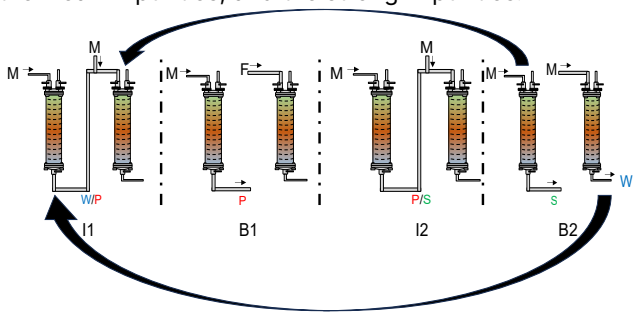

Figure 1: Diagram of MCSGP operating cycle

Operation proceeds through four phases: Two interconnected (I1 and I2) and two batch (B1 and B2) (Figure 1). During the interconnected phases, outlet streams containing off specification product fractions are recycled between columns to enhance recovery, while batch phases allow for the elution of high purity product. Strong and weak impurity fractions are discarded. After the cycle shown in Figure 1, the roles of the columns reverse and the process is repeated.

A high fidelity, experimentally validated, process model, originally developed by Müller-Späth et al. [5] was utilised to generate data of the MCSGP process. This lumped kinetic model, comprises over 4000 ordinary differential equations [6], after spatial discretization, representing a computational bottleneck for real time applications and uncertainty quantification (UQ).

METHODOLOGY

Data Generation and Process Model

Data is generated according to the strategy proposed by Michalopoulou et al. [7]. In which Sobol sampling is used to sample nine design variables through ranges presented in Table 1.

Table 1: Inputs and Bounds for Data Generation

Variable	Lower Bound	Upper Bound	Units
Weak Feed Concentration	0.05	0.08	mg/ml
Product Feed Concentration	0.3	0.5	mg/ml
Strong Feed Concentration	0.03	0.05	mg/ml
Batch Flowrates	0.1	1.0	ml/min
I1 Flowrate	0.1	1.0	ml/min
I2 Flowrate	0.1	1.0	ml/min
Modifier Feed Concentration	1.0	3.0	mg/ml
Initial Modifier Concentration Phase I1 column 1	2.0	3.0	mg/ml
Initial Modifier Concentration Phase I1 column 2	0.8	1.2	mg/ml

The model is simulated to Cyclic Steady State (CSS) and input-output data, consisting of these nine design variables and the output concentrations of each species for a full CSS cycle is taken to create the dataset. For quantifying aleatoric uncertainty in this methodology, heteroscedastic noise is added to the measured concentrations of species.

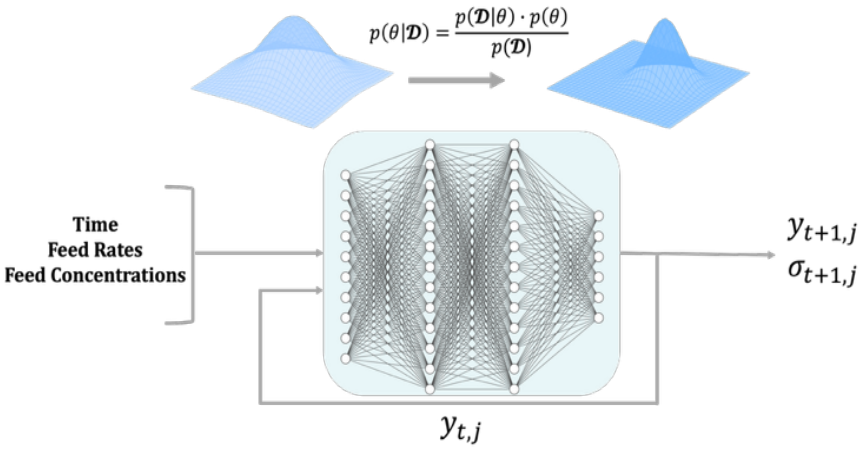

Figure 2. Autoregressive Bayesian Neural Network Architecture

Autoregressive Bayesian Neural Network

A dataset of multivariate time sequences $\left\{\left(x_{1:T}^{(n)}, y_{1:T}^{(n)}\right)\right\}_{n=1}^{N}$, where $x_t \in \mathbb{R}^{d_x}$ are exogenous inputs and $y_t \in \mathbb{R}^{d_y}$ are the targets, is constructed. The model is trained autoregressively to predict $\hat{y}_t^{(n)} = f_\theta\left(x_{t-C:t}^{(n)}, \hat{y}_{t-C:t-1}^{(n)}\right)$. Using a fixed context length C. Predictions are generated sequentially, with the previous C predicted outputs fed back as inputs for subsequent time steps.

The predictor, f_θ, is a Bayesian neural network with a factorised Gaussian variational posterior over weights and biases. For each layer ℓ, the posterior is parameterised as shown in Eq. (1) and (2).

$$q(W_\ell) = \mathcal{N}\left(\mu_{W_\ell}, \sigma_{W_\ell}^2\right) \quad (1)$$

$$q(b_\ell) = \mathcal{N}\left(\mu_{b_\ell}, \sigma_{b_\ell}^2\right) \quad (2)$$

Where W_ℓ and b_ℓ denote the weights and biases of layer ℓ of the network. The standard deviations are parameterised via Eq. (3).

$$\sigma = log(1 + exp(\rho)) \quad (3)$$

With, ρ, a learned parameter for each weight and bias, to ensure positivity and improved training stability. The BNN includes a heteroscedastic variance head that predicts the log-variance, $log\left(\sigma_{t,j}^2\right) \forall j = 1, \ldots, d_y$, for each output dimension at each timestep. A schematic of the network is displayed in Figure 2.

Variational Inference

The training of the BNN approximates the posterior distribution over network parameters (θ) given by Eq. (4)

$$p(\theta|\mathcal{D}) = \frac{p(\mathcal{D}|\theta)p(\theta)}{p(\mathcal{D})} \quad (4)$$

Where, $p(\mathcal{D})$ is the evidence or the marginal likelihood shown in Eq. (5).

$$p(\mathcal{D}) = \int p(\mathcal{D}|\theta)p(\theta)\, d\theta \quad (5)$$

Which requires integration over all possible parameter realisations and is non-conjugate and therefore intractable to compute analytically requiring an approximate inference algorithm. In this case variational inference was chosen as it can be seamlessly integrated into gradient descent optimisation algorithms.

Variational inference introduces a parameterised variational distribution and seeks a member of this family that is the closest to the true posterior, defined by Eq. (6) the Kullback-Leibler (KL) divergence.

$$q_\phi^*(\theta) = \arg\min_{q_\phi} D_{KL}\left(q_\phi(\theta) \,\middle|\, p(\theta \mid \mathcal{D})\right) \quad (6)$$

Direct evaluation of Eq. (6) is infeasible, as it requires the unknown posterior, therefore optimisation is performed indirectly by maximising the evidence lower bound (ELBO) defined in Eq. (7).

$$\mathcal{L}_{\mathcal{ELBO}} = \mathbb{E}_{q_\phi(\theta)}[\log p(\mathcal{D} \mid \theta)] - \mathrm{KL}\left(q_\phi(\theta)|p(\theta)\right) \quad (7)$$

The ELBO defines a lower bound on the log-evidence and is related to the intractable KL divergence in Eq. (8).

$$log\ p(\mathcal{D}) = \mathcal{L}_{\mathcal{ELBO}} + \mathrm{KL}\left(q_\phi(\theta)||p(\theta|\mathcal{D})\right) \quad (8)$$

Since the KL divergence on the right-hand side of Eq. (8) is non-negative, maximising the ELBO is the equivalent to minimising the KL divergence between the variational approximation and the true posterior. The ELBO consists of two competing terms, the expected log-likelihood term that encourages the model to explain the observed data, while the KL term acts as a regulariser towards the prior distribution.

The variational loss function applied in this algorithm is defined by Eq. (9).

$$\mathcal{L} = -E_{q(\theta)}\left[\sum_{t=C+1}^{T} \log p\left(\hat{y}_t \mid \hat{\mu}_t, \hat{\sigma}_t^{\,2}\right)\right] + \beta\,\mathrm{KL}(\mathrm{q}(\theta)||\mathrm{p}(\theta)) \tag{9}$$

To enable gradient based optimisation of the ELBO, the expectation with respect to the variational posterior is approximated using Monte Carlo sampling combined with the reparameterization trick [8]. Specifically, for layer ℓ, weights and biases are sampled from Eq. (10) and Eq. (11).

$$W_\ell = \mu_{W_\ell} + \sigma_{W_\ell}\epsilon_\ell \tag{10}$$

$$b_\ell = \mu_{b_\ell} + \sigma_{b_\ell}\epsilon'_\ell \tag{11}$$

Where $\epsilon \sim \mathcal{N}(0, \mathrm{I})$. This formulation expresses the stochastic network parameters as deterministic functions of the variational parameters and auxiliary noise variables (ϵ), allowing the Monte Carlo estimated ELBO to be computed via automatic differentiation.

Model Training

The training data consists of 400 cyclic steady state sequences generated from the mechanistic model, with the nine varied design variables and three output concentration profiles. A further 200 sequences are used for validation, and 90 sequences are held out for testing and visualisation.

The BNN receives 13 input features at each time step. These consist of the nine design variables, the most recent prediction of the three outlet concentrations, and time. The network contains three fully connected hidden layers with sizes 64, 64 and 32, respectively, each followed by ReLU activations. The architecture and model hyperparameters were optimised using Tree Structured Parzen Estimator based Bayesian Optimisation [9]. The final output layer consists of six neurons, predicting the mean and heteroscedastic noise estimate for each concentration.

During training the KL divergence term is computed independently for each layer and scaled by the batch size to avoid batch-size dependence or the variational objective. To improve optimisation stability, the KL weight β, is linearly annealed from 0 to 1 during an initial warmup phase. After warmup, full variational regularisation is applied.

The ELBO is approximated using $\mathcal{K}$ Monte Carlo samples of the network parameters. For each particle independent random keys are used to sample weights and biases, producing $\mathcal{K}$ autoregressive rollouts.

During training teacher forcing is employed to stabilise the learning. At each time step, the history value fed back to the model is chosen according to Eq. (11).

$$y_t^{hist} = \begin{cases} y_t, & with\ \text{probability}\ \ p_{TF} \\ \hat{\mu}_t, & with\ \text{probability}\ (1 - p_{TF}) \end{cases} \tag{11}$$

Where p_{TF} is the probability of teacher forcing. This probability is annealed linearly over a fraction of the total optimisation steps in training, gradually transitioning from fully guided rollouts towards fully self-predicted trajectories.

When physical bounds are supplied the predictive mean is mapped through a sigmoid transformation (Eq. 12)

$$\hat{\mu}_t = \mathrm{a} + (b - a)\sigma(\hat{\mu}_t^{raw}) \tag{12}$$

Where a and b represent the lower and upper bounds and $\sigma(\cdot)$ is the sigmoid function. In this case bounds are chosen as [0, 1] mg/ml. This deterministic and differentiable mapping enforces hard constraints on the mean prediction while preserving gradient based optimisation of the variational loss. Since epistemic uncertainty arises from variability in the network parameters, constraining the sampled mean predictions also constrains the resulting epistemic uncertainty distribution. In contrast, the heteroscedastic variance head remains unbounded, allowing the model to freely represent the aleatoric noise.

Model Inference

At inference time, predictive uncertainty is obtained via Monte Carlo sampling of the learned variational posterior. In all experiments, 1000 parameter samples are drawn. For each sample, a full autoregressive rollout of the concentration trajectories is generated.

The predictive mean is computed as the mean trajectories across all samples. Epistemic uncertainty is quantified as the variability across these sampled mean predictions. This sampling-based formulation allows the predictive distribution to be non-Gaussian and enables direct estimation of prediction quantiles.

Aleatoric uncertainty is obtained from the heteroscedastic output head. Since the variance prediction itself depends on stochastic network parameters, the aleatoric estimate exhibits epistemic uncertainty as a second order effect. The total predictive uncertainty is therefore composed of both epistemic and aleatoric contribution, combined under a conditional independence assumption.

Computation

In this work the BNN architecture and training algorithm were constructed using JAX, to enable efficient automatic differentiation and hardware acceleration [10]. Just-in-time (JIT) compilation was used extensively to optimise the autoregressive rollout and ELBO computation, allowing entire sequence scans to be compiled into fused kernels. Vectorisation was performed across batches and Monte Carlo particles to enable efficient parallel evaluation of stochastic network samples during training and inference. Gradient descent was performed with AdamW using the Optax suite [11], [12]. All experiments are performed using an NVIDIA RTX 4090 GPU.

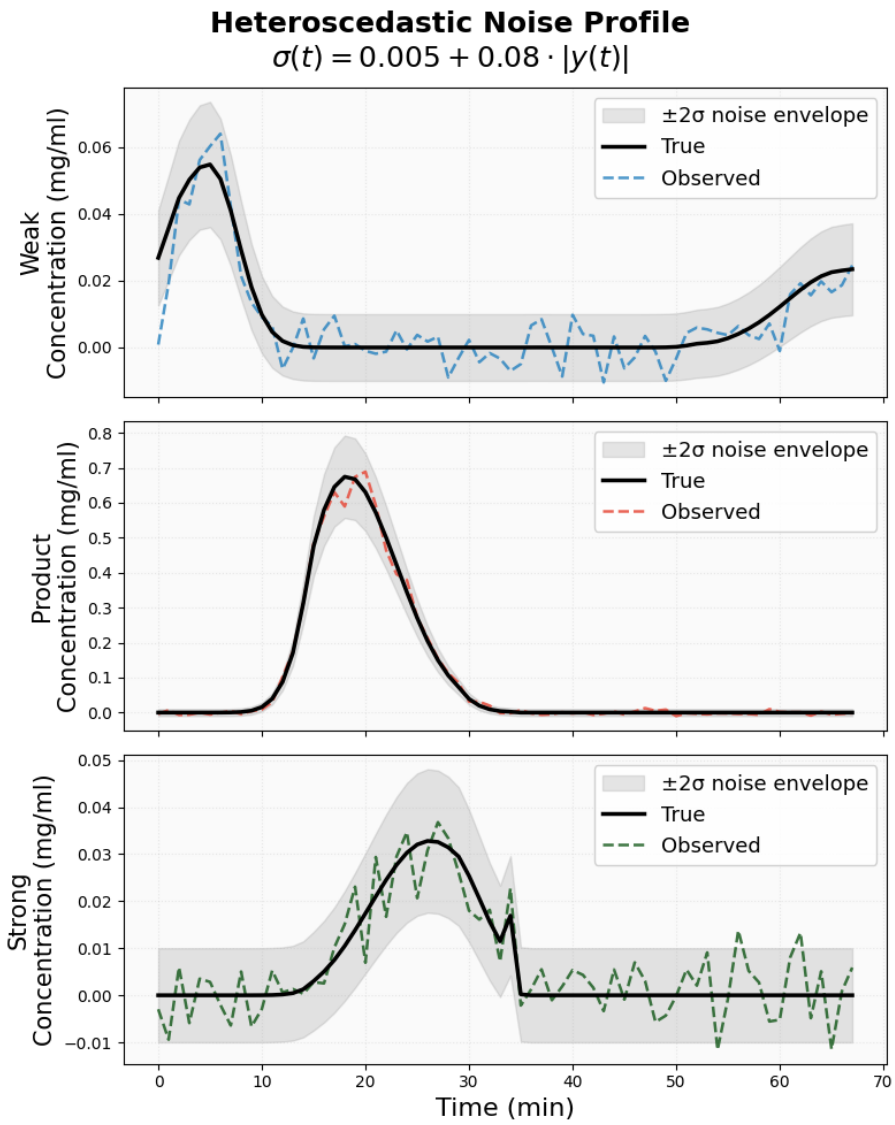

Figure 3. Nominal heteroscedastic noise profiles applied to outlet concentrations.

RESULTS AND DISCUSSION

Nominal Test Case

The nominal test case takes the train, test and validation sets and applies noise to the outputs before training and testing. The noise model in this scenario is Gaussian heteroscedastic noise with noise standard deviation defined as a sum of a constant baseline component and a signal magnitude dependent component given by Eq. (13)

$$y_{t,j}^{noisy} = y_{t,j} + \epsilon_{t,j},\ \epsilon_{t,j} \sim \mathcal{N}\left(0, \left(\sigma_0 + \alpha|y_{t,j}|\right)^2\right) \quad (13)$$

Where σ_0 represent the baseline noise and α controls the signal dependent noise, and where j represents each component. For this nominal case $\sigma_0 = 0.005, \alpha = 0.08$ was used, an example profile is shown in Figure 3.

Validation metrics of deterministic prediction accuracy and uncertainty quantification capability are displayed in Table 3. The mean squared error (MSE) quantifies the accuracy of the predictive mean, while the empirical coverage, the frequency at which the model's predicted intervals contain the observed values, evaluates the calibration of the predictive intervals. The continuous ranked probability score (CRPS) [13] is a proper scoring rule that assess the difference between the predicted and observed cumulative distributions.

Figure 4 displays a representative outlet concentration prediction profile from the test set. The predictive means closely follow the true, noiseless, profiles. While the predictive uncertainty bands capture the epistemic and aleatoric uncertainty throughout time. A subset of

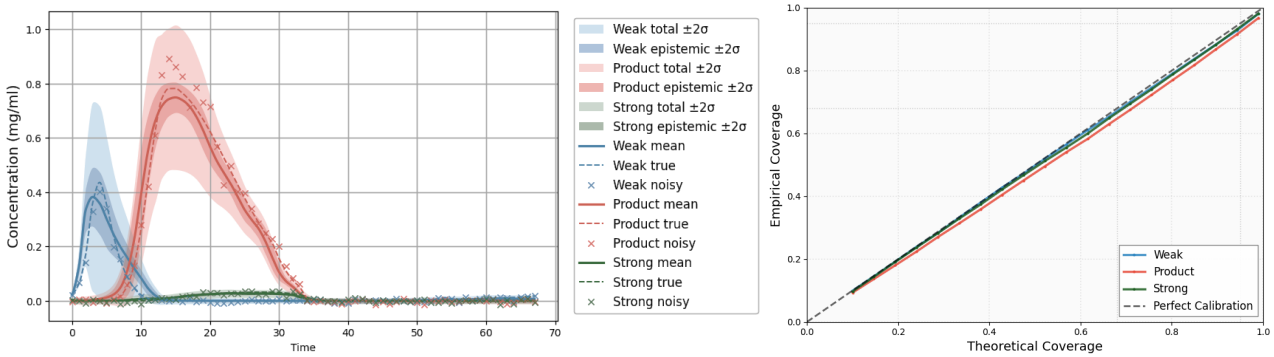

Figure 4. Left**:** Predicted outlet concentration profiles for representative test sequence. Right: Calibration plots for nominal case, over all test set sequences.

noisy realisations are marked with a cross to demonstrate the capability of the model to capture the heteroscedastic noise.

Table 2: Validation Metrics for Nominal Case

Metric	Weak Impurities	Product	Strong Impurities	Units
MSE	6.32e-4	1.79e-3	1.37e-4	$[mg/ml]^2$
95% Coverage	93.4%	92.2%	93.8%	-
CRPS	5.99e-3	1.13e-2	4.03e-3	mg/ml

The probabilistic calibration of the predictive distributions is assessed via calibration curves shown in Figure 4, across all sequences in the test set, for each component. The empirical coverage closely matches the nominal confidence levels across all components, with slight overconfidence in the product profiles. Overall, the nominal test case demonstrates that the predictive distribution from the BNN achieves accurate predictions and is able to filter noise to predict close to true noiseless values with low MSE, within its predicted epistemic uncertainty. While the noise levels are accurately captured by the heteroscedastic head, with a mean absolute error in noise levels of 8e-4.

Secondary Noise Model

To better reflect realistic UV sensor behaviour typically observed in chromatographic systems, and test the algorithm in different, more complex and realistic noise regimes, a secondary noise model is introduced. Unlike the nominal case, which assumes temporally independent noise observations, this model incorporates additional structure, including temporal correlation and baseline drift. In this case there is an additive noise given by Eq. (14)

$$y_{t,j}^{noisy} = y_{t,j} + \epsilon_{t,j} \tag{14}$$

Where $\epsilon_{t,j}$ is composed of three contributions:

(i) The heteroscedastic white noise displayed in Eq. (15)

$$\epsilon_{het,t,j} \sim \mathcal{N}\left(0, \left(\sigma_0 + \alpha|y_{t,j}|\right)^2\right) \tag{15}$$

(ii) Temporally correlated first order autoregressive noise ($AR(1)$) shown in Eq. (16)

$$\epsilon_{ar,t,j} = \rho\epsilon_{ar,t-1,j} + \eta_{t,j},$$
$$\eta_{t,j} \sim \mathcal{N}\left(0, (\sigma_{ar})^2\right) \tag{16}$$

Where $\rho \in (-1, 1)$ controls the strength of temporal correlation and σ_{ar} determines the magnitude of the correlated noise.

(iii) Baseline drift modelled as a random walk given by Eq. (17)

$$d_{t,j} = d_{t-1,j} + \xi_{t,j} \tag{17}$$

with $\xi_{\{t,j\}}$ denoting a zero mean Gaussian increment.

Figure 5 illustrates the new noise profiles on an example sequence. This test case uses: $\sigma_0 = 0.002, \alpha = 0.02, \rho = 0.95, \sigma_{ar} = 0.002$.

The model architecture, training epochs and data set size are identical to the nominal test case for comparability. Quantitative validation metrics obtained under the secondary regime are reported in Table 3. Compared to the nominal case, point prediction error increases modestly across all components, reflecting the increased complexity in filtering temporally correlated noise and baseline drift. Despite this, the empirical 95% coverage remains close to the nominal level across all components, indicating that the predictive uncertainty adapts to the new noise regime. The increase in CRPS reflects a reduced predictive sharpness and increased uncertainty, which is consistent with the more challenging noise model.

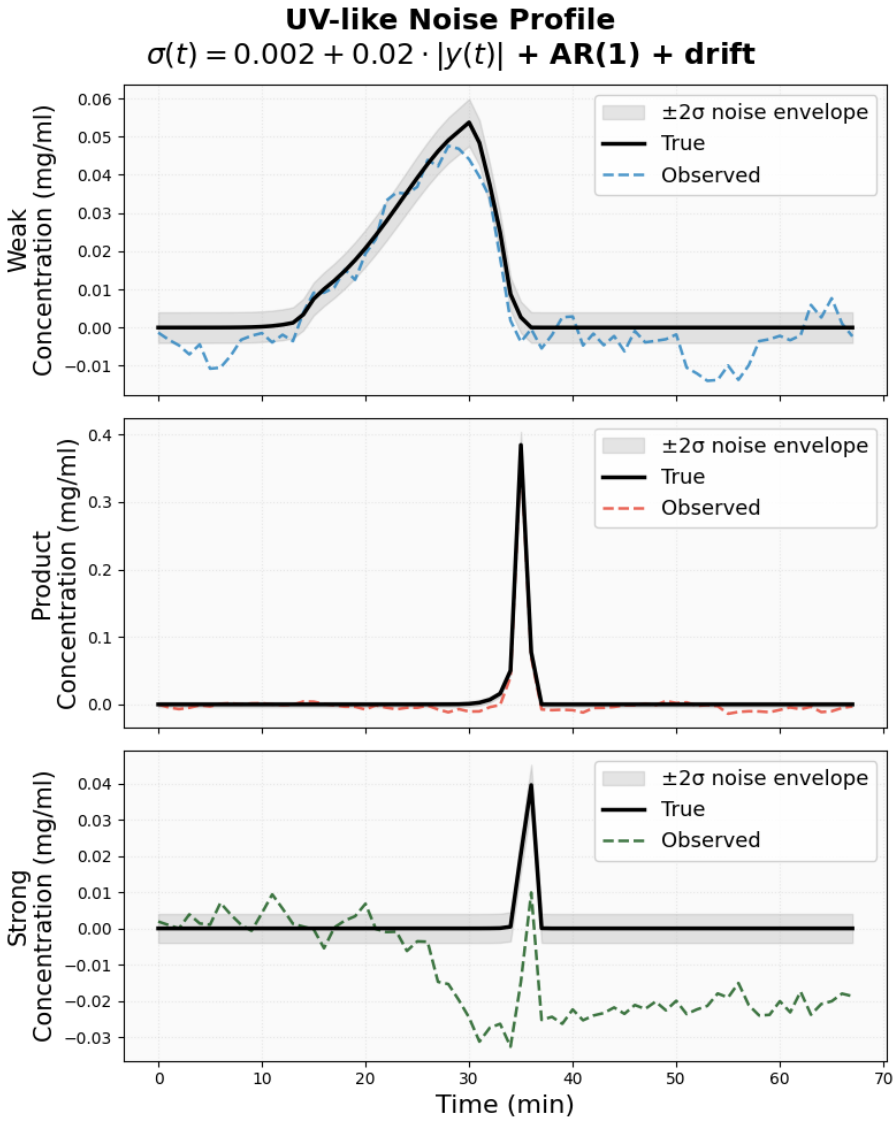

Figure 5. Secondary noise model profiles applied to outlet concentrations.

Figure 6 shows a representative prediction under the secondary noise model. Despite the added complexity of the noise regime, the predictive mean is still able to track the noiseless truth, while the uncertainty bands adapt to the new regime. The predictive uncertainty increases in regions effected by baseline drift, reflecting reduced confidence in the inferred signal.

The right-hand side of Figure 6 shows the calibration of the network remains tight across all components, with slight overconfidence in some regions.

Table 3: Validation Metrics for Secondary Case

Metric	Weak Impurities	Product	Strong Impurities	Units
MSE	7.54e-4	2.73e-3	2.40e-4	$[mg/ml]^2$
95% Coverage	91.8%	92.6%	90.6%	-
CRPS	1.14e-2	1.85e-2	9.23e-3	mg/ml

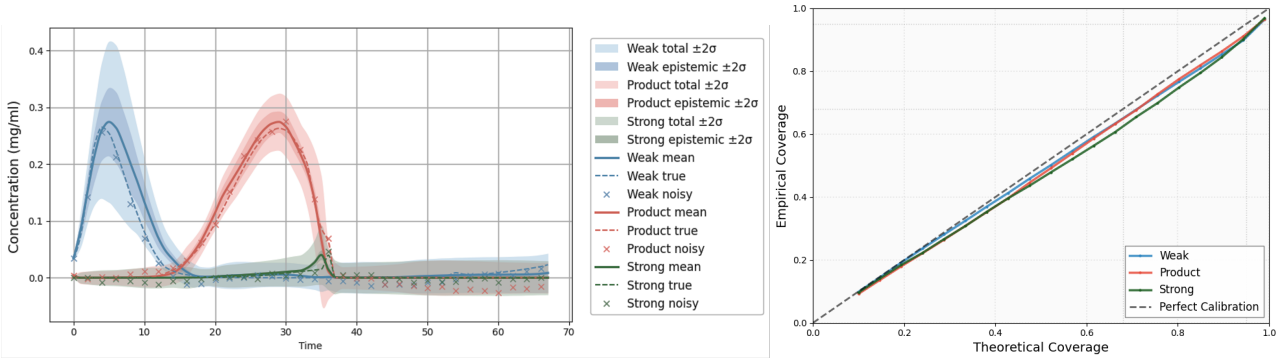

Figure 6. Left: Predicted outlet concentration profiles for representative test sequence, under secondary noise model. Right: Calibration plots for secondary case, over all test set sequences.

CONCLUSION AND OUTLOOK

This work has demonstrated the applicability of autoregressive Bayesian neural networks trained via variational inference for modelling and uncertainty quantification in downstream bioprocessing operations. Using the MCSGP case study, the proposed framework was shown to produce accurate predictive means while also providing well calibrated uncertainties. The decomposition of predictive uncertainty into epistemic and aleatoric components enables meaningful interpretation of model confidence.

The nominal test case demonstrated that the BNN is able to accurately recover noiseless concentration profiles from heteroscedastic observations, while maintaining reliable uncertainty calibration. The introduction of a secondary, more complex, noise model showed the framework remains robust under more challenging, but practically relevant conditions.

A key strength of the proposed approach is the autoregressive formulation, which allows the model to generate predictions sequentially without reliance on cyclic steady state conditions. Looking forward, the framework will be tested under fully dynamic operation or transient disturbances, allowing deployment in online monitoring, optimisation or control.

A particularly promising direction in the application of the framework is in upstream bioprocessing, where biological variability arising from cellular mechanisms introduces substantially higher uncertainty than downstream steps. Eventually, targeting an uncertainty-aware holistic modelling pipeline connecting the upstream and downstream processing units.

Training data requirements remain a key limitation for Bayesian Neural Networks over using Gaussian Processes. Future work will explore smaller models and lower training data availability, to demonstrate applicability in the data sparse environments often found in bioprocessing. Particularly looking at the effect of training data availability on epistemic uncertainty estimates.

ACKNOWLEDGEMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) for the Imperial College London Doctoral Training Partnership (DTP)

AUTHOR IDENTIFIERS

Author ORCIDs:
Spencer G: 0009-0002-5375-874X
Narayanan H: 0000-0003-3580-284X
Wirnsperger C: 0009-0000-8511-1428
Butté A: 0000-0003-3506-3792
Kontoravdi C: 0000-0003-0213-4830
Papathanasiou M. M: 0000-0002-8886-062

REFERENCES

1. Quinteros DA, Bermúdez JM, Ravetti S, Cid A, Allemandi DA, Palma SD. Therapeutic use of monoclonal antibodies: general aspects and challenges for drug delivery. Nanostructures for Drug Delivery :807-833 (2017). https://doi.org/10.1016/b978-0-323-46143-6.00025-7
2. M. Sendera, A. Sorkhei, and T. Kuśmierczyk, 'Revisiting the equivalence of Bayesian neural networks and Gaussian processes: on the importance of learning activations', in *Proceedings of the Forty-First Conference on Uncertainty in Artificial Intelligence*, in UAI '25, vol. 286. Rio de Janeiro, Brazil: JMLR.org, Jul. 2025, pp. 3675–3700.
3. Akhare D, Luo T, Wang JX. Hybridndiff-uq: uncertainty quantification for hybrid neural differentiable modeling. Theoretical and Applied Mechanics Letters 16:100609 (2026). https://doi.org/10.1016/j.taml.2025.100609
4. Krättli M, Steinebach F, Morbidelli M. Online control

of the twin-column countercurrent solvent gradient process for biochromatography. Journal of Chromatography A 1293:51-59 (2013). https://doi.org/10.1016/j.chroma.2013.03.069
5. Müller?Späth T, Krättli M, Aumann L, Ströhlein G, Morbidelli M. Increasing the activity of monoclonal antibody therapeutics by continuous chromatography (MCSGP). Biotech & Bioengineering 107:652-662 (2010). https://doi.org/10.1002/bit.22843
6. Papathanasiou MM, Avraamidou S, Oberdieck R, Mantalaris A, Steinebach F, Morbidelli M, Mueller?Spaeth T, Pistikopoulos EN. Advanced control strategies for the multicolumn countercurrent solvent gradient purification process. AIChE Journal 62:2341-2357 (2016). https://doi.org/10.1002/aic.15203
7. Foteini Michalopoulou and Maria M. Papathanasiou, 'Assessment of data-driven modeling approaches for chromatographic separation processes'. Accessed: Apr. 28, 2025. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/epdf/10.1002/aic.18600
8. D. P. Kingma, T. Salimans, and M. Welling, 'Variational Dropout and the Local Reparameterization Trick', in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2015. Accessed: Feb. 02, 2026. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2015/hash/bc7316929fe1545bf0b98d114ee3ecb8-Abstract.html
9. T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, 'Optuna: A Next-generation Hyperparameter Optimization Framework', Jul. 25, 2019, *arXiv*: arXiv:1907.10902. doi: 10.48550/arXiv.1907.10902.
10. J. Bradbury *et al.*, *JAX: composable transformations of Python+NumPy programs*. (2018). [Online]. Available: http://github.com/jax-ml/jax
11. I. Loshchilov and F. Hutter, 'Decoupled Weight Decay Regularization', arXiv.org. Accessed: Jan. 08, 2026. [Online]. Available: https://arxiv.org/abs/1711.05101v3
12. DeepMind *et al.*, *The DeepMind JAX Ecosystem*. (2020). [Online]. Available: http://github.com/google-deepmind
13. Matheson JE, Winkler RL. Scoring rules for continuous probability distributions. Management Science 22:1087-1096 (1976). https://doi.org/10.1287/mnsc.22.10.1087