DOI: 10.1111/mice.12901

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Hierarchical Bayesian modeling for knowledge transfer across engineering fleets via multitask learning

L. A. Bull[1] | **D. Di Francesco**[1] | **M. Dhada**[2] | **O. Steinert**[3] | **T. Lindgren**[4] | **A. K. Parlikad**[2] | **A. B. Duncan**[1,5] | **M. Girolami**[1,6]

[1]The Alan Turing Institute, The British Library, London, UK

[2]Institute for Manufacturing, Department of Engineering, University of Cambridge, Cambridge, UK

[3]Strategic Product Planning and Advanced Analytics, Scania CV, Scania AB (publ), Södertälje, Sweden

[4]Department of Computer and Systems Sciences, Stockholm University, Kista, Sweden

[5]Department of Mathematics, Imperial College London, London, UK

[6]Department of Engineering, University of Cambridge, Cambridge, UK

Correspondence
M. Girolami, The Alan Turing Institute, The British Library, London, NW1 2DB, UK and Department of Engineering, University of Cambridge, CB3 0FA, Cambridge, UK.
Email: mag92@cam.ac.uk

Abstract

A population-level analysis is proposed to address data sparsity when building predictive models for engineering infrastructure. Utilizing an interpretable hierarchical Bayesian approach and operational fleet data, domain expertise is naturally encoded (and appropriately shared) between different subgroups, representing (1) use-type, (2) component, or (3) operating condition. Specifically, domain expertise is exploited to constrain the model via assumptions (and prior distributions) allowing the methodology to automatically share information between similar assets, improving the survival analysis of a truck fleet (15% and 13% increases in predictive log-likelihood of hazard) and power prediction in a wind farm (up to 82% reduction in the standard deviation of maximum output prediction). In each asset management example, a set of correlated functions is learnt over the fleet, in a combined inference, to learn a population model. Parameter estimation is improved when subfleets are allowed to share correlated information at different levels in the hierarchy; the (averaged) reduction in standard deviation for interpretable parameters in the survival analysis is 70%, alongside 32% in wind farm power models. In turn, groups with incomplete data automatically borrow statistical strength from those that are data-rich. The statistical correlations enable knowledge transfer via Bayesian transfer learning, and the correlations can be inspected to inform which assets share information for which effect (i.e., parameter). Successes in both case studies demonstrate the wide applicability in practical infrastructure monitoring, since the approach is naturally adapted between interpretable fleet models of different in situ examples.

1 | INTRODUCTION

Data sparsity can cause significant issues in practical applications of reliability, performance, and safety assessment. Particularly structural monitoring (Worden & Manson, 2007), prognostics (O'Connor & Kleyner, 2012), or performance and health management (Kim et al., 2017). In these domains, comprehensive (or high variance; Paleyes et al., 2020) data are rarely available a priori; instead, measurements arrive incrementally, throughout the life cycle of the monitored system (Bull et al., 2019). For example, the data recorded from the system in

unusual environments, or following damage, might take years to collect. Labeling to annotate the measurements can also be limited or expensive, requiring input from a domain expert. Such incomplete data motivate *sharing* information between similar assets; specifically, whether systems with comprehensive data (or established models) can support those with incomplete information.

The concept of knowledge transfer, from one machine to another, has led to the development of population-based (Bull et al., 2021; Gardner, Bull, Gosliga, et al., 2021; Gosliga et al., 2021) or fleet monitoring (Zaccaria et al., 2018). Initial investigations (mostly) consider the quantification of *similarity* between systems (Gosliga et al., 2021) and tools for the *transfer* of data and/or models from *source* to *target* domains (Bull et al., 2021; Gardner, Bull, Dervilis, et al., 2021; Michau & Fink, 2019). An alternative approach is considered here, whereby a combined inference is made given the measurements from a collected group of systems (Dhada et al., 2020). Specifically, a set of correlated, hierarchical models is learnt, given the information recorded from the collected population. Two case studies are presented: survival analysis of an operational truck fleet and wind-power predictions for an operational wind farm. Population-level models are learnt using hierarchical Bayesian modeling (Gelman et al., 2013; Wand, 2009) providing robust predictions and variance reduction compared to independent models and two benchmarks. The *multitask learning* (MTL) approach (Murphy, 2012; Wand, 2009) automatically shares information between correlated domains (i.e., subgroups) such that assets with sparse information borrow statistical strength from those that are data-rich (via correlated variables).

1.1 | Why learn fleet models?

Throughout this work, the term *fleet* refers to a population of assets that constitute engineering infrastructure, for example, civil structures (bridges and roads) or vehicles (trains in a rail network). The problem setting from each case study is introduced here to motivate MTL from *in situ* fleet data.

1.1.1 | Truck fleets

The first example concerns the survival analysis of components (alternators and turbochargers) in a fleet of heavy-duty trucks maintained by Scania CV. The components are maintained in a run-to-failure strategy as failure models are unavailable and it is infeasible for drivers to sense incipient failure. Nonetheless, the associated downtime can incur high costs: relating to late goods delivery, reloading, and towing vehicles to the workshop.

For such components, survival analysis (O'Connor & Kleyner, 2012) is critical to estimate the time to failure, and therefore fundamental when designing a maintenance plan. The analysis considers failure occurrences in the population over some specified time period. The period must be sufficiently long, such that reliability functions can be evaluated based on observed failures or drop-outs (Birolini, 2013). Specifically, this work focuses on the hazard function $\lambda(t)$ which defines the instantaneous rate of failure—it is the probability $P(\cdot)$ of a component failing at time t, given that it has survived until time t (Birolini, 2013),

$$\lambda(t) = \frac{P(t \leq T < t + dt | T \geq t)}{dt} \tag{1}$$

here T denotes the *time of failure*. Empirically, this is calculated as the fraction of trucks that failed to the number of trucks that survived, in a given time interval.

Importantly, each sample from the reliability function requires at least one failure in the historical fleet data. For this reason, if failures are rare in certain subfleets, the data that represent the corresponding function will be sparse—Figure 3 visualizes this. If subfleets with more failures can inform predictions in groups where failures are rare, this greatly extends the value of the measured data (and the failure events themselves).

1.1.2 | Wind farms

The second case study considers power prediction for a group of operational wind turbines. Here, the regression tasks are *power curves*, which map from wind speed to power output for a specific turbine (Papatheou et al., 2017). The associated function can be used as an indicator of performance and is useful in monitoring procedures (Rogers et al., 2020). Data-based methods approximate this relationship from operational measurements, typically recorded using Supervisor Control and Sensory Data Acquisition (SCADA) systems (Yang et al., 2013). Various techniques have been proposed to model data that correspond to *normal* operation (Carrillo et al., 2013; Lydia et al., 2014; Thapar et al., 2011). In practice, however, only a subset of measurements represent this relationship. In particular, power *curtailments* will appear as additional functional components; these usually correspond to the output power being controlled (or otherwise limited) by the operator. Reasons for this action include adhering to the requirements of the electrical grid (Hur & Leithead, 2014; Waite & Modi, 2016), the mitigation of loading/wake effects (Bontekoning et al., 2017), and restrictions enforced by planning regulations—such data are presented in Figure 16.

Critically, different turbines experience different conditions (i.e., power curves) at varying intervals. If the

power of a particular turbine is regularly limited by the operator (as a result of its location in the farm) measurements collected from this operation become far more valuable when they can be shared between turbines. In this case, fleet modeling can be adopted to share (or *pool*) information.

1.2 | Novelty

In view of these applications, the proposed fleet modeling approach favors explainability (with some caveats) since each model is informative.

(1) Rather than black-box, a fleet model is built while encoding multilevel a priori knowledge of fleet behavior and model constraints, given domain expertise.
(2) The proposed model automatically determines the level of knowledge transfer between asset groups, learning the intertask correlations from data and combining this with a priori engineering knowledge.
(3) In turn, the approach provides formal uncertainty quantification of the fleet effects (parameters) at various asset group granularities (system-specific, operating condition, or population wide).
(4) Each subgroup predictor shares information and the associated *fleet model* provides new insights, which are greater than the sum of its parts (single-task learning [STL]).

Such fleet models are desirable, since they enable downstream analyses, to determine which groups of asses share information for which (interpretable) parameter; additionally, the model naturally integrates with experimental design or decision processes; formalizing the expected optimal action, or the value data collection activities—these concepts are demonstrated in the second case study, Section 6.4.

The approach is particularly suited to sparse, incremental data, that are found in many (practical) monitoring applications—for example, in the first (survival analysis) case study, one domain owns a single training observation.

1.3 | Layout

The paper layout is as follows. Section 2 summarizes existing work relating to population monitoring of engineering systems. Section 3 states the contributions of this work. Section 4 introduces a general methodology for knowledge transfer via hierarchical Bayesian modeling. Sections 5 and 6 present the truck fleet and wind farm case studies. Section 7 offers concluding remarks.

2 | RELATED WORK

A summary of fleet-monitoring literature is provided. The term *knowledge transfer* is used generally to refer to methods that learn from multiple related data sets. Specific definitions of *transfer learning* are contentious: This work follows Murphy (2012), which views MTL as the combined inference of a set of related tasks, while domain adaptation (DA) is a method of transforming data, such that the same task can be learnt for multiple domains. Both approaches are considered as *transfer learning*—especially when domains share interpretable, parameterized models.

2.1 | Fine-tuning and DA

When monitoring engineering populations, the majority of literature focuses on *transfer learning*. Transfer learning seeks to improve predictions in a *target domain* given the information in a (more complete) *source domain*. Many examples consider crack detection via image classification using convolutional neural networks (CNNs). For example, Dorafshan et al. (2018), Gao and Mosalam (2018), and Jang et al. (2019) detect cracks over a number of domains by *fine-tuning* the parameters of a CNN trained on a source domain to aid generalization in the target.

DA is viewed as another variant of transfer learning in engineering applications (DA) (Li et al., 2019; Wang et al., 2019; Zhang et al., 2017). These techniques define some mapping from domain data into a shared space (possibly one of the original domains) where a *single* model is used to make predictions. For example, Michau and Fink (2019) apply a neural network mapping for DA in the condition monitoring of a fleet of power plants. DA has also been investigated by (kernelized) linear projection, discussed in a structural health monitoring context by Gardner, Liu, et al. (2020) and Gardner, Fuentes, et al. (2020) considering methods for knowledge transfer between simulated source and target structures, as well as a simulated source and experimental target structure (Gardner et al., 2022). Damage detectors have also been transferred between systems via DA in a group of tailplane structures using ground-test vibration data (Bull et al., 2021). To accommodate for class imbalance and data sparsity, often associated with monitoring data, Poole et al. (2022) introduce statistic alignment methods for adaptation procedures.

2.2 | MTL

An alternative view of population-level models considers MTL. While the multitask approach also assumes that the predictors (i.e., *tasks*) are correlated over the fleet,

the parameters across domains are learnt at the same time with equal importance. A combined inference allows *domain-specific* models to share information between related tasks, improving the accuracy in domains where data are limited (Sun et al., 2021).

Examples of MTL are less prevalent when modeling engineering infrastructure. Wan and Ni (2019) successfully use a Gaussian process (GP) to learn correlations between tasks in a multioutput regression. The GP is built using a carefully specified kernel (Bonilla et al., 2007) to capture the task and intertask relationships. The experiments capture correlations between temperature/acceleration sensing systems on a single structure (the Canton Tower), rather than multiple assets in a fleet. Similarly, Li et al. (2021) apply correlated GPs to address the missing data problem over multiple sensors of a hydroelectric dam. The results demonstrate successful knowledge transfer between measurement channels. Considering aerospace engines, Seshadri et al. (2020) apply GPs for knowledge transfer between multiple axial measurement planes when interpolating temperature fields within an aircraft engine. Sharing information between planes significantly improves the spatial representation of the response.

Hierarchical Bayesian modeling offers another multitask framework. A model is built with a "hierarchy" of parameters, whereby domain-specific tasks are correlated via *shared* latent variables (explained in Section 4). The approach was introduced to structural monitoring by Huang et al. (2019) and Huang and Beck (2015), who utilize hierarchical models to learn multiple, correlated regression models for modal analysis. A shared sparseness profile is inferred over all tasks and related measurement channels, improving damage detection and data recovery by considering the correlation between damage scenarios or adjacent sensors on the same structure. Some recent, related applications include Di Francesco et al. (2021), who use hierarchical models to build corrosion models given evidence from multiple locations, and Papadimas and Dodwell (2021), where the results from a series of materials experiments (i.e., coupon samples) are combined to inform the estimation of material properties. Also, Dhada et al. (2020) implement hierarchical Gaussian mixture models to cluster simulated data that represent novelty detection for asset management; the model parameters are interpretable in terms of the data distribution, rather than the application domain.

2.3 | Wider monitoring methods

It is worth considering more general developments in the literature, and how they relate to fleet monitoring. Multitask neural networks, in particular, show promise when the size (or features) of monitoring data permits their application; for example, Zhang et al. (2020) design a deep architecture for guided wave data sets. Similarly, Tsialiamanis et al. (2022) successfully investigate neural networks for knowledge transfer by mapping measurements from multiple structures onto a common manifold, to learn a shared representation.

Recent developments in structural health monitoring, such as those relating to modal analysis (Perez-Ramirez et al., 2019), should naturally integrate with a population approach—whereby different effects of dynamic models are learnt at various granularities over the fleet. Developments in signal processing for complex civil structures (Amezquita-Sanchez et al., 2017; Li et al., 2017) might also be utilized to extract features that inform an appropriate level of information sharing between large in situ structures.

A primary motivation of this work, however, is to consider structures/domains with very sparse (or absent) data—for example, those recently in operation, or new environmental conditions. In turn, model comparisons here are limited to parametric (or *shallow* Sukhija & Krishnan, 2020) methods of knowledge transfer, centered around interpretable models—each benchmark is outlined in Section 4.4. A (general) comparison of fleet monitoring approaches is provided in Table 1, to motivate the proposed method (labeled *Hierarchical Modeling, MTL*) and its relevance in many engineering applications.

2.4 | Bayesian versus "deep" knowledge transfer

The distinction between *hierarchical* (Bayesian) and *deep* (neural network) approaches to transfer learning is important. The differences emphasize why, in many applications, the proposed (hierarchical) method is required for infrastructure monitoring. Key comparisons from Table 1 are expanded:

(1) Both address relative data sparsity (between domains) however, the level of sparsity is method dependent: Generally, deep methods are suited to complex features and big data; hierarchical methods are suited to standard measurements and interpretable models.
(2) Both improve predictions over multiple asset groups; however, the proposed hierarchical approach provides uncertainty quantification of the nested subgroups, enabling downstream (statistical) analyses—for example, experimental design or decision processes (demonstrated in Section 6.4).
(3) Encoding domain (engineering) expertise is natural for multilevel Bayesian models—for example, the

TABLE 1 Fleet modeling—A (generalized) comparison of methodologies

Method	Single task learning	Complete pooling	Fine tuning (neural nets.)	Domain adaptation	Neural Nets. (MTL)	*Hierarchical modeling (MTL)*
Knowledge transfer	none	Data-level	Pretraining on similar data sets	Data-level (transformed)	Correlated weights, shared layers	Correlated parameters, tied parameters
Interpretable	Model dependant	Model dependant	Nonparametric/ black box	Model dependent	Nonparametric/ black box	Yes
Task-specific or shared models	Task-specific	Shared only	Task-specific (once retrained)	Shared only	Both	Both (natural interpretation)
Data set size	Any (model dependent)	Any (model dependent)	Large	Model dependent	Large	Any

knowledge that all turbines in a wind farm have the same maximum power, but the rate at which they limit to a maximum will depend on turbine location.

(4) Conversely, for neural networks, encoding domain expertise is difficult since they are nonparametric; in turn, the inferences (and model constraints) at different levels of fleet granularity are less intuitive.

3 | CONTRIBUTION

The main contributions of this work are twofold: (1) MTL with hierarchical Bayesian modeling allows information to be shared between distinct (but related) systems using operational fleet data (wind turbines and trucks) rather than multiple sensors on a single structure; (2) various *mixed effects* are considered in the hierarchy, such that certain characteristics (parameters) can be learnt at the individual, group, or population level. In turn, prior engineering knowledge can be encoded at different levels in the hierarchy and parameters can be shared for various (nested) subgroups. The hierarchical models are easily formulated around interpretable parameters and the resultant structure allows insightful analyses of the predicted variables, indicating which groups of systems share information for which effect.

When MTL for engineered infrastructure, it is crucial to establish an appropriate level of knowledge transfer (data pooling) between systems or domains. If information is inappropriately shared, this can lead to *negative transfer*, whereby population models prove worse than conventional (single task) learning. Importantly, the proposed model automatically determines an appropriate level of knowledge transfer, by learning the intertask correlations from the data and combining this with engineering knowledge—encoded as prior distributions within the hierarchical structure.

The resultant approach permits formal uncertainty quantification at various levels of the predictive model, and, in turn, various granularities of fleet behavior (e.g., system-specific, condition-specific, or population-wide). Multiple levels of uncertainty quantification enable natural integration with decision processes, or experimental design procedures, considering the whole fleet. In turn, the model can be used to inform fleet interactions within a wider asset management program. To highlight this novelty, the hierarchical model is integrated with a demonstrative decision process in the second (wind farm) case study.

Similarly, while the proposed hierarchical model makes inferences from observations at the subfleet level only (i.e., task-specific outputs), predictions can be made at various levels—including larger groups and the aggregated population. Inference of the joint population model (from task-specific observations) presents the knowledge transfer mechanism. The resultant structure produces both *shared* and *task-specific* models—this is not true for any of the benchmarks, which learn one of the two options (i.e., STL, complete pooling [CP], DA—Section 4.4).

4 | HIERARCHICAL BAYESIAN MODELING FOR MTL WITH MIXED EFFECTS

Consider fleet data, recorded from a population of engineering systems, which are separated into K groups or *subfleets*. The population data can then be denoted,

$$\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^{K} = \left\{ \{x_{ik}, y_{ik}\}_{i=1}^{N_k} \right\}_{k=1}^{K} \tag{2}$$

where $\mathbf{y}_k$ is target response vector for inputs $\mathbf{x}_k$ and $\{x_{ik}, y_{ik}\}$ are the ith pair of observations in group k. There are N_k observations in each group and thus $\sum_{k=1}^{K} N_k$ observations in total. The aim is to learn a set of K predictors, one for each group, related to classification or regression tasks. Without loss of generality, this work

focuses on the regression setting, where the tasks satisfy

$$\{y_{ik} = f_k(x_{ik}) + \epsilon_{ik}\}_{k=1}^{K}$$

that is, the output is determined by evaluating one of K latent functions with additive noise ϵ_{ik}. Note, for classification, logistic regression would involve modifying the above likelihood for categorization (a Bernoulli distribution) and passing $f_k(x_{ik})$ through the logit function to ensure predictions are between zero and one (binary classification) (Murphy, 2012).

The mapping f_k is assumed to be correlated between subfleets. In consequence, the models should be improved by learning the parameters in a joint inference over the whole population. In machine learning, this is referred to as *MTL*; in statistics, such data are usually modeled with hierarchical models (Gelman & Hill, 2006; Kreft & De Leeuw, 1998).

4.1 | Hierarchical Bayesian modeling

In practice, while certain subfleets might have rich, historical data, others (particularly those recently in operation) will have limited training data. In this setting, learning separate, independent models for each group will lead to unreliable predictions. On the other hand, a single regression of all the data (CP) will result in poor generalization. Instead, hierarchical models can be used to learn separate models for each group while encouraging task parameters to be correlated (Murphy, 2012)— the established theory is summarized here.

Consider K linear regression models,

$$\left\{\mathbf{y}_k = \mathbf{\Phi}_k \boldsymbol{\alpha}_k + \boldsymbol{\epsilon}_k\right\}_{k=1}^{K} \tag{3}$$

where $\mathbf{\Phi}_k = [\mathbf{1}, \mathbf{x}_k]$ is the $N_k \times 2$ *design* matrix; $\boldsymbol{\alpha}_k$ is the 2×1 vector of *weights*; and the noise vector is $N_k \times 1$ and normally distributed[1] $\boldsymbol{\epsilon}_k \sim \mathrm{N}(0, \sigma_k^2 \mathbf{I})$. $\mathbf{1}$ is a vector of ones, $\mathbf{I}$ is the identity matrix, and $\mathrm{N}(m, s)$ is the normal distribution with mean m and (co)variance s. The likelihood of the target response vector is then

$$\mathbf{y}_k | \mathbf{x}_k \sim \mathrm{N}\left(\mathbf{\Phi}_k \boldsymbol{\alpha}_k,\ \sigma_k^2 \mathbf{I}\right) \tag{4}$$

$$\therefore \quad y_{ik} | x_{ik} \sim \mathrm{N}\left(\alpha_1^{(k)} + \alpha_2^{(k)} x_{ik},\ \sigma_k^2\right)$$

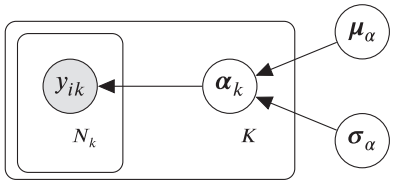

FIGURE 1 DGM of hierarchical linear regression

In a Bayesian manner, one can set a shared hierarchy of prior distributions over the weights (slope and intercept) for the groups $k \in \{1, \ldots, K\}$,

$$\{\boldsymbol{\alpha}_k\}_{k=1}^{K} \overset{\text{i.i.d}}{\sim} \mathrm{N}\left(\boldsymbol{\mu}_\alpha, \mathrm{diag}\{\boldsymbol{\sigma}_\alpha^2\}\right) \tag{5}$$

$$\boldsymbol{\mu}_\alpha \sim \mathrm{N}(\mathbf{m}_\alpha, \mathrm{diag}\{\mathbf{s}_\alpha\}) \tag{6}$$

$$\boldsymbol{\sigma}_\alpha \overset{\text{i.i.d}}{\sim} \mathrm{IG}(a, b) \tag{7}$$

In words, (5) assumes that the weights $\{\boldsymbol{\alpha}_k\}_{k=1}^{K}$ are normally distributed $\mathrm{N}(\cdot)$ with mean $\boldsymbol{\mu}_\alpha$ and covariance[2] $\mathrm{diag}\{\boldsymbol{\sigma}_\alpha^2\}$. Similarly, (6) states that the prior expectation of the weights $\boldsymbol{\alpha}_k$ is normally distributed with mean $\mathbf{m}_\alpha$ and covariance $\mathrm{diag}\{\mathbf{s}_\alpha\}$; (7) states that the prior deviation of the slope and intercept is inverse-Gamma distributed $\mathrm{IG}(\cdot)$ with shape and scale parameters a and b respectively.

Selecting appropriate prior distributions, and their associated hyperparameters $\{\mathbf{m}_\alpha, \mathbf{s}_\alpha, a, b\}$, is essential to the success of hierarchical models. In this work, prior elicitation is justified by encoding engineering knowledge in each case study as weakly informative priors (Gelman et al., 2013). The directed graphical model (DGM) in Figure 1 visualizes the general hierarchical regression. The nodes show observed/latent variables as shaded/nonshaded, respectively; arrows show conditional dependencies, and plates show multiple instances of subscripted nodes.

The K weight vectors $\boldsymbol{\alpha}_k$ are correlated via the common latent variables $\{\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha^2\}$, that is, parent nodes in Figure 1. Note that Equations (5) to (7) encode *prior* belief of the independence between latent variables. In this work, this does not restrict the covariance structure of the posterior distribution for $\{\boldsymbol{\alpha}_k\}_{k=1}^{K}$ since it is approximated using Markov Chain Monte Carlo (MCMC, summarized in Section 4.3).

Via correlations in the posterior distribution, sparse domains borrow statistical strength from those that are data-rich. Crucially, to *share* information between tasks, the parent nodes $\{\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha^2\}$ must be inferred from the population data. In this way, the subfleet parameters $\boldsymbol{\alpha}_k$ are (indirectly) influenced by the wider population. Consider that, if $\{\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha^2\}$ were fixed constants, rather than variables

[1] In this first introductory example, the additive noise variance σ_k^2 is observed—in the next example, it is unobserved.

[2] The operator $\mathrm{diag}\{\mathbf{a}\}$ forms a square diagonal matrix with the elements from $\mathbf{a}$ on the main diagonal and zeros elsewhere.

inferred from data, each model would be conditionally independent, preventing the *flow* of information between domains (Murphy, 2012).

4.2 | Mixed-effects modeling

The hierarchical structure allows *effects* (i.e., interpretable latent variables) to be learnt at different levels, as well as "prior" information. Specifically, the parameters of the model itself (3) can be learnt at the system, subfleet, or population level. The inference of parameters at various levels of hierarchy, while encoding engineering/domain knowledge at each level, constitutes significant novelty here.

Returning to the regression example (3), consider that the variance σ_k^2 of the noise ϵ_k is in fact unknown. While one could learn K domain-specific noise variance terms σ_k^2, it is typically assumed that the noise is equivalent across tasks. Sharing the parameter and inferring it from the population can significantly reduce the uncertainty in its prediction. Of course, this assumption should be justified given an understanding of the problem at hand; for example, the same sensing system collects all the population data. In terms of notation, (3) remains the same, however, the domain-specific noise vector $\boldsymbol{\epsilon}_k$ is now distributed $\boldsymbol{\epsilon}_k \sim \mathrm{N}(0, \sigma^2\mathbf{I})$. The removal of subscript-k from the noise variance implies that the size of σ^2 remains the same while the number of the subfleets K increases (unlike $\boldsymbol{\alpha}_k$). Intuitively, σ^2 is now a *tied* parameter (Murphy, 2012).

Similarly, it makes sense to also infer *effects* at the population level, to further reduce model uncertainty.[3] Throughout this work, it is assumed that *shared* effects also enter the model linearly, for the target response vector $\mathbf{y}_k$ and inputs $\mathbf{x}_k$,

$$\left\{ \mathbf{y}_k = \underbrace{\boldsymbol{\Phi}_k \boldsymbol{\alpha}_k}_{\text{random}} + \underbrace{\boldsymbol{\Psi}_k \boldsymbol{\beta}}_{\text{fixed}} + \boldsymbol{\epsilon}_k \right\}_{k=1}^{K} \tag{8}$$

where $\boldsymbol{\Psi}_k$ is some design matrix of inputs, and $\boldsymbol{\beta}$ is the corresponding vector of weights. Again, there is no subscript-k for $\boldsymbol{\beta}$ (like σ^2) as it is tied between subfleets. Following Kreft and De Leeuw (1998), the $\boldsymbol{\beta}$ coefficients are considered *fixed effects*, as they are learnt at the population level and shared, while $\boldsymbol{\alpha}_k$ are *random effects*, as they vary between *individuals*. Intuitively, a model with both fixed and random effects can be considered a *mixed* (effects) model (Gelman et al., 2013; West et al., 2006). Figure 2 shows the modified DGM of the hierarchical regression. The key differences are nodes outside of the K plate—these are the tied parameters, learnt at the population level.

[3] For example, the intercept would be a shared parameter, with zero mean, in a related linear regression of Hooke's law for several materials tests.

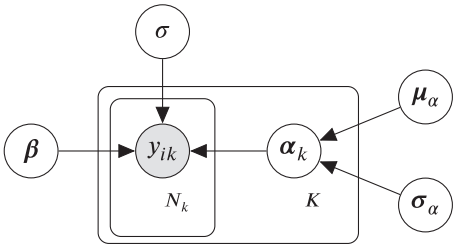

FIGURE 2 DGM of hierarchical linear regression with mixed effects

As Gelman et al. (2013) point out, the terms *random* and *fixed* originate from a frequentist perspective and are somewhat confusing in a Bayesian context where all parameters are random, or (equivalently) fixed with unknown values. The terminology is used, however, as it is intuitive considering engineering applications and consistent with established literature in modeling panel or longitudinal data (Gelman & Hill, 2006). One should also consider that interpreting mixed-effects models remains challenging, even when models are parameterized. If the effects are not (linearly) independent, the fixed and random coefficients can influence each other, making it difficult to reliably recover their relationships. In turn, the modeling assumptions must be carefully considered when emphasizing interpretability.

4.3 | Inference

In view of graphical models, the observed variables are referred to as *evidence* nodes. For example, the hierarchical regression in Figure 1 would have the following set of evidence nodes:

$$\mathcal{E} = \{[\mathbf{y}_k]\} \tag{9}$$

where $[\mathbf{y}_k]$ is shorthand to denote complete set $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K\}$. On the other hand, the latent variables are *hidden* nodes,

$$\mathcal{H} = \{[\boldsymbol{\alpha}_k], \boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha\} \tag{10}$$

Bayesian inference relies on finding the posterior distribution of $\mathcal{H}$ given $\mathcal{E}$, that is, the distribution of the unknown parameters given the data,

$$\begin{aligned} p(\mathcal{H}|\mathcal{E}) &= \frac{p(\mathcal{H}, \mathcal{E})}{p(\mathcal{E})} \\ &= \frac{p([\mathbf{y}_k, \boldsymbol{\alpha}_k], \boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha)}{p(\mathbf{y}_k)} \\ &= \frac{p([\mathbf{y}_k]|[\boldsymbol{\alpha}_k])p([\boldsymbol{\alpha}_k]|\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha)p(\boldsymbol{\mu}_\alpha)p(\boldsymbol{\sigma}_\alpha)}{\int\int\int p([\mathbf{y}_k, \boldsymbol{\alpha}_k], \boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha)\, d\boldsymbol{\alpha}_k d\boldsymbol{\mu}_\alpha d\boldsymbol{\sigma}_\alpha} \end{aligned} \tag{11}$$

DGM representations are useful since inference can be aided by graph-theoretic results. The systematic application of graph-theoretic algorithms has led to a number of probabilistic programming languages—here, models are implemented in `Stan` (Carpenter et al., 2017). The parameters are inferred using MCMC, via the no U-turn implementation of Hamiltonian Monte Carlo (Hoffman et al., 2014). Throughout, the burn-in period is 1000 iterations and 2000 iterations are used for inference. Code based on the first case study is publicly available on GitHub.[4]

4.4 | Engineering applications

In each case study, hierarchical models are formulated for knowledge transfer between asset models. The first concerns survival analysis of truck fleets (hazard curves) and the second concerns power prediction for turbines (power curves). Engineering expertise is encoded in a number of ways: to (1) inform prior elicitation, (2) determine which effects are random or fixed, and (3) formulate interpretable parameters. In turn, population modeling offers insights as to which subfleets share information for which (interpretable) effect.

Importantly, by considering the collected population, the training data can, in effect, be extended. In turn, parameter estimation is improved while increasing the reliability of predictions. There are, of course, important considerations when building such models—prior elicitation, mixed-effects formulation, negative transfer—these concepts are discussed throughout.

Throughout, the predictive performance of the multitask methodology (MTL) is compared to three fleet monitoring benchmarks:

(1) STL: the predictive model learnt from each domain independently.
(2) CP: the predictive model learnt from the collected fleet data, assuming all data are generated by a single task.
(3) (CRL) Correlation alignment for DA: sequentially treating each task $\hat{k}$ as the target domain, and embedding the remaining (source) domains onto the joint distribution $p(\mathbf{y}_{\hat{k}}, \mathbf{x}_{\hat{k}})$ using CORAL (Sun et al., 2017). All measurements are treated as one task, and a single model is learnt, to predict the *target* test data.

For sensible comparisons, the predictive model is consistent across all benchmarks—what differs is the effective *presentation* of data during inference. Note that parameter interpretation becomes problematic in domain adaptation (CRL) since the (source) joint distributions $\{p(\mathbf{y}_k, \mathbf{x}_k)\}_{k=1}^{K}$ have been transformed onto the target (Poole et al., 2022). Once transformed, making predictions for new source observations is nontrivial. These caveats highlight a benefit of the proposed methodology; however, comparisons to CRL are included to emphasize that adaptation alone is insufficient to treat all fleet monitoring problems, especially with parameterized models and sparse data.

By nature of the practical applications (and data sensitivity) in each case study, validation to a ground truth for *parameters* is not feasible; for this reason, models are compared to the available (response) ground truth and quantified by the predictive log-likelihood (e.g., (22)).

[4] Rather than the operational data presented here, the code uses simulated data (in view of data sensitivity).

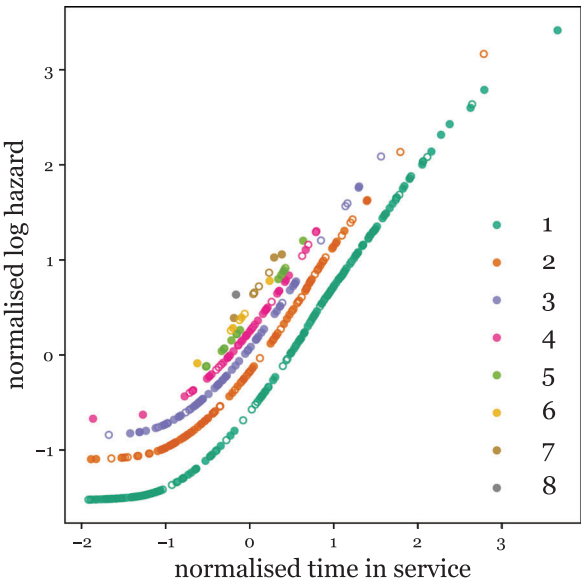

FIGURE 3 Log hazard function data for alternators in the truck fleet. Training and testing markers are • and ∘, respectively. Colors correspond to subfleet labels, associated with the task index $k \in \{1, 2, \ldots, 8\}$

5 | TRUCK-FLEET SURVIVAL ANALYSIS

The hazard data for truck fleet alternators are shown in Figure 3. Herein, this work considers the log-hazard, since it is easier to visualize. There are 437 observations in total, split into a 75% training set and 25% test set. The data are z-score normalized in view of data sensitivity and certain (specific) details are omitted. The observations represent the complete monitoring data set, since no observations we lost via normalization, truncation, or censorship. It is clarified that normalization affects the *direct* interpretation of the parameters. In practice, however, one can recover interpretable values by transforming back into the

original space. Here, for the purpose of discussion, the relative parameter values and their relationships remain interpretable. To generate the hazard data, the total time in service for all assets was divided into intervals of 1 day; for each day, the ratio of the number of components that failed to the number that survived (so far) is calculated. The choice of interval length is dependent on the application—here 1 day is sufficient compared to the maintenance horizon.

The subfleets were manually labeled in collaboration with the engineers at Scania. Colors correspond to different subpopulations, where the total number of groups (and, therefore, hazard functions) is $K = 8$. Note that certain domains are more sparse than others, with the most extreme case being $k = 8$, owning a single observation. The population model will look to utilize data-rich domains with more information ($k \in \{1, 2, 3\}$) to support the sparse domains ($k \in \{5, 6, 7, 8\}$). The number of task-wise observations is as follows:

N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	$\sum_{k=1}^{K} N_k$
180	108	70	49	15	7	7	1	437

5.1 | Task regression formulation

When analyzing survival data, it is convenient to assume the survival time T is parametrically distributed since the parameters are interpretable and formulate a specific hazard function. A straightforward example is presented when T is exponentially distributed, leading to a constant hazard (Rodrıguez, 2010).

Rather than constant, Figure 3 shows the log-hazard is near-linear for a large proportion of the input domain, with a notable nonlinear effect at low t values (early hours in service). Therefore, it is assumed the best (parametric) approximation of the marginal $p(T = t)$ is the Gompertz distribution (G) for each subfleet (Rodrıguez, 2010),

$$\begin{aligned} p(T = t) &= \mathrm{G}(t\ ;\ \gamma, \phi) \\ &= (\gamma e^{\phi t}) \exp\left\{ -\frac{\gamma}{\phi}(e^{\phi t} - 1) \right\} \end{aligned} \tag{12}$$

This is convenient, since (12) is formulated such that log-hazard is linear in time t,

$$\begin{aligned} \log \lambda_{\mathrm{G}}(t) &= \log \gamma + \phi t \\ &= \alpha_1 + \alpha_2 t \end{aligned} \tag{13}$$

Since only hazard data were available, tasks are fit directly to (13) rather than the distribution over the time at failure (12). The correct likelihood, however, should consider the distribution (12) as the tasks directly—this avoids assumptions of a Gaussian likelihood for the log-hazard. Instead, the (log) hazard uncertainty would be naturally represented by the variance of γ and ϕ. Unfortunately, this was not possible here in view of data availability. For a better interpretation of the parameters in practice, and agreement with Kolmogorov's axioms, the likelihood of the population model should represent the time-at-failure T directly.

Considering the data in Figure 3, a weighted sum of H B-spline bases functions $b_h(t)$ is included to model the (nonparametric) discrepancy between the linear Gompertz hazard and the empirical data,

$$\begin{aligned} \log \lambda(t) &= \alpha_1 + \alpha_2 t + \sum_{h=1}^{H} \beta_h b_h(t) \\ &= \log \lambda_{\mathrm{G}}(t) + \sum_{h=1}^{H} \beta_h b_h(t) \end{aligned} \tag{14}$$

Cubic B-splines (Appendix A) are selected as they are smooth with compact support, resulting in a sparse design matrix for the β_h terms.[5] This property is suitable since the nonlinear response acts in specific (compact) regions of the input. In effect, (14) defines a semiparametric (or a partially linear) regression (Wand, 2009) with kernel smoothing to approximate the hazard functions for each subfleet.

5.2 | Mixed-effects formulation

From Figure 3, one observes the underlying linear trend $\{\alpha_1 + \alpha_2 t\}$ is varying between subfleets while the nonlinear effect $\sum_{h=1}^{H} \beta_h b_h(t)$ appears consistent over the population. In other words, while the data are poorly described by a (linear) Gompertz hazard function, the (nonparametric) discrepancy remains consistent.

Therefore, the associated spline weights $\boldsymbol{\beta} = \{\beta_h\}_{h=1}^{H}$ are assumed to be fixed effects and learnt at the population level. On the other hand, task-specific linear weights are inferred, which are correlated via common latent variables (random effects) $\boldsymbol{\alpha}_k = \{\alpha_1^{(k)}, \alpha_2^{(k)}\}$.

[5] An appropriate number of splines H will be determined through cross-validation. H is treated deterministically to simplify implementation and improve stability since the uncertainty of H is less informative compared to more interpretable parameters

The mixed effect model can now be expressed in the general notation from (8),

$$\left\{ \mathbf{y}_k = \underbrace{\mathbf{\Phi}_k \boldsymbol{\alpha}_k}_{\text{random}} + \underbrace{\mathbf{\Psi}_k \boldsymbol{\beta}}_{\text{fixed}} + \boldsymbol{\epsilon}_k \right\}_{k=1}^{K}$$

Specifically, for each subfleet k: $\mathbf{y}_k$ is the output of the log-hazard (14) with additive noise $\boldsymbol{\epsilon}_k$; $\mathbf{x}_k$ are the inputs corresponding to time t; $\boldsymbol{\alpha}_k$ is the varying linear weight vector with design matrix $\mathbf{\Phi}_k = [\mathbf{1}, \mathbf{x}_k]$; and $\boldsymbol{\beta}$ is the tied/fixed weight vector, with a design matrix of splines,

$$\mathbf{\Psi}_k = \left[b_1(\mathbf{x}_k), b_2(\mathbf{x}_k), \dots, b_H(\mathbf{x}_k) \right] \tag{15}$$

The resultant graphical model corresponds to Figure 2 and the likelihood of the response is,

$$y_{ik} | x_{ik}, \theta_k \sim \mathrm{N}\left(\alpha_1^{(k)} + \alpha_2^{(k)} x_{ik} + \sum_{h=1}^{H} \beta_h b_h(x_{ik}),\ \sigma^2 \right) \tag{16}$$

where $\theta_k = \{\boldsymbol{\alpha}_k, \boldsymbol{\beta}, \boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha, \sigma\}$ is the set of parameters indexed to task k.

5.3 | Weakly informative priors

Primarily considering $\boldsymbol{\alpha}_k$, it is possible to encode prior knowledge of the expected functions, since the linear component corresponds to a Gompertz survival model (13). It is acknowledged that, in this case, the specific hyperparameter values are less meaningful as the data are normalized; however, their interpretation remains relevant.

Specifically, $\boldsymbol{\alpha}_k$ is distributed according to Equations (5) to (7), with hyperparameters,

$$\mathbf{m}_\alpha = [0, 1.5]^\top, \qquad \mathbf{s}_\alpha = [2, 0.5]^\top \tag{17}$$

$$a = 1, \qquad b = 1 \tag{18}$$

The first element of $\mathbf{m}_\alpha$ corresponds to the intercept and postulates the baseline log-hazard.[6] (This is 0 since the data are centered). The second element of $\mathbf{m}_\alpha$ is the expected slope of the log-hazard. (Set to 1.5 as one expects hazard to increase exponentially under the Gompertz model with a gradient > 1 when normalized). The $\mathbf{s}_\alpha$ values indicate a weakly informative prior under the ranges imposed by z-score normalization. Similarly, the a, b values encourage correlation between subfleet models, such that the prior mode of the standard deviation of the generating distribution of $\boldsymbol{\alpha}_k$ is $b/(a+1) = 1/2$ (this intentionally overestimates the deviation $\boldsymbol{\sigma}_\alpha$ between subfleets, such that the population model weakly constrains $\boldsymbol{\alpha}_k$).

The shared prior over the variance of the additive noise $\boldsymbol{\epsilon}_k$ is set to,

$$\sigma \sim \mathrm{IG}(3, 0.8) \tag{19}$$

whose mode is at 0.2, indicating that the standard deviation of the noise is expected to be significantly less (around five times) than that of the output, that is, a high signal-to-noise ratio.

Following a standard approach (Gelman et al., 2013) the basis function model can be centered around the linear component ($\log \lambda_{\mathrm{G}}(t)$) via specification of the $\boldsymbol{\beta}$ prior. Specifically, one can postulate a *shrinkage* prior with a high density at zero, to (effectively) exclude basis functions by encouraging their expected posterior weights to be near-zero—while also having heavy tails to avoid over-shrinkage. A standard hierarchical prior is used (Tipping, 2001), which exhibits these desired properties,

$$\beta_h \sim \mathrm{N}(0, \sigma_h^2), \qquad \sigma_h^2 \sim \mathrm{IG}(\upsilon, \upsilon) \tag{20}$$

where υ is some small nonzero value—in this case $\upsilon = 10^{-3}$.

To summarize, without any data, the prior postulates that the underlying log-hazard is expected to be linear, corresponding to a Gompertz survival model (13). The discrepancy between this simple (parameterized) behavior and the data will be modeled by nonparametric splines, resulting in a semiparametric regression (14) for each task. Figure 4 visualizes the implications of the model and prior, which shows the posterior predictive distribution inferred from the most data-rich domain only ($k = 1$, STL). This experiment is used to validate an appropriate number of splines for the population model, which is found to be $H = 5$ through 20-fold cross-validation, presented in Appendix B. It is intuitive to note, the same independence can, in effect, be achieved for parameters with hierarchical priors (i.e., $\boldsymbol{\alpha}_k$) by letting the variance of their generating distribution become very large (Gardner, Fuentes, et al., 2020) (i.e., $\boldsymbol{\sigma}_\alpha \to \infty$).

Following Section 4.3, and collecting all task parameters $\mathbf{\Theta} = \{[\boldsymbol{\alpha}_k], \boldsymbol{\beta}, \boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha, \sigma\}$, the posterior distribution can be written,

$$p(\mathbf{\Theta} | [\mathbf{y}_k]) = \frac{p([\mathbf{y}_k], \mathbf{\Theta})}{p([\mathbf{y}_k])} = \frac{p([\mathbf{y}_k] \mid \mathbf{\Theta}) p(\mathbf{\Theta})}{\int p([\mathbf{y}_k], \mathbf{\Theta}) d\mathbf{\Theta}} \tag{21}$$

[6] Or the exponentiated initial rate-of-failure.

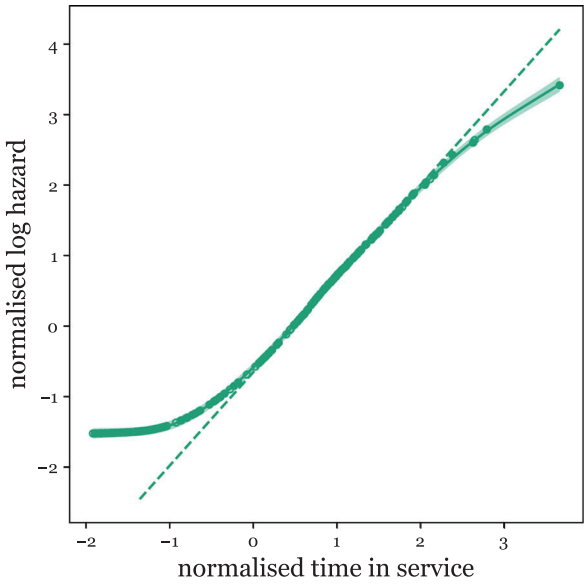

FIGURE 4 Basis function model for the data-rich domain ($k = 1$). The parametric Gompertz component (13) is the dashed line and the posterior mean of the semiparametric model (14), including splines, is the solid line

where $p([\mathbf{y}_k] \mid \boldsymbol{\Theta})$ is indexed by (16) and the joint prior $p(\boldsymbol{\Theta})$ is defined by (17) to (20). MCMC is used for inference since (21) is intractable.

Having conditioned on the training data $[\mathbf{y}_k]$, predictions can be made for the unobserved response $\mathbf{y}_k^*$ at $\mathbf{x}_k^*$ using the posterior predictive distribution,

$$p(\mathbf{y}_k^* \mid \mathbf{x}_k^*, [\mathbf{y}_k]) = \int p(\mathbf{y}_k^* \mid \mathbf{x}_k^*, \boldsymbol{\Theta}) p(\boldsymbol{\Theta} \mid [\mathbf{y}_k]) d\boldsymbol{\Theta} \quad (22)$$

(Conditioning on $\mathbf{x}_k^*$ is included here to emphasize prediction.)

5.4 | Results

To motivate sharing information within the fleet, the regression tasks for each subfleet are initially learnt independently. This corresponds to learning separate (task-specific) parameters, which are independent, preventing the flow of information via correlated variables or tied parameters. The separated models can be visualized by removing the K plate from the DGM in Figure 2, while including k-subscripts for σ^2 and $\boldsymbol{\beta}$. Figure 5 presents these updates.

Figure 6 shows the resulting domain-wise regression (i.e., STL). The posterior-predictive distributions $p(\mathbf{y}_k^* | \mathbf{x}_k^*, \mathbf{x}_k, \mathbf{y}_k)$ make sense under the model/prior formulation, however, independent models fail to consider that valuable information might be shared between the task

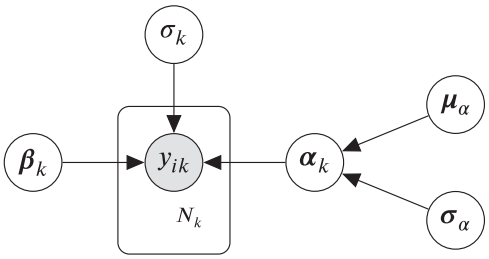

FIGURE 5 DGM for independent linear models

TABLE 2 Out-of-sample (average) predictive log-likelihood for 25% test data: $\log p(\mathbf{y}_k^* | \mathbf{x}_k^*)$

model	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$\mathcal{L}$
CP	−24.13	0.84	−4.49	−4.02	−2.41	−2.77	−6.50	−43.49
CRL	79.29	49.50	20.07	11.38	4.24	2.94	**5.47**	172.88
STL	150.24	94.24	57.66	47.04	8.51	−3.17	0.95	355.47
MTL	**166.18**	**98.23**	**64.78**	**58.09**	**25.7**	**10.17**	−13.58	**409.57**

relationships. In turn, the posterior predictive distribution presents large uncertainty, especially in sparse domains.

Hierarchical modeling is now utilized to learn the parameters in a combined inference from the population data. The mean and standard deviation of samples drawn from the MTL posterior predictive distribution are shown in Figure 7. Visually, the predictive distributions $p(\mathbf{y}_k^* | \mathbf{x}_k^*, \{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K)$ better represent belief of the underlying task functions by leveraging information between domains. In particular, information from data-rich domains ($k \in \{1, 2, 3, 4\}$) informs the (fixed) nonlinear effect.

The predictive (log) likelihood for out-of-sample test data (25%) is evaluated for a large number of trials (100) via bootstrap sampling (Murphy, 2012). The combined population log-likelihood $\mathcal{L}$ increases significantly, from 355 to 410, highlighting improvements following inference at the fleet level. Table 2 presents the relative changes for each task, where CRL is CORAL for joint adaptation.[7] Compared to STL there is a relative improvement in all domains (other than $k = 7$) especially those domains with sparse training data. In particular, leveraging information enables more reliable extrapolation to late hours in service where the test data are likely to be sparse. It is believed that the likelihood decrease occurs in domain $k = 7$, since the subfleet labeling may be unreliable—the hazard data could in fact represent more than one group when observing Figure 3. Improvements to the labeling procedure are discussed as future work, Section 7.

[7] Domain $k = 8$ is excluded since there is only one observation in the historical fleet data.

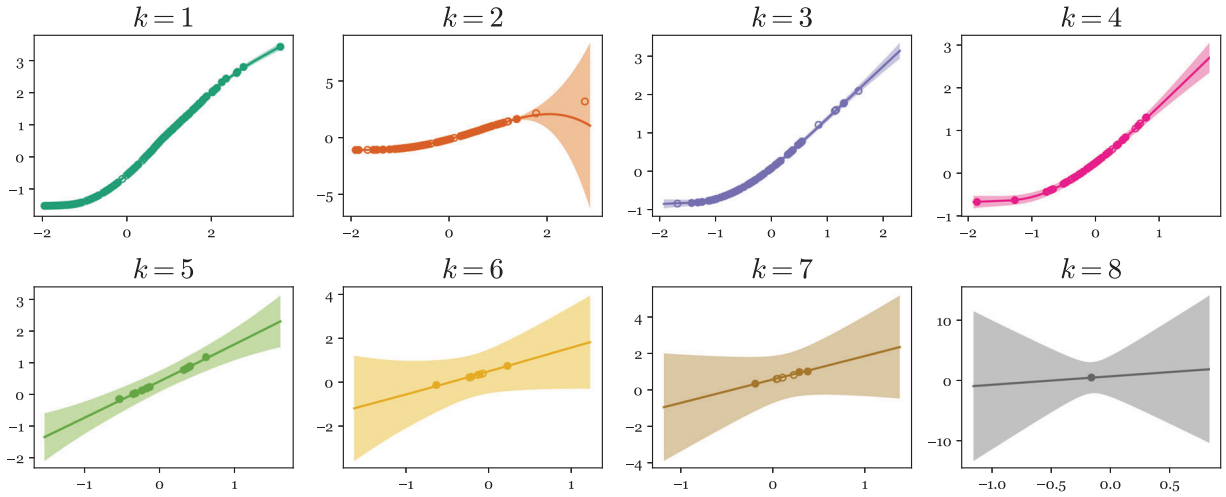

FIGURE 6 Posterior predictive distribution $p(\mathbf{y}_k^*|\mathbf{x}_k^*, \mathbf{x}_k, \mathbf{y}_k)$: the mean and three-sigma deviation for K independent regression models

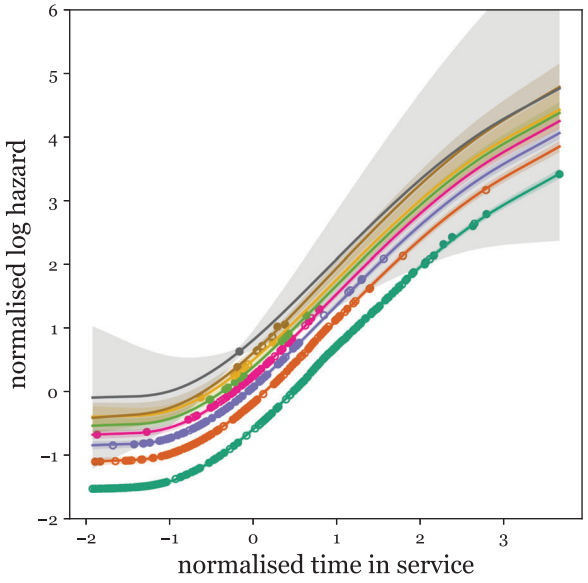

FIGURE 7 Posterior predictive distribution $p(\mathbf{y}_k^*|\mathbf{x}_k^*, \{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K)$: the mean and three-sigma deviation for multitask learning with mixed effects

CP and CRL benchmarks behave as expected. CP presents the lowest overall log-likelihood $\mathcal{L}$, which makes sense considering the disparity between tasks. CRL successfully improves from CP by transforming the source data (all remaining domains) into the target k, especially when $k = 7$. However, the total likelihood remains lower than STL, which indicates a high risk of negative transfer—in fact, CRL improves predictions in only $k = \{6, 7\}$.

Reductions in the posterior variance of the parameters via MTL are also considered, compared to STL. Figures 8 and 9 show the posterior distribution of the slope and intercepts, respectively: These parameters correspond to the random (linear) effect of the Gompertz model $\alpha_1 + \alpha_2 t$ (13). Variance reductions are most significant in sparse domains (bottom row) and less significant in the data-rich domains (top row). This follows intuition since the population model allows sparse domains to *borrow* information via the shared parent nodes $\{\boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha^2\}$ while the data-rich domains are largely unaffected. Quantitatively, the average reduction in standard deviation for the (interpretable) linear weights is 90% and 73% for the slopes and intercepts, respectively.

Figure 10 shows the posterior distribution of the fixed weights $\boldsymbol{\beta}$. Under the prior specification, these weights adaptively deviate from zero to model the discrepancy from the linear effect in sparse/compact regions of the input (via nonparametric splines). Building on intuition, by tying these parameters, the expected values shift toward the expectation of the data-rich, independent models ($k \in \{1, 2, 3, 4\}$). In other words, in the population-level inference, the fixed effect is learnt from the domains, which have data to describe it.

Likewise, Figure 11 shows improvements in the estimate of $\sigma_{(k)}$ when tying the noise effect. The posterior variance is reduced, while the expected values indicate a lower noise variance. This should be expected since by pooling the data to learn $\sigma_{(k)}$ the training set is effectively extended; in turn, the posterior moves further away from the weakly informative prior (19).

5.5 | Modeling additional failures and the risk of negative transfer

The assumptions that select the tied parameters are critical—this caveat is widely acknowledged. If any assumptions prove inappropriate or nongeneral, the multitask learner can risk negative transfer, whereby

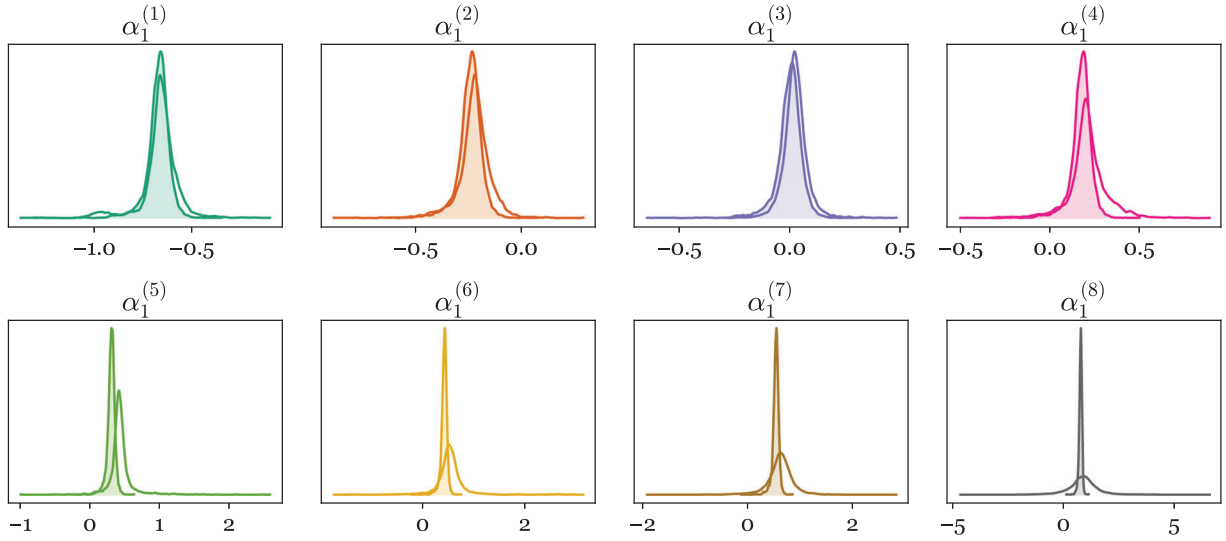

FIGURE 8 Variance reduction in the posterior distribution of the intercept parameters $\alpha_1^{(k)}$ for alternator components. Independent models (hollow) compared to population-level modeling (shaded)

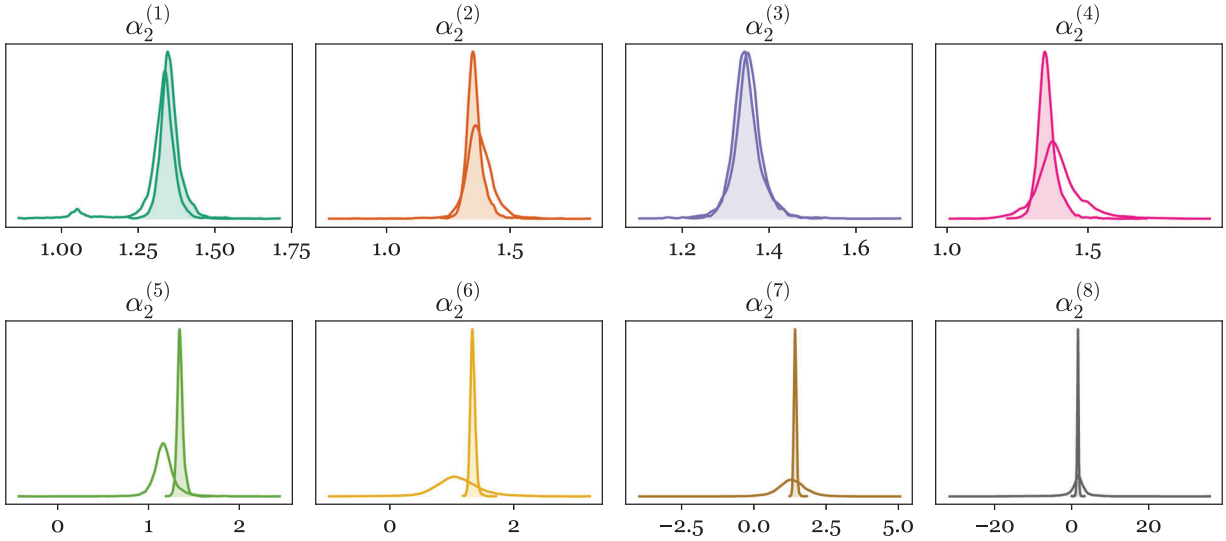

FIGURE 9 Variance reduction in the posterior distribution of the slope parameters $\alpha_2^{(k)}$ for alternator components. Independent models (hollow) compared to population-level modeling (shaded)

predictions are worse than conventional (i.e., single-task) learning—that is, in this case, independent models. In a probabilistic setting, negative transfer manifests as inappropriate intertask correlations; to control these dependencies one could utilize shrinkage (Gelman & Hill, 2006) or automatic relevance determination (Tipping, 2001) (between tasks) to protect against such issues; these ideas are suggested for future work.

To highlight concerns of negative transfer, the empirical hazard data are considered from another component in the same fleet of vehicles, turbochargers. The survival data are presented Figure 12, which are calculated following the same procedure as the alternators. Critically, manually labeling the alternator data is problematic, since it becomes infeasible to categorize observations as the generating functions become more compact, or toward the end of the operational life. The associated *unlabeled* data are highlighted with small ∘ markers in Figure 12.

There are various options when considering these data. One could treat the observations as a single (pooled) subfleet or task, with a large expected variance; alternatively, the labels themselves could be treated as an additional latent variable, such that categorization into task groups is unsupervised. Here, the unlabeled data are removed during preprocessing, since modeling them is out of the scope of this work; alternative solutions are proposed in the concluding remarks, Section 7. The resulting turbocharger data set has 287 (normalized) observations over six tasks,

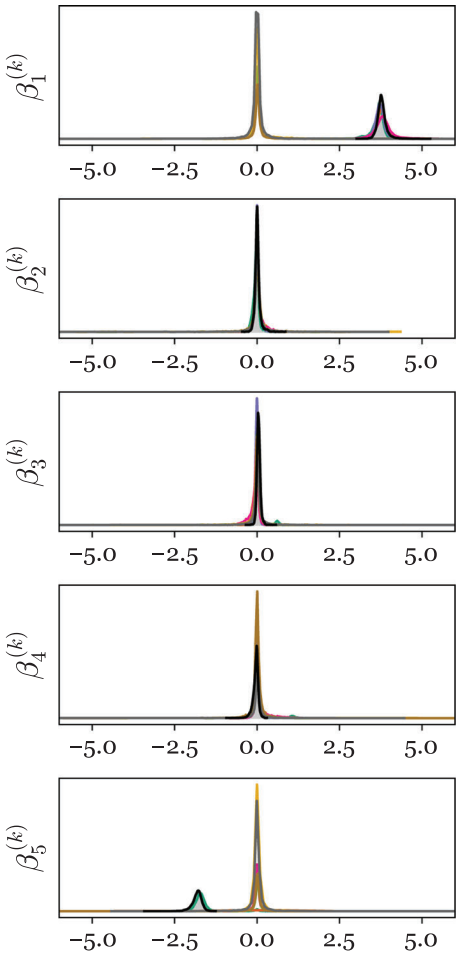

FIGURE 10 Posterior distribution of the weight parameters $\boldsymbol{\beta}_{(k)} = \{\beta_h^{(k)}\}_{h=1}^{H}$. Comparison between the tied population-level parameters (grey shaded) and independent models (hollow) for each domain $k \in \{1, \dots, 8\}$. Zoomed sections for $h = 1$ and 5 are provided in Appendix D

such that $k \in \{1, 2, \dots, 6\}$, and the number of observations in each domain is as follows:

N_1	N_2	N_3	N_4	N_5	N_6	$\sum_{k=1}^{K} N_k$
112	60	32	28	25	30	287

As before, the data are split into 75% training and 25% test sets.

From Figure 12, one observes that the turbocharger hazard data are similar to Figure 3 (alternators). Since the components operate within the same fleet of vehicles, it is assumed that information can be shared between the associated predictors by extending the task-set in the hierarchical model. A naïve approach assumes the same formulation of mixed effects, and simply extends the total number of tasks such that $K = 14$ (i.e., $8 + 6$) then infers

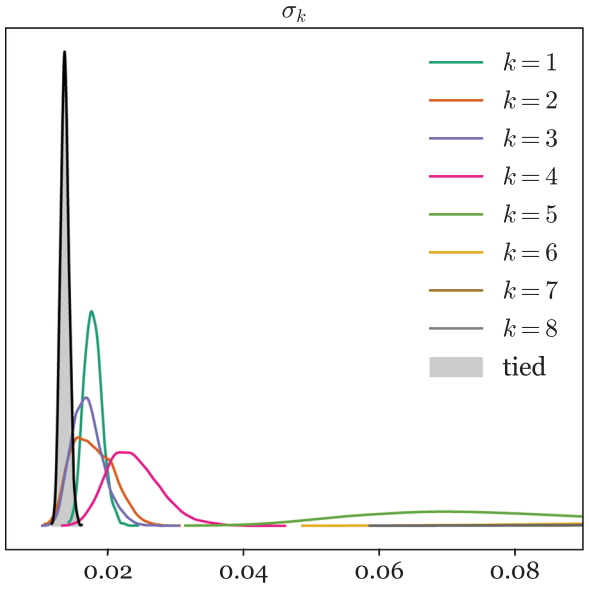

FIGURE 11 Posterior distribution of the noise parameter $\sigma_{(k)}$. Comparison between the tied population-level parameters (gray shaded) and independent models (hollow) for each domain $k \in \{1, \dots, 8\}$

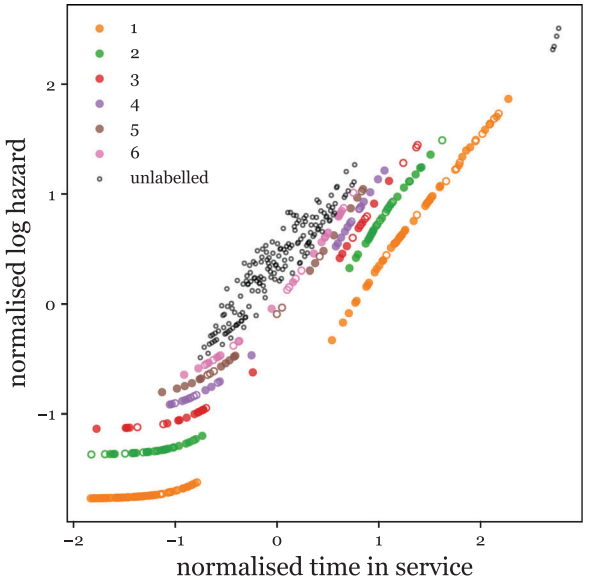

FIGURE 12 Log hazard data for turbochargers in the truck fleet. Training and testing markers are • and ∘, respectively. Colors correspond to subfleet labels

the parameters from both alternator and turbocharger hazard data. Appendix E presents the posterior predictive distribution of such a model. While the model interpolates well, the extrapolation behavior[8] is problematic for later

[8] At the population level, this is not extrapolation, since the response at late hours in service is learnt from the alternator domain.

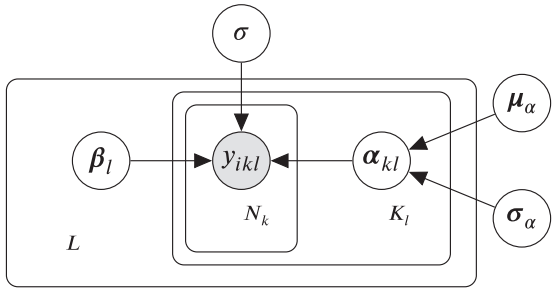

FIGURE 13 DGM of hierarchical linear regression with mixed effects. Introducing a higher level group, such that the total number of tasks is $L \times K_l$

hours in service. This is because the model assumes that the discrepancy (from the Gompertz model) is equivalent for both components, as the nonparametric weights $\boldsymbol{\beta}$ remain tied over all tasks. The unlabeled data are evidence that this assumption is inappropriate, as the model would generalize poorly to these data, plotted in Appendix E. The resultant model would have a high risk of negative transfer.

Instead, the mixed effect model is reformulated, whereby a separate, nonparametric discrepancy $\{\boldsymbol{\beta}_l\}_{l=1}^{L}$ is learnt for the alternator ($l = 1$) and turbocharger ($l = 2$) tasks—introducing two higher level subgroups, such that $L = 2$. As before, the parameters of the linear component remain correlated via the shared parent nodes, allowing knowledge transfer between all 14 tasks (both alternators and turbochargers). In turn, the model and prior now postulate a varying underlying linear trend for all tasks (the Gompertz model); however, the discrepancy from this behavior is component-specific (a separate $\boldsymbol{\beta}_l$ for each component). The modifications can be visualized by updating the DGM from Figure 2 to include higher level subgroups $l \in \{1, 2\}$, presented in Figure 13, where $l = 1$ alternators or $l = 2$ turbochargers.

A key difference is the new L-plate and the associated subscripts: K_l is the number of subfleets for each component, such that $K_1 = 8$ (alternators) or $K_2 = 6$ (turbochargers); while $\boldsymbol{\beta}_l$ indicates a separate (independent) weight vector for each component. The collected tasks become

$$\left\{ \left\{ \mathbf{y}_{kl} = \underbrace{\boldsymbol{\Phi}_{kl}\boldsymbol{\alpha}_k}_{\text{random}} + \underbrace{\boldsymbol{\Psi}_{kl}\boldsymbol{\beta}_l}_{\text{fixed}} + \boldsymbol{\epsilon}_{kl} \right\}_{k=1}^{K_l} \right\}_{l=1}^{L} \tag{23}$$

In turn, the likelihood of the response is modified,

$$y_{ikl}|x_{ikl}, \boldsymbol{\theta}_{kl} \sim \mathrm{N}\left(\alpha_1^{(kl)} + \alpha_2^{(kl)} x_{ikl} + \sum_{h=1}^{H} \beta_h^{(l)} b_h(x_{ikl}),\ \sigma^2 \right) \tag{24}$$

TABLE 3 Out-of-sample (average) predictive log-likelihood for 25% test data, $\log p(\mathbf{y}_{kl}^*|\mathbf{x}_{kl}^*)$. Here, l corresponds to the component label (alternator $l = 1$ or turbocharger $l = 2$) while k is the subfleet label for each component. (The complete log-likelihood considers all groups and components $\mathcal{L}$).

Alternators: $l = 1$							
Model	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
CP	−24.13	0.84	−4.49	−4.02	−2.41	−2.77	−6.50
CRL	79.29	49.50	20.07	11.38	4.24	2.94	**5.47**
STL	150.24	94.24	57.66	47.04	8.51	−3.17	0.95
MTL	**164.54**	**96.98**	**62.79**	**57.51**	**24.7**	**11.12**	−9.13
Turbochargers: $l = 2$							
Model	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$\mathcal{L}$
CP	−9.85	0.35	−1.16	−0.23	−0.46	−4.00	−58.83
CRL	46.96	23.27	15.13	8.39	9.85	10.64	287.12
STL	**90.37**	48.35	21.63	17.13	**13.73**	23.74	570.41
MTL	81.34	**53.14**	**35.97**	**23.94**	11.2	**32.17**	**646.28**

where $\boldsymbol{\theta}_{k,l} = \{\boldsymbol{\alpha}_{k,l}, \boldsymbol{\beta}_l, \boldsymbol{\mu}_\alpha, \boldsymbol{\sigma}_\alpha, \sigma\}$ is the parameter set indexed to group k and component l. Figure 14 plots the mean and standard deviation of samples drawn from the posterior distribution of the extended population model (compared to independent turbocharger models). By specifying component-specific weights $\boldsymbol{\beta}_l$ the representation of uncertainty improves when extrapolating in the turbocharger domain. Reductions in the posterior predictive distribution are also observed $p(\mathbf{y}_{kl}^*|\mathbf{x}_{kl}^*)$ (ignoring other conditionals) for alternator tasks ($l = 1$) since the population data have been extended for the linear component. Likewise, variance reductions are observed in the posterior distributions of the intercept and slope, visualized in Appendix F. Quantitatively, the average reduction in standard deviation for the (interpretable) linear weights is 51% and 67% for the (turbocharger) slopes and intercepts, respectively.

Fleet-level inference improves the (bootstrapped) predictive log-likelihood from 570 to 646, compared to STL, highlighting improvements in predictive capability for the combined fleet over both components. The task-wise predictive likelihood is presented in Table 3 for the alternator ($l = 1$) and turbocharger ($l = 2$) domains, compared to the same benchmarks. Note, however, that the likelihood fails to increase from STL for certain alternator tasks ($k = 1$ or 5) reiterating the risk of negative transfer in the extended model. Ideally, the data set should be much larger to determine if negative transfer has occurred and whether the current assumptions are appropriate. As before, while CRL improves on CP the adaption approach is not suitable for the task set, and predictions remain worse than STL. The sparsity of measurements prohibits reliable transformations of the source data into the target domain.

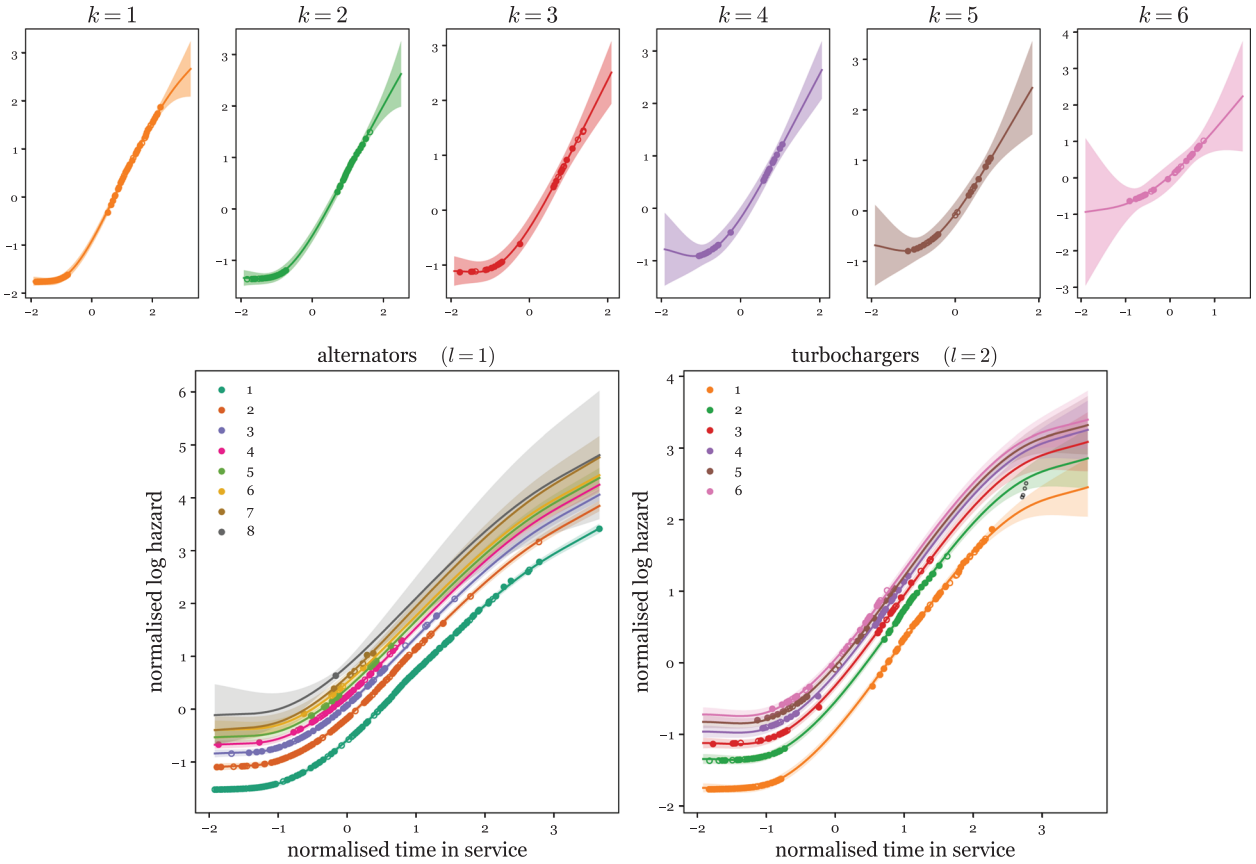

FIGURE 14 Posterior predictive distribution, the mean and three-sigma deviation for: (top) K independent regression models of turbocharger hazard $p(\mathbf{y}^*_{kl}|\mathbf{x}^*_{kl}, \mathbf{x}_{kl}, \mathbf{y}_{kl})$, (bottom) multitask learning with mixed effects for all turbocharger and alternator tasks $p(\mathbf{y}^*_{kl}|\mathbf{x}^*_{kl}, \{\{\mathbf{x}_{kl}, \mathbf{y}_{kl}\}_{k=1}^{K_l}\}_{l=1}^{L})$

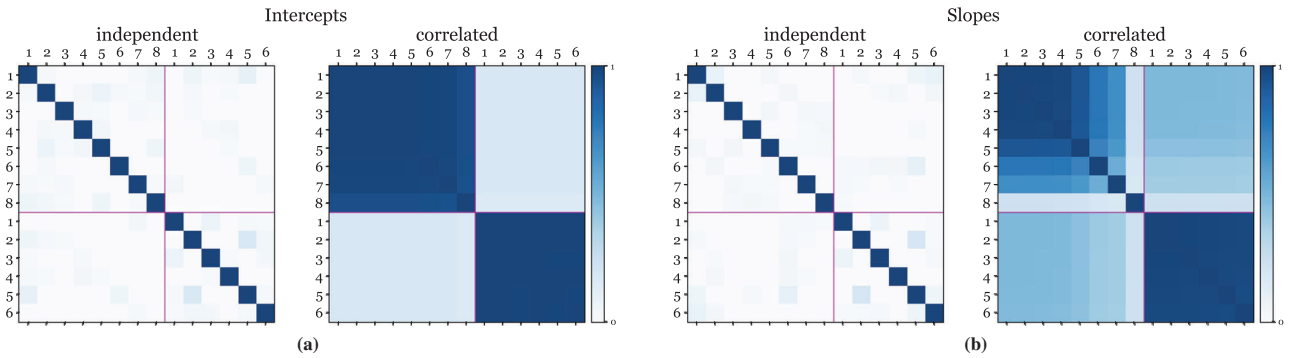

FIGURE 15 Pearson correlation coefficient of the conditional posterior distribution for the linear coefficients $\boldsymbol{\alpha}_k$ (slopes and intercepts). Purple lines separate the alternator tasks (up to 8) and turbocharger tasks (up to 6)

Figure 15 is insightful since it informs which correlations in the hierarchy *transfer* or *share* information between the subfleet (k) or component (l) groups. The heat-map corresponds to the Pearson correlation coefficient of the posterior distribution between variables that share parent nodes in the graphical model (i.e. $\boldsymbol{\alpha}_{kl}$)—these correlations enable MTL. Intuitively, Figure 15a shows increased correlation between the intercepts of the same component, with two clear blocks of 8×8 (alternators) and 6×6 (turbochargers). The intercept correlation

structure is interpretable since components of the same type are likely to have a correlated baseline hazard.

The slope correlation structure in Figure 15b is more descriptive. In the top left block, the alternators are less correlated as domains become more sparse (from $1 \rightarrow 8$); this makes sense since the level of correlation is reduced where there are fewer data to support task correlation. The effect is most obvious for $k = 8$ (alternators), which only has a single training point. In both Figure 15a and b, the structured covariance of $\boldsymbol{\alpha}_{kl}$ highlights how inter-task correlation contributes to variance reduction in the fleet model.

5.6 | Practical implications

In the field, use-type labels could be used to make sub-fleet (rather than global) predictions, which has major implications when informing efficiency or safety-critical interactions with the fleet. For example, task-specific estimations of remaining useful life would be associated with less uncertainty, and the hierarchical model allows both population estimates (from the generating distributions) and task-specific estimates. These multilevel predictions present a key contribution of this work; in turn, a multilevel decision process could be designed for more reliable interactions with the fleet—such as vehicle servicing or component replacement. A hypothetical decision process is demonstrated in the next case study.

6 | WIND FARM POWER PREDICTION

To demonstrate the wide applicability of hierarchical models, power prediction is presented for a wind farm case study. Figure 16 shows power curve data, including curtailments, provided by Visualwind and recorded from three operational turbines. The turbines are the same make and model but in different locations. As before, the data are normalized in view of data sensitivity and certain (specific) details are omitted—the same comments regarding interpretability, data truncation, and censorship apply. The work in Bull et al. (2021) demonstrates a suitable method to represent similar normal and curtailed functions in a combined model; however, each function f_k is assumed independent—in turn, there is no knowledge transfer between task parameters. Here, knowledge transfer is enabled by correlating the regression models in a hierarchical formulation.

There are 10,581 observations in total. The data were labeled in weekly subsets, according to turbine $k \in \{1, 2, 3\}$ and operational condition (normal or curtailed) $l \in \{1, 2\}$. Each point corresponds to a 10-min average of power y_{ikl}

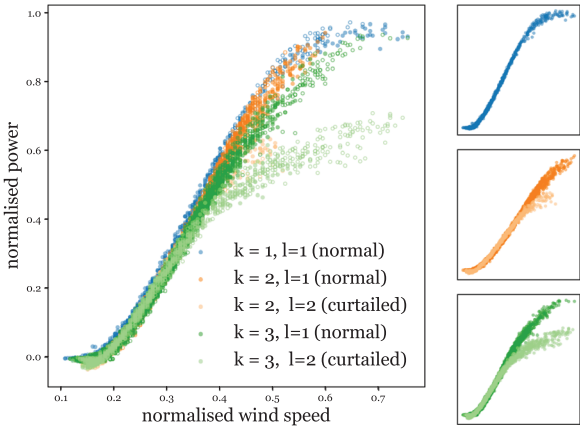

FIGURE 16 Power-curve data from three $k \in \{1, 2, 3\}$ wind turbines of the same make and model. Relationships correspond to normal $l = 1$ and ideal $l = 2$ operation

and wind speed x_{ikl}. The first turbine has 2 weeks of data, the second has 4 weeks, and the third has 11.5 weeks. Missing values and very sparse outliers were removed from the data set (using the local outlier factor algorithm; Breunig et al., 2000). Since the first turbine presents a normal power curve only ($l = 1$) there is a total of five tasks, $\sum_{l=1}^{L} K_l = K_1 + K_2 = 3 + 2 = 5$. As before, specific tasks have less data than others, with the number of observations per group as follows:

	N_{1l}	N_{2l}	N_{3l}	$\sum_{k=1}^{K_l}$
Normal ($l = 1$)	1075	1869	5845	8789
Curtailed ($l = 2$)	-	637	1155	1792

The proportions of training data are listed below. The observations remain ordered to test generalization to measurements from later operational life.

$k = 1$	$k = 2$	$k = 3$
90%	66%	66%

The splits are intentionally inconsistent, to allow a combined inference to leverage information from the data-rich tasks (with historical data) to support sparse tasks (systems recently in operation). In particular, referring to Figure 16, the normal data from the first turbine ($k = 1, l = 1$: dark blue) should support the sparse normal tasks ($k \in \{2, 3\}$: dark orange and green); while the data-rich curtailment from the third turbine (light green) should support the curtailed relationship of the second turbine (light orange).

6.1 | Task regression formulation

A standard power curve model assumes segmented linear regression (Lydia et al., 2014). A similar formulation is adopted here,

$$P(x_i) = \begin{cases} 0 & x_i < p \\ m_1(x_i - p) & p < x_i < q \\ m_2(x_i - q) + m_1(q - p) & q < x_i < r \\ P_m & x_i > r \end{cases}$$

$$m_2 \triangleq \frac{P_m - m_1(q-p)}{(r-q)} \tag{25}$$

Although simple, (25) presents interpretable parameters—visualized in Appendix C. p is the cut-in speed and r is the rated speed (for normal operation); the change point q corresponds to the initiation of the limit to maximum power P_m (where $p < q < r$). The gradients m_1 and m_2 approximate the near-linear response between p-q and q-r, respectively. The second change point and gradient $\{q, m_2\}$ enable *soft* curtailments, rather than a hard limit at maximum power P_m.

6.2 | Mixed effects and prior formulation

From knowledge of turbine operation, the expected power before cut-in should be zero for all turbines (i.e., a fixed effect). The cut-in speed p is also tied as a fixed effect and learnt at the population level since all turbines have the same design. Similarly, the max power P_m is tied between operational labels $l \in \{1, 2\}$ such that one parameter is learnt for the normal tasks ($l = 1$) and one for the curtailed tasks ($l = 2$). Conversely, the change points $\{q, r\}$ and gradients $\{m_1, m_2\}$ are assumed to be correlated between all tasks, that is, correlated via shared parent nodes. In turn, one would expect the curtailed relationships ($l = 2$) to be more correlated (and share more information) than the normal relationships ($l = 1$) and vice versa.

The (expected) tasks are summarized as segmented mixed effects,

$$\left\{ \{\hat{y}_i^{(kl)} = \cdots \right.$$

$$\left. \begin{cases} 0 & x_i < p \\ m_1^{(kl)}(x_i - p) & p < x_i < q^{(kl)} \\ m_2^{(kl)}(x_i - q^{(kl)}) + m_1^{(kl)}(q^{(kl)} - p) & q^{(kl)} < x_i < r^{(kl)} \\ P_m^{(l)} & q^{(kl)} < x_i < r^{(kl)} \end{cases} \cdots \right\}_{k=1}^{K_l} \Bigg\}_{l=1}^{L} \tag{26}$$

$$m_2^{(kl)} \triangleq \frac{P_m^{(l)} - m_1^{(kl)}(q^{(kl)} - p)}{(r^{(kl)} - q^{(kl)})} \tag{27}$$

where the fixed effects are green and the random effects are purple. Each segment of the regression could be presented in a similar formulation to (23) such that each component is a standard varying intercepts/slope model (Gelman et al., 2013). Matrix notation is avoided, however, to present the model (and priors) around parameters $\{P_m, m_1, m_2, p, q, r\}$. The likelihood of the response can be specified using (27),

$$y_{ikl} | x_{ikl}, \theta_{kl} \sim \mathrm{N}\left(\hat{y}_i^{(kl)},\ \sigma^2\right) \tag{28}$$

where $\theta_{kl} = \{P_m^{(l)}, m_1^{(kl)}, p, q^{(kl)}, r^{(kl)}\}$ is the parameter set indexed to turbine k and curtailment l.

Given their interpretability, weakly informative priors are postulated for each parameter. For the change points,

$$p \sim \mathrm{N}(\mu_p, \sigma_{cp}^2), \quad q^{(kl)} \sim \mathrm{N}(\mu_q, \sigma_{cp}^2), \quad r^{(kl)} \sim \mathrm{N}(\mu_r, \sigma_{cp}^2)$$

$$\mu_p \sim \mathrm{N}(0.2, 0.5), \quad \mu_q \sim \mathrm{N}(0.4, 0.5), \quad \mu_r \sim \mathrm{N}(0.6, 0.5)$$

$$\sigma_{cp} \sim \mathrm{IG}(1, 1) \tag{29}$$

These priors reflect that change points are expected to occur at regular intervals across the input domain with relatively high variance (relative to a normalized scale). The priors for gradient and maximum power are

$$m_1^{(kl)} \sim \mathrm{N}(\mu_{m_1}, \sigma_{m_1}^2)$$

$$\mu_{m_1} \sim \mathrm{N}(2.5, 0.5), \quad \sigma_{m_1} \sim \mathrm{IG}(1, 1) \tag{30}$$

$$P_m^{(1)} \sim \mathrm{N}(1, 0.1), \qquad P_m^{(2)} \sim \mathrm{N}(0.8, 0.1) \tag{31}$$

These distributions postulate the expected gradient m_2 in a normalized space; unit max power $P_m^{(1)}$ for normal operation; and a typical 80% curtailment (Bull et al., 2021) for the limited output $P_m^{(2)}$. No prior is required for m_2 since it is specified by $\{P_m, m_1, p, q, r\}$ in (27). As with the truck-fleet example, the IG(1, 1) distributions weakly encourage intertask correlations, such that the prior intentionally overestimates the deviation between task parameters. Similarly, the posterior can be specified using (21), where $p([\mathbf{y}_k] \mid \boldsymbol{\Theta})$ is indexed by (28) and the joint prior $p(\boldsymbol{\Theta})$ is defined using (29) to (31). As before, this is intractable and inferred with MCMC.

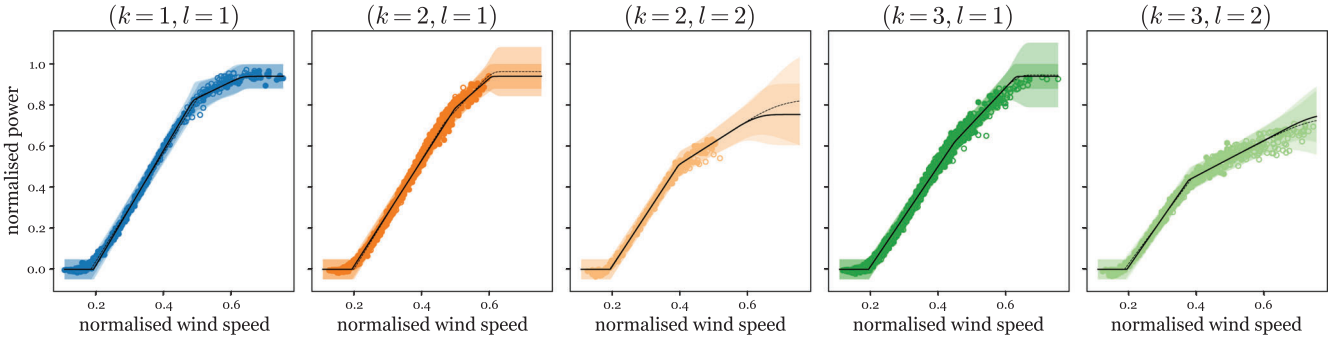

FIGURE 17 Posterior predictive distribution, the mean, and three-sigma deviation for: (light shading, dashed line) K independent power curve models $p(\mathbf{y}^*_{kl}|\mathbf{x}^*_{kl}, \mathbf{x}_{kl}, \mathbf{y}_{kl})$. (dark shading, solid line) multitask learning with mixed effects $p(\mathbf{y}^*_{kl}|\mathbf{x}^*_{kl}, \{\{\mathbf{x}_{kl}, \mathbf{y}_{kl}\}_{k=1}^{K_l}\}_{l=1}^{L})$

TABLE 4 Predictive log-likelihood $\log p(\mathbf{y}^*_{kl}|\mathbf{x}^*_{kl})$. Here l corresponds to the operating condition (normal $l = 1$, or curtailed $l = 2$) while k is turbine identifier.

Method	$k=1,\ l=1$	$k=2,\ l=1$	$k=3,\ l=1$	$k=2,\ l=2$	$k=3,\ l=2$	$\mathcal{L}$
CP	−168	1555	4681	594	452	7114
CRL	52	1451	3359	282	−6	5138
STL	202	**1619**	5147	538	**722**	8229
MTL	**218**	1599	**5206**	**549**	686	**8258**

6.3 | Results

Figure 17 shows posterior predictive distribution from fleet-level inference—compared to independent STL models, plotted with light shading. Intuitively, variance reduction is most obvious for sparse or poorly described domains (orange and dark green). There is an overall increase in the predictive likelihood when fleet modeling, compared to STL, from 8229 to 8258. Table 4 quantifies changes in task-wise predictions compared to the benchmarks: There is a likelihood increase in all domains other than $(k = 2, l = 1)$ and $(k = 3, l = 2)$. It is believed that reductions occur since the model is constrained such that, to maximise the overall likelihood, the performance in data-rich domains is reduced in a trade-off. In other words, the prior belief is best suited to data-rich tasks—when the prior becomes more informed by data, it becomes less suitable in data-rich domains; instead, the prior represents the population. (Consider that the overall likelihood $\mathcal{L}$ increases, despite task-wise fluctuations.) To combat this, uninformative priors should be considered (Gelman et al., 2013); these are discussed in Section 7.

CRL performs less competitively in the wind turbine example since the measurement distributions shift significantly between each task, training, and testing (testing data correspond to following weeks). In particular, when the source data represent a more complete power curve, the alignment with sparse domains becomes partial, and CRL can produce unreasonable embeddings.

Figure 18 shows the posterior distribution of the parameters inferred at the independent and fleet level. The cut-in speed q moves toward an average of the independent models, with reduced variance; this should be expected since q becomes tied as a population estimate. The change points q cluster intuitively, such that the normal and curtailed tasks form two groups (dark and light shades). The estimated r parameters are significantly improved through partial pooling—in particular, the green and orange domains shift much further from the weakly informative prior. There is a notable reduction in the variance across all tasks for the slope estimate m_2. The average reduction in standard deviation across these parameters is 25%.

Figure 19 presents insights relating to maximum power estimates P_m. The tied parameter for the normal maximum $P_m^{(k,1)}$ moves toward the data-rich estimate (blue) while the curtailed maximum $P_m^{(k,2)}$ moves toward an average of the relevant tasks (where $l = 2$). In both operating conditions, parameter tying enables the move from vague posteriors to distributions with clear expected values. The average reduction in standard deviation for the normal maximum is 82%, alongside 37% for the curtailed maximum.

Finally, Figure 20 plots the Pearson correlation coefficient of the pair-wise conditionals of q between tasks. (q is presented since it is the most structured/insightful.) It

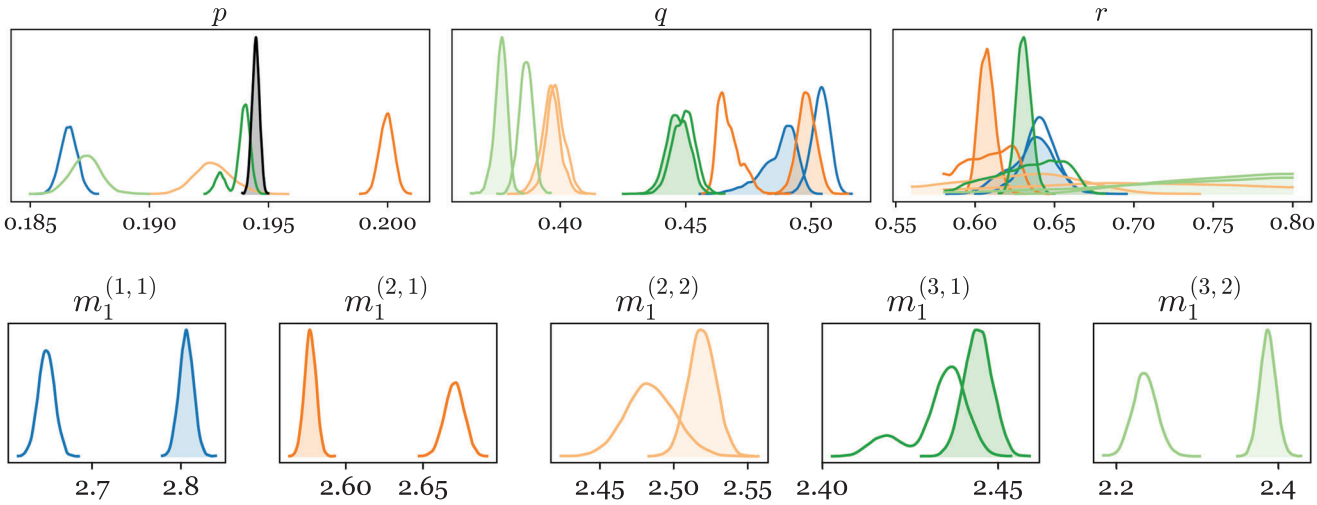

FIGURE 18 Changes in the posterior distribution: the cut-in speed p, initiation of curtailment q, rated speed r, and linear slope m_1. Independent models (hollow) compared to population-level modeling (shaded). When the parameter is tied (or fixed) the distributions are black

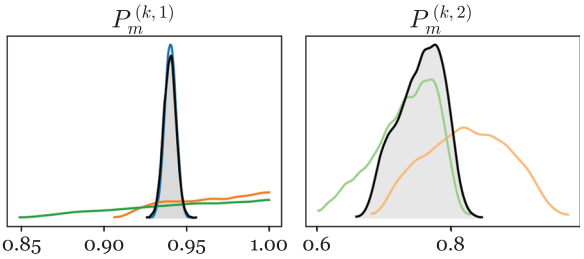

FIGURE 19 Changes in the posterior distribution of the independent models (hollow) compared to population-level modeling (shaded)

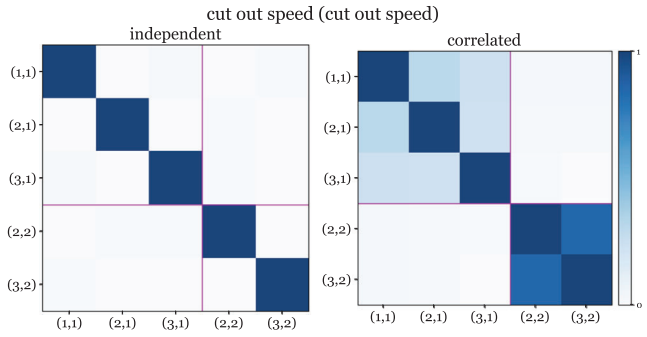

FIGURE 20 Pearson correlation coefficient of the conditional posterior distribution for the rated wind speed q, tick labels correspond to (k, l). Purple lines separate the normal ($l = 1$) from the curtailed task parameters ($l = 2$)

is clear that, by moving to a hierarchical model, the correlation between related tasks is appropriately captured, with two distinct blocks associated with the normal and curtailed groups.

6.4 | Practical implications: Decision analysis

In practice, probabilistic predictions from the power model can be used to support decisions at any level of the hierarchy, including the population level. For example, population-level decisions are useful if the operator does not wish to commit to interacting with a specific turbine.

Consider a decision problem, whereby an operator must commit to delivering a minimum power in some upcoming time window. This involves decision making under uncertainty, and the formal (statistical) procedure to identify the expected optimal action requires a probabilistic quantification of wind speed and power output. The latter can be achieved by sampling from the posterior predictive distribution at the population level, that is, $p(\mathbf{y}^* \mid \mathbf{x}^*, \boldsymbol{\theta}_l)$, where $\boldsymbol{\theta}_l = \{P_m^{(l)}, m_1^{(l)}, p, q^{(l)}, r^{(l)}\}$ is sampled directly from the generating distributions. Figure 21 is an example of such a prediction for a given wind speed.

Predictions at this level of the hierarchy are useful since they assume the operator cannot commit to a specific turbine (at this stage). Such predictions would not be available from domain-specific (independent) models; conversely, CP (or domain adapted) predictions would not formally consider the additional variability associated with nonspecific turbine identity.

In this example, the operator has three options, each associated with a payout (positive utility) upon successful delivery of power and a penalty fine (negative utility) if the turbine generates insufficient power—these values are presented in Table 5. A prior probabilistic model of (normalized) wind speed $\mathbf{x}_{pr}$ is shown in Figure 22, as

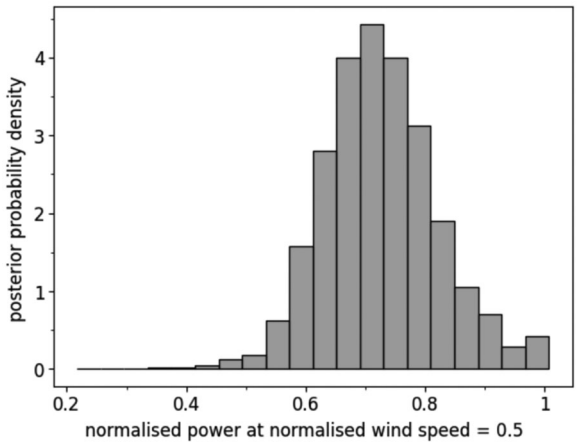

FIGURE 21 Samples from the posterior predictive distribution of normalized power, at an arbitrary input of normalized wind speed = 0.5

TABLE 5 Financial outcomes of decision analysis

Power level	Payout	Penalty fine
L_0: 0.0	0.0	−0.0
L_1: 0.5	0.3	−0.3
L_2: 0.75	0.75	−1.0

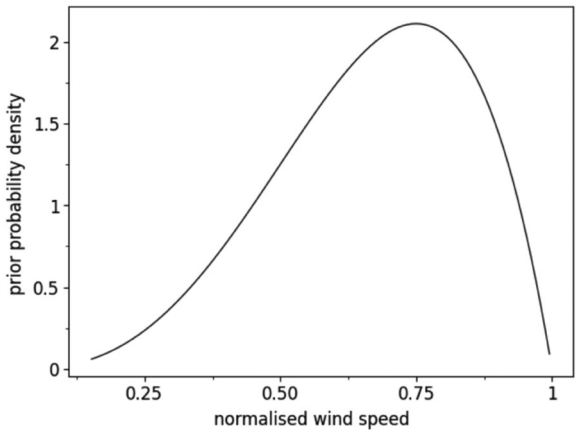

FIGURE 22 The prior distribution of normalized wind speed

described by,

$$\mathbf{x}_{pr} \sim \text{Beta}(4, 2) \tag{32}$$

(In practice, this information would likely come from a forecasting model.)

Figure 23 shows the decision-event tree representation of the problem. Here, the square (decision) node P_L is associated with the available power commitments in Table 5, such that $P_L = \{L_0, L_1, L_2\}$. The circular (probabilistic) node $(\mathbf{y}^* \mid \mathbf{x}_{pr})$ is the probabilistic prediction of power, given the prior model of wind speed. Finally, the

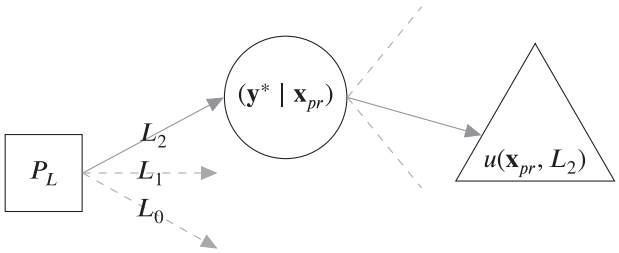

FIGURE 23 Decision-event tree representation of power-level decision analysis

triangular (utility) node shows the expected consequence of the decision.

For each instance, the expected optimal action $P_L^* \in \{L_0, L_1, L_2\}$ and associated expected utility $E[u(\mathbf{x}_{pr}, P_L^*)]$ are calculated,

$$P_L^* = \text{argmax}_{P_L} \text{E}[u(\mathbf{x}_{pr}, P_L)] \tag{33}$$

$$\text{E}[u(\mathbf{x}_{pr}, P_L^*)] = \text{E}[\text{pay}_{P_L^*}] - \text{E}[\text{penalty}_{P_L^*}] \tag{34}$$

where

$$\text{E}[\text{pay}_{P_L}] = P(\mathbf{y}_* \geq P_L) \times \text{payout}_{P_L} \tag{35}$$

$$\text{E}[\text{penalty}_{P_L}] = P(\mathbf{y}_* < P_L) \times \text{penaltyfine}_{P_L} \tag{36}$$

This information can then be used to rank decision alternatives (Schlaifer & Raiffa, 1961). For example, in the prior decision tree (Figure 23) the path associated with the highest power level L_2 is optimal (i.e., $P_L^* = L_2$)—this was found to have the highest expected utility of 0.33, compared with 0.0 for L_0, and 0.246 for L_1.

A further application quantifies the expected value of data collection activities. Figure 24 extends the problem in Figure 23 to include another decision M: whether to measure wind speed (m) or not ($\bar{m}$). In the case where measurements are taken, predictions can be made using the new data $\mathbf{x}_m$. A so-called *preposterior* decision analysis (Berger, 2013; Jordaan, 2005) can be completed, by sampling from the prior model to generate hypothetical measurements.

When assuming *perfect* data, whereby each measurement removes all uncertainty from wind speed (32), the expected (preposterior) utility is 0.566. The difference in expected utility—with (m) or without ($\bar{m}$) wind measurement—is the expected value of the data $\mathbf{x}_m$ in the context of solving the decision problem. This expected value of perfect information (VoPI) can be estimated using Monte Carlo sampling,

$$VoPI = \frac{1}{N} \sum_{i=1}^{N} \left(E[u(\mathbf{x}_m, P_L^*)] \right) - E[u(\mathbf{x}_{pr}, P_L^*)] \tag{37}$$

Here, the VoPI is 0.236. The results are presented in Figure 25, which shows a histogram of expected utilities

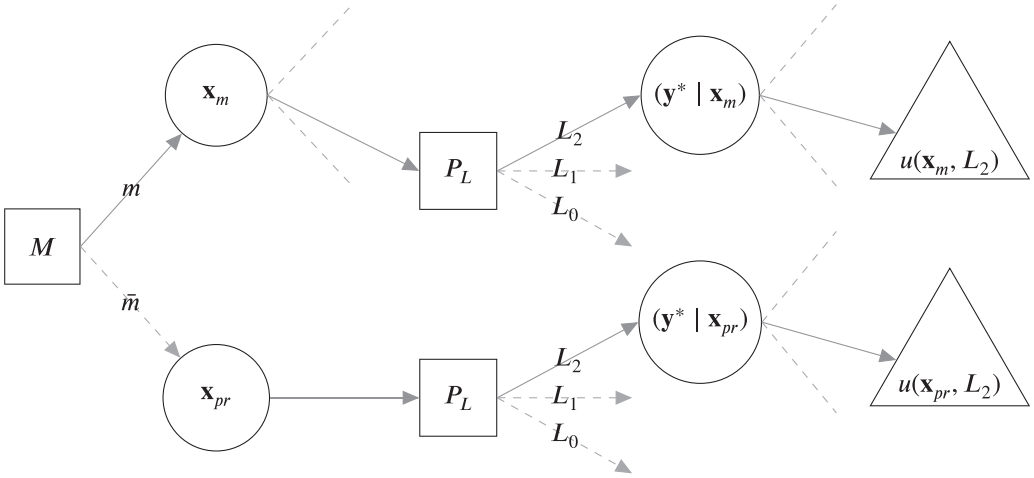

FIGURE 24 Decision-event tree representation of the value of information analysis

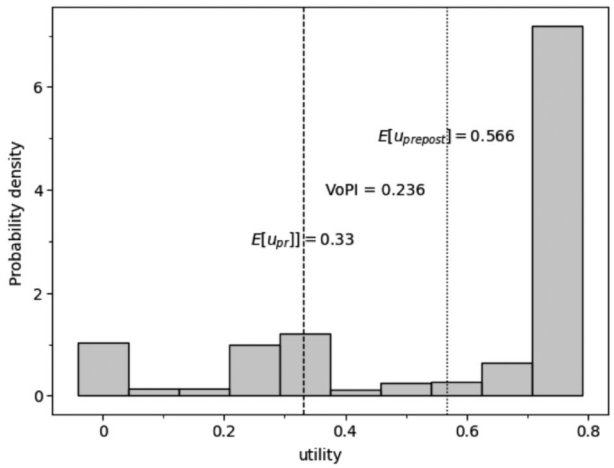

FIGURE 25 Expected utilities associated with hypothesized wind speed measurements. The expected VoPI is shown as the difference between expected utilities with ($E[u_{prepost}]$) and without ($E[u_{prior}]$) the wind speed data

associated with each of the hypothesized, perfect measurements (samples from the prior model). The mean of these values $E[u_{prepost}]$ is labeled next to the dotted line. The expected utility without the data $E[u_{pr}]$ is labeled as a dashed line, and the difference (37) is the expected value of the data.

To summarize, hierarchical Bayesian modeling has provided a full quantification of uncertainty, reflective of different asset subgroups and classes. In turn, the model enables a formal downstream analysis of variable interactions and integration with a utility-based decision process (demonstrated here). The implications are significant since various concepts can be quantified, for example, the expected *optimal action* or the *value of data collection* activities.

7 | CONCLUDING REMARKS

Hierarchical Bayesian modeling with mixed effects is demonstrated as an effective method of sharing information between models of fleets of assets in engineering. Parameter estimation and predictive capabilities are improved (for the combined fleet) in two case studies, utilizing the same flexible MTL framework. Important considerations are discussed when formulating each population model: prior elicitation, mixed-effects formulation, and negative transfer—these concepts are critical to the success of population-level inference.

The proposed hierarchical methodology is desirable since it enables downstream analyses of the fleet model. The method is used to determine which asset models are correlated for which interpretable parameter, at various groupings (e.g., operating condition, system-specific, population-wide). The multivariate (and multilevel) uncertainty quantification enabled by the model is then propagated through a demonstrative decision analysis for the second case study, to consistently and coherently identify expected optimal actions. The expected value of data collection is also quantified, in the context of the decision problem and the underlying model.

The first application concerns the survival analysis of turbocharger and alternator components in an operational fleet of trucks (maintained by Scania). A semiparametric hazard curve model is improved through partial pooling and parameter tying (15% and 13% increases in predictive log-likelihood) where selected parameters are inferred at the population level, rather than vehicle subgroups. The method builds on engineering intuition since correlations in the hierarchy can be inspected to determine which groups of vehicles or components are correlated for

which effects in the survival model (i.e., interpretable parameters).

The second study presents power prediction for a group of wind turbines. The SCADA monitoring data were provided by Visualwind, measured from the same model of turbine in different locations. Correlated power curve models are learnt as a segmented (piece-wise) linear regression, described by interpretable parameters. By moving to a population-level inference, parameter estimation is improved, as well as model generalization (for the combined population estimates). In particular, the estimation of maximum power is significantly improved for turbines with fewer data and recently in operation (up to 82% reduction in the standard deviation of maximum output prediction).

The success of these models depends on the reliability of the domain knowledge encoded in the prior distributions. In this case, priors were postulated as weakly informative, since interpretable parameters and domain expertise allowed sensible prior elicitation. In turn, an appropriate level of knowledge transfer could be determined automatically, given the model and the data, reducing the risk of negative transfer. When such elicitation is infeasible, future work should consider the use of uninformative priors (Gelman, 2006), especially for the (variance) parameters that control the level of correlation between tasks.

Future work should consider an objective method to categorize subfleet data in a practical setting; this might include clustering assets from specification or operations data. The labeling of data into distinct tasks can be nontrivial in an engineering setting and requires investigation. An interpretable, MTL procedure could also be developed around other modes of learning, such as dynamic classification (Rafiei & Adeli, 2017), ensemble learning (Alam et al., 2020), fuzzy methods (Adeli & Hung, 1995), or reinforcement learning (Wilson et al., 2007)—to investigate more complex feature types, decision problems, and larger data sets. Finally, extending the multilevel model to capture parameter relationships over the fleet should prove insightful; for example, if the coefficients of the power model were regressed on spatial/temporal inputs for the wind farm, one could simulate (sample) more varied hypothetical members of the population at certain locations or timescales.

ACKNOWLEDGMENTS

A. B. Duncan, D. Di Francesco, and L. A. Bull were supported by Wave 1 of The UKRI Strategic Priorities Fund under the EPSRC Grant EP/W006022/1, particularly the *Ecosystems of Digital Twins* theme within that grant and The Alan Turing Institute. M. Dhada was supported by the Next Generation Converged Digital Infrastructure project (EP/R004935/1) funded by the Engineering and Physical Sciences Research Council and BT. This research was supported by Scania CV (Sweden) and Visualwind (UK). The authors would also like to thank Dr. Paul Gardner and Jack Poole for their helpful conversations while writing this paper.

REFERENCES

Adeli, H., & Hung, S. L. (1995). *Machine learning: neural networks, genetic algorithms, and fuzzy systems*. John Wiley & Sons, Inc.

Alam, K. M., Siddique, N., & Adeli, H. (2020). A dynamic ensemble learning algorithm for neural networks. *Neural Computing and Applications*, *32*(12), 8675–8690.

Amezquita-Sanchez, J. P., Park, H. S., & Adeli, H. (2017). A novel methodology for modal parameters identification of large smart structures using music, empirical wavelet transform, and Hilbert transform. *Engineering Structures*, *147*, 148–159.

Berger, J. O. (2013). *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media.

Birolini, A. (2013). *Reliability engineering: Theory and practice*. Springer Science & Business Media.

Bonilla, E. V., Chai, K., & Williams, C. (2007). Multi-task Gaussian process prediction. In J. Platt, D. Koller, Y. Singer, & S. Roweis (Eds.), *Advances in Neural Information Processing Systems*, (Vol. 20). Curran Associates, Inc.

Bontekoning, M., Perez-Moreno, S. S., Ummels, B., & Zaaijer, M. (2017). Analysis of the reduced wake effect for available wind power calculation during curtailment. *Journal of Physics: Conference Series*, *854*, 012004.

Breunig, M. M., Kriegel, H. P., Ng, R. T., & Sander, J. (2000). LOF: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on management of data* (Vol. 29, pp. 93–104). New York, NY, USA: Association for Computing Machinery.

Bull, L., Gardner, P., Dervilis, N., Papatheou, E., Haywood-Alexander, M., Mills, R., & Worden, K. (2021). On the transfer of damage detectors between structures: An experimental case study. *Journal of Sound and Vibration*, *501*, 116072.

Bull, L., Gardner, P., Gosliga, J., Rogers, T., Dervilis, N., Cross, E., Papatheou, E., Maguire, A., Campos, C., & Worden, K. (2021). Foundations of population-based SHM, part I: Homogeneous populations and forms. *Mechanical Systems and Signal Processing*, *148*, 107141.

Bull, L., Gardner, P., Rogers, T., Dervilis, N., Cross, E., Papatheou, E., Maguire, A., Campos, C., & Worden, K. (2021). Bayesian modelling of multivalued power curves from an operational wind farm. *Mechanical Systems and Signal Processing*, *169*, 108530.

Bull, L., Rogers, T., Wickramarachchi, C., Cross, E., Worden, K., & Dervilis, N. (2019). Probabilistic active learning: An online framework for structural health monitoring. *Mechanical Systems and Signal Processing*, *134*, 106294.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P., & Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*(1).

Carrillo, C., Montaño, A. O., Cidrás, J., & Díaz-Dorado, E. (2013). Review of power curve modelling for wind turbines. *Renewable and Sustainable Energy Reviews*, *21*, 572–581.

Dhada, M., Girolami, M., & Parlikad, A. K. (2020). Anomaly detection in a fleet of industrial assets with hierarchical statistical modeling. *Data-Centric Engineering*, *1*, E21.

Di Francesco, D., Chryssanthopoulos, M., Faber, M. H., & Bharadwaj, U. (2021). Decision-theoretic inspection planning using imperfect and incomplete data. *Data-Centric Engineering*, *2*, E18.

Dorafshan, S., Thomas, R. J., & Maguire, M. (2018). Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete. *Construction and Building Materials*, *186*, 1031–1045.

Gao, Y., & Mosalam, K. M. (2018). Deep transfer learning for image-based structural damage recognition. *Computer-Aided Civil and Infrastructure Engineering*, *33*(9), 748–768.

Gardner, P., Bull, L., Dervilis, N., & Worden, K. (2021). Overcoming the problem of repair in structural health monitoring: Metric-informed transfer learning. *Journal of Sound and Vibration*, *510*, 116245.

Gardner, P., Bull, L., Dervilis, N., & Worden, K. (2022). On the application of kernelised Bayesian transfer learning to population-based structural health monitoring. *Mechanical Systems and Signal Processing*, *167*, 108519.

Gardner, P., Bull, L., Gosliga, J., Dervilis, N., & Worden, K. (2021). Foundations of population-based SHM, part III: Heterogeneous populations–mapping and transfer. *Mechanical Systems and Signal Processing*, *149*, 107142.

Gardner, P., Fuentes, R., Dervilis, N., Mineo, C., Pierce, S., Cross, E., & Worden, K. (2020). Machine learning at the interface of structural health monitoring and non-destructive evaluation. *Philosophical Transactions of the Royal Society A*, *378*(2182), 20190581.

Gardner, P., Liu, X., & Worden, K. (2020). On the application of domain adaptation in structural health monitoring. *Mechanical Systems and Signal Processing*, *138*, 106550.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, *1*(3), 515–534.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., & Rubin, D. (2013). *Bayesian data analysis* (3rd ed.). Chapman and Hall/CRC.

Gelman, A., & Hill, J. (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.

Gosliga, J., Gardner, P., Bull, L., Dervilis, N., & Worden, K. (2021). Foundations of population-based SHM, part II: Heterogeneous populations—Graphs, networks, and communities. *Mechanical Systems and Signal Processing*, *148*, 107144.

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*(1), 1593–1623.

Huang, Y., & Beck, J. L. (2015). Hierarchical sparse Bayesian learning for strucutral health monitoring with incomplete modal data. *International Journal for Uncertainty Quantification*, *5*(2), 139–169.

Huang, Y., Beck, J. L., & Li, H. (2019). Multitask sparse Bayesian learning with applications in structural health monitoring. *Computer-Aided Civil and Infrastructure Engineering*, *34*(9), 732–754.

Hur, S. h., & Leithead, W. (2014). Curtailment of wind farm power output through flexible turbine operation using wind farm control. In *European wind energy association annual event (EWEA 2014)* (pp. 1–9).

Jang, K., Kim, N., & An, Y. K. (2019). Deep learning–based autonomous concrete crack evaluation through hybrid image scanning. *Structural Health Monitoring*, *18*(5-6), 1722–1737.

Jordaan, I. (2005). *Decisions under uncertainty: Probabilistic analysis for engineering decisions*. Cambridge University Press.

Kim, N. H., An, D., & Choi, J. H. (2017). *Prognostics and health management of engineering systems*. Springer International Publishing.

Kreft, I. G., & De Leeuw, J. (1998). *Introducing multilevel modeling*. Sage.

Li, X., Zhang, W., Ding, Q., & Sun, J. Q. (2019). Multi-layer domain adaptation method for rolling bearing fault diagnosis. *Signal Processing*, *157*, 180–197.

Li, Y., Bao, T., Chen, Z., Gao, Z., Shu, X., & Zhang, K. (2021). A missing sensor measurement data reconstruction framework powered by multi-task Gaussian process regression for dam structural health monitoring systems. *Measurement*, *186*, 110085.

Li, Z., Park, H. S., & Adeli, H. (2017). New method for modal identification of super high-rise building structures using discretized synchrosqueezed wavelet and hilbert transforms. *The Structural Design of Tall and Special Buildings*, *26*(3), e1312.

Lydia, M., Kumar, S. S., Selvakumar, A. I., & Kumar, G. E. P. (2014). A comprehensive review on wind turbine power curve modeling techniques. *Renewable and Sustainable Energy Reviews*, *30*, 452–460.

Michau, G., & Fink, O. (2019). Domain adaptation for one-class classification: monitoring the health of critical systems under limited information. *International Journal of Prognostics and Health Management*, *10*(4).

Murphy, K. P. (2012). *Machine learning: A probabilistic perspective*. MIT Press.

O'Connor, P., & Kleyner, A. (2012). *Practical reliability engineering*. John Wiley & Sons.

Paleyes, A., Urma, R. G., & Lawrence, N. D. (2020). Challenges in deploying machine learning: A survey of case studies. *ACM Computing Surveys (CSUR)*. New York, NY: ACM.

Papadimas, N., & Dodwell, T. (2021). A hierarchical Bayesian approach for calibration of stochastic material models. *Data-Centric Engineering*, *2*, E20.

Papatheou, E., Dervilis, N., Maguire, A., Campos, C., Antoniadou, I., & Worden, K. (2017). Performance monitoring of a wind turbine using extreme function theory. *Renewable Energy*, *113*, 1490–1502.

Perez-Ramirez, C. A., Amezquita-Sanchez, J. P., Valtierra-Rodriguez, M., Adeli, H., Dominguez-Gonzalez, A., & Romero-Troncoso, R. J. (2019). Recurrent neural network model with Bayesian training and mutual information for response prediction of large buildings. *Engineering Structures*, *178*, 603–615.

Poole, J., Gardner, P., Dervilis, N., Bull, L., & Worden, K. (2022). On statistic alignment for domain adaptation in structural health monitoring. *Structural Health Monitoring*, 14759217221110441. https://doi.org/10.1177/14759217221110441

Rafiei, M. H., & Adeli, H. (2017). A new neural dynamic classification algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3074–3083.

Rodrıguez, G. (2010). *Parametric survival models*. Rapport technique, Princeton University, Princeton.

Rogers, T., Gardner, P., Dervilis, N., Worden, K., Maguire, A., Papatheou, E., & Cross, E. (2020). Probabilistic modelling of wind turbine power curves with application of heteroscedastic Gaussian process regression. *Renewable Energy*, *148*, 1124–1136.

Schlaifer, R., & Raiffa, H. (1961). *Applied statistical decision theory*. Wiley.

Seshadri, P., Duncan, A., Thorne, G., Parks, G., Dıaz, R. V., & Girolami, M. (2020). Bayesian assessments of aeroengine performance with transfer learning. *arXiv preprint arXiv:2011.14698*.

Sukhija, S., & Krishnan, N. C. (2020). Shallow domain adaptation. In H. Venkateswara, & S. Panchanathan (Eds.), *Domain adaptation in computer vision with deep learning* (pp. 23–40). Springer.

Sun, B., Feng, J., & Saenko, K. (2017). Correlation alignment for unsupervised domain adaptation. In G. Csurka (Ed.), *Domain adaptation in computer vision applications* (pp. 153–171). Springer.

Sun, Z., Barp, A., & Briol, F. X. (2021). Vector-valued control variates. *arXiv preprint arXiv:2109.08944*.

Thapar, V., Agnihotri, G., & Sethi, V. K. (2011). Critical analysis of methods for mathematical modelling of wind turbines. *Renewable Energy*, *36*(11), 3166–3177.

Tipping, M. E. (2001). Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, *1*(Jun), 211–244.

Tsialiamanis, G., Mylonas, C., Chatzi, E., Wagg, D., Dervilis, N., & Worden, K. (2022). On an application of graph neural networks in population-based shm. In R. Madarshahian, & F. Hemez (Eds.), *Data science in engineering* (Vol. 9, pp. 47–63). Springer.

Waite, M., & Modi, V. (2016). Modeling wind power curtailment with increased capacity in a regional electricity grid supplying a dense urban demand. *Applied Energy*, *183*, 299–317.

Wan, H. P., & Ni, Y. Q. (2019). Bayesian multi-task learning methodology for reconstruction of structural health monitoring data. *Structural Health Monitoring*, *18*(4), 1282–1309.

Wand, M. (2009). Semiparametric regression and graphical models. *Australian & New Zealand Journal of Statistics*, *51*(1), 9–41.

Wang, Q., Michau, G., & Fink, O. (2019). Domain adaptive transfer learning for fault diagnosis. In *2019 Prognostics and system health management conference (PHM-Paris)* (pp. 279–285). IEEE.

West, B. T., Welch, K. B., & Galecki, A. T. (2006). *Linear mixed models: A practical guide using statistical software*. Chapman and Hall/CRC.

Wilson, A., Fern, A., Ray, S., & Tadepalli, P. (2007). Multi-task reinforcement learning: A hierarchical Bayesian approach. In *Proceedings of the 24th international conference on machine learning* (pp. 1015–1022).

Worden, K., & Manson, G. (2007). The application of machine learning to structural health monitoring. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*(1851), 515–537.

Yang, W., Court, R., & Jiang, J. (2013). Wind turbine condition monitoring by the approach of SCADA data analysis. *Renewable Energy*, *53*, 365–376.

Zaccaria, V., Stenfelt, M., Aslanidou, I., & Kyprianidis, K. G. (2018). Fleet monitoring and diagnostics framework based on digital twin of aero-engines. In *Turbo Expo: Power for land, sea, and air* (Vol. 51128, pp. V006T05A021). American Society of Mechanical Engineers.

Zhang, B., Hong, X., & Liu, Y. (2020). Multi-task deep transfer learning method for guided wave-based integrated health monitoring using piezoelectric transducers. *IEEE Sensors Journal*, *20*(23), 14391–14400.

Zhang, W., Peng, G., Li, C., Chen, Y., & Zhang, Z. (2017). A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. *Sensors*, *17*(2), 425.

How to cite this article: Bull, L. A., Di Francesco, D., Dhada, M., Steinert, O., Lindgren, T., Parlikad, A. K., Duncan, A. B., & Girolami, M. (2023). Hierarchical Bayesian modeling for knowledge transfer across engineering fleets via multitask learning. *Computer-Aided Civil and Infrastructure Engineering*, *38,* 821–848.
https://doi.org/10.1111/mice.12901

APPENDIX A: B-SPLINES

Assuming uniform knot locations $x_{h+k} = x_h + \delta k$, cubic B-splines are defined as the following piece-wise cubic polynomial Gelman et al. (2013):

$$b_h(x) = \begin{cases} \frac{1}{6}u^3 & x \in (x_h, x_{h+1}), \quad u = (x - x_h)/\delta \\ \frac{1}{6}(1 + 3u + 3u^2 - 3u^3) & x \in (x_{h+1}, x_{h+2}), \quad u = (x - x_{h+1})/\delta \\ \frac{1}{6}(4 - 6u^2 + 3u^3) & x \in (x_{h+2}, x_{h+3}), \quad u = (x - x_{h+2})/\delta \\ \frac{1}{6}(1 - 3u + 3u^2 - u^3) & x \in (x_{h+3}, x_{h+4}), \quad u = (x - x_{h+3})/\delta \\ 0 & \text{otherwise} \end{cases} \tag{A.1}$$

APPENDIX B: CROSS-VALIDATION SCANIA

Figure B.1

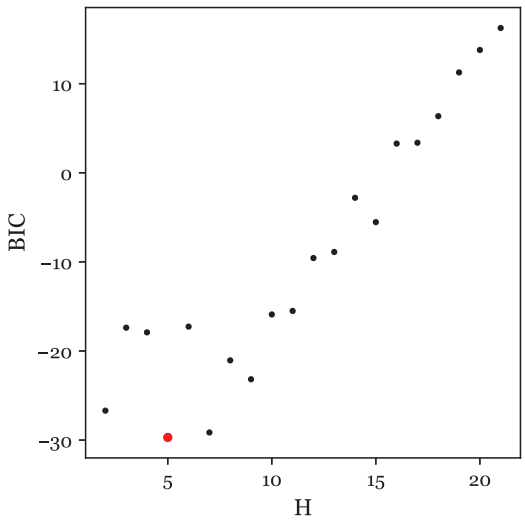

FIGURE B.1 Validation of an appropriate number of splines using the Bayesian information criterion (BIC) and 20-fold cross-validation. The best model $H = 5$ is highlighted with a red marker

APPENDIX C: SEGMENTED (PIECE-WISE) POWER CURVE MODEL

Figure C.1

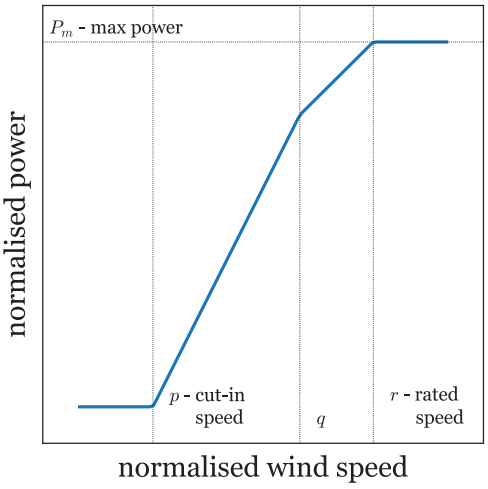

FIGURE C.1 The segmented linear power curve model, indicating interpretable parameters $\{p, q, r, P_m\}$

APPENDIX D: ZOOMED SPLINE WEIGHTS

Figure D.1

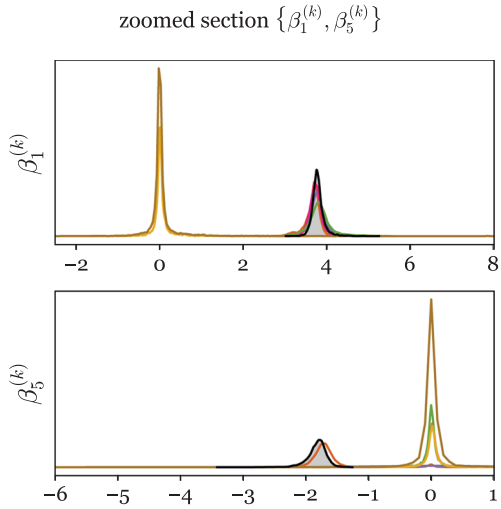

FIGURE D.1 Zoomed sections of the posterior distribution of the spline weights $\beta_h^{(k)}$ (those that deviate from zero $h \in \{1, 5\}$)

APPENDIX E: TURBOCHARGER MODEL: CONSISTENT MODEL FORMULATION

Figure E.1

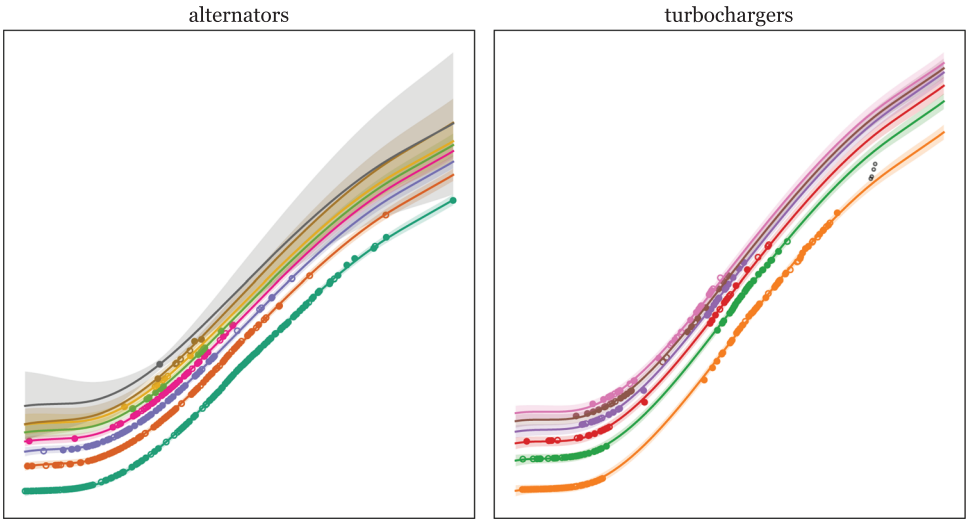

FIGURE E.1 Posterior predictive distribution $p(\mathbf{y}_k^*|\mathbf{x}_k^*, \{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^K)$: the mean and three-sigma deviation for multitask learning with mixed effects

APPENDIX F: TURBOCHARGER MODEL: VARIANCE REDUCTION PLOTS

Figure F.1

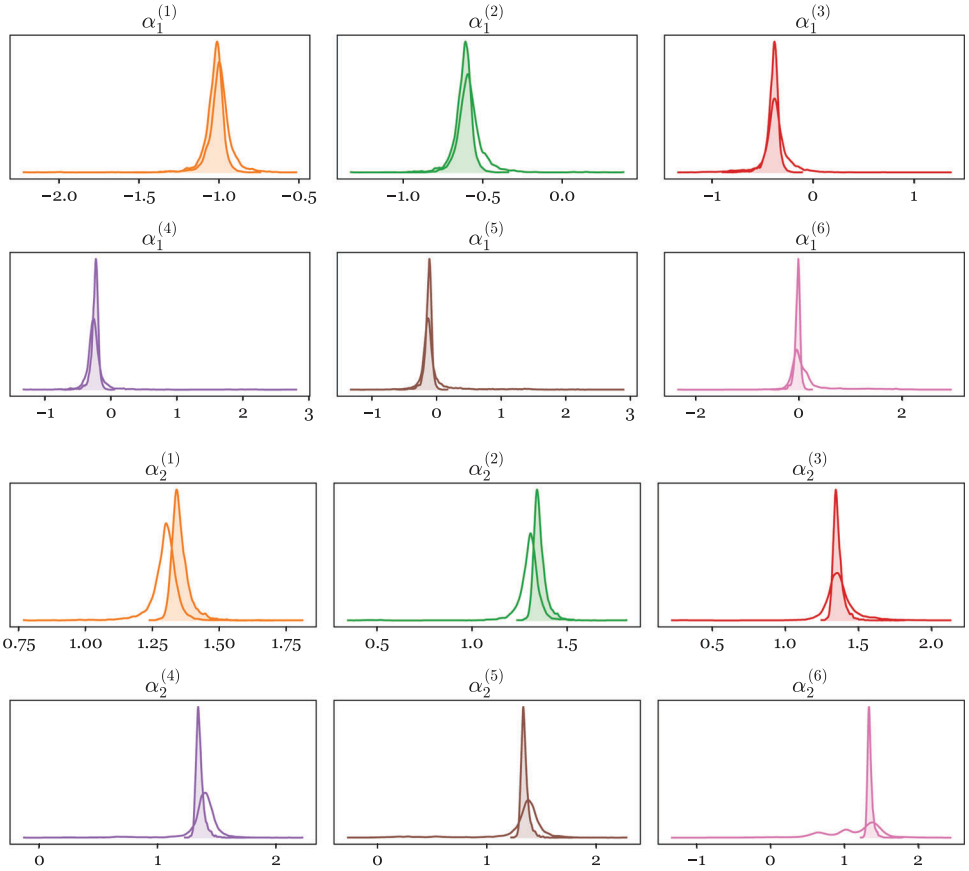

FIGURE F.1 Variance reduction in the posterior distribution of the intercept $\alpha_1^{(k)}$ and slope $\alpha_2^{(k)}$ parameters for turbocharger components. Independent models (hollow)/population level modeling (shaded)