Bayesian Expandable Ambient Magnetic Mapping: Continual Learning with a Physics-Constrained Bayesian Neural Network

Anonymous Authors

***Abstract*— Leveraging anomalies and features in magnetic fields offers a passive, high-precision methodology for GNSS-denied navigation in contested environments requiring accurate prior maps. Mapping magnetic fields and their anomalies is challenging to achieve with precision. We introduce *Bayesian expandable ambient magnetic mapping*, a physics-constrained Bayesian neural network framework for modeling and interpolation of indoor magnetic fields. By explicitly embedding Maxwell's equations as a hard constraint into the network and deploying a hybrid prior that leverages Kalman-style updating and *maximum a posteriori* strategies, the presented method supports continual learning while sacrificing minimal variance and incurring only a fraction of the computational cost of traditional BNN sampling approaches. The framework is validated for accuracy using simulated datasets, and physically validated on a ground robot equipped with quantum magnetometers for high-precision mapping. Code, videos, and datasets are available at `https://beamm-map.github.io`.**

I. INTRODUCTION

Ensuring accurate and reliable localization for autonomous vehicles remains a central challenge for robotic navigation. Although the widely used Global Navigation Satellite System (GNSS) offers sub-meter accuracy for outdoor positioning, its reliability diminishes or becomes inapplicable in certain scenarios, such as indoor navigation or during extended operations in large aerial or underwater spaces. Common workarounds for GNSS signal degradation, such as inertial sensor fusion, suffer from accumulated drift. At the same time, visual data- or feature-based localization methods, such as Simultaneous Localization and Mapping (SLAM), have been successful in indoor navigation but face limitations when distinctive landmarks are scarce or change dynamically [1].

Magnetic-based navigation (MagNav) offers a favorable solution by exploiting the ubiquitous Earth magnetic field, which is globally available, infrastructure-independent, and resistant to spoofing [2]. Magnetic-based localization leverages the spatial frequency features of the given magnetic field map to estimate position, as demonstrated in both ground-based [3], [4], aerial [5], [6], and water platforms[7], [8].

Mapping in magnetic navigation is not a new problem. The first large-scale maps were generated through satellite magnetometry [9]. With the advent of small indoor robots, localization using ambient magnetic fields garnered interest in the robotics community [10], a task that requires accurate prior surveys. [11] found that while there is substantial spatial variation in magnetic fields, there is minimal temporal variation in an interference-free field, with [12] reinforcing this point by presenting high-resolution magnetic field components. [13] completed a grid-based method for collecting prior magnetic intensity maps and showed that high-resolution localization is significantly enhanced. The methodology is still used in more recent work, albeit via tedious grid-based formulations with interpolation [14], [15]. While this is effective for deployment in localization systems, an unstructured, organic formulation would yield faster mapping and higher-fidelity maps.

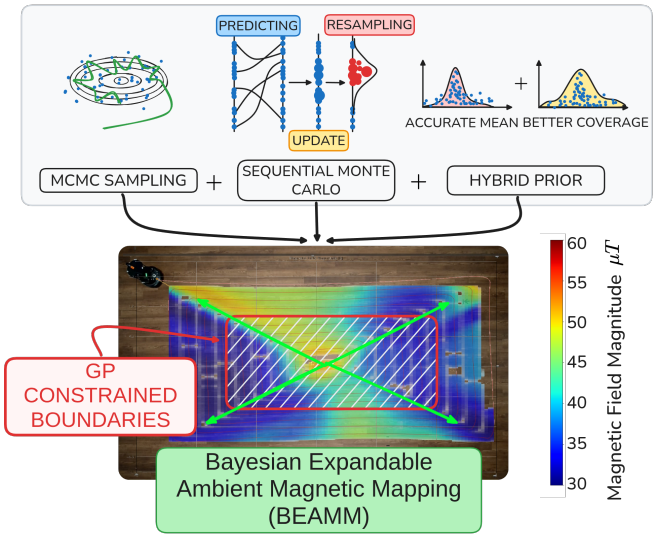


Fig. 1: BEAMM combines Monte Carlo posterior sampling for network parameters through an informed hybrid before enabling fast posterior exploration. Additionally, a sequential Monte Carlo filtering scheme is implemented for continual learning to mitigate catastrophic forgetting. The combined architecture provides an expandable domain-mapping implementation for ambient magnetic fields.

An unstructured formulation for mapping is not new for other sensing modalities. While unstructured SLAM approaches [16], [17] have matured for other modalities, magnetic-field SLAM remains limited by the low spatial resolution of ambient fields [18], with aerial implementations achieving only ≈ 11 m root-mean square error [19].

Recently, *Gaussian Process*-based approaches that leverage magnetic-field physics via Maxwell's equations have been developed to accurately and physically constrain magnetic maps via sparse Hilbert-space approximations [20], [21]. These are landmark methodologies for effectively predicting and interpolating the non-linear field with calibrated uncertainty estimates. However, the methods presented in [20] and [21] are limited to small, fixed domains due to their boundary conditions and suffer from scalability issues due to a limited set of basis functions. [22] improved upon this method by introducing a distributed sparse GP formulation

that uses a probabilistic smoothing framework to ensure that domains between trained instances are sufficiently smooth. Similarly, [23] used kernel interpolation to scale up GP regression and continuously map growing areas. However, both are still limited to *a priori* fixed domains and cannot have continuously expanded maps without a new set of basis functions.

To overcome domain-size and retraining constraints while preserving uncertainty estimates, we propose **B**ayesian **E**xpandable **A**mbient **M**agnetic **M**apping (BEAMM), a physics-constrained Bayesian neural network (BNN) that enforces Maxwell's equations and continually learns a magnetic field map without a fixed set of basis functions. An architectural overview for BEAMM is given in Fig. 1.

A. Contributions

The following are the main contributions that are presented here:

1) **Expandable Bayesian magnetic mapping**: We propose a Bayesian magnetic mapping framework that supports continual domain growth without basis-function reinitialization, addressing a core limitation of fixed-domain GP methods.
2) **Hard physics-structured representation**: We introduce a scalar-potential BNN formulation that enforces curl-free structure by construction rather than through soft constraints.
3) **Online robotic deployment**: We develop a practical Bayesian inference and continual learning strategy and validate it on a live ground robot equipped with a hybrid quantum magnetometer, with code and datasets released publicly.

II. RELATED WORK

Uncertainty Aware Representations: While GPs provide well-calibrated uncertainty, their scalability is limited by the fixed basis-function set required for sparse Hilbert-space approximations [20]. [22] addressed this by distributing the regression across multiple gradient-deterministic conditional training (G-DTC) experts, aggregated via a local Bayesian committee machine (LBCM), thereby enabling large-domain mapping. However, DTC approximations tend to underestimate uncertainty, and LBCM aggregation can yield overconfident predictions, both of which lead to poorly calibrated uncertainty estimates compared with a full GP posterior. Parallel efforts have explored neural-network alternatives to GPs for magnetic-field modeling. In [24], physics-constrained GANs were employed to interpolate and extrapolate magnetic fields, while [25] used generative inpainting via the DeepFill model to bypass the computational cost of GP inference.

Physics Constrained Learning: Several extensions to physics-informed neural networks (PINN) have been proposed to address this uncertainty gap. Epistemic PINNs (E-PINNs) augment the base PINN with an auxiliary network that estimates the epistemic uncertainty associated with each prediction [26], providing a practical route to uncertainty quantification without abandoning the soft-constraint training paradigm. Bayesian PINNs (B-PINNs) take a more principled approach by placing a posterior distribution directly over the network weights [27], much like a traditional BNN, yielding well-characterized uncertainty estimates. However, because B-PINNs still encode physics as soft loss-based constraints rather than hard structural constraints, they cannot guarantee physical consistency across the learned field.

Fast Uncertainty Aware Learning: Approximate Bayesian inference methods offer an intermediate between full GP posteriors and deterministic neural networks. Gal and Ghahramani demonstrated that MC Dropout approximates a deep GP posterior at inference time through stochastic dropping of neurons, providing a computationally efficient uncertainty estimate without explicit weight distributions [28]. [29] introduced Probabilistic Backpropagation (PBP), which propagates uncertainty through network layers by maintaining a Gaussian approximation over weights, yielding better-calibrated estimates than MC Dropout. However, PBP's mean-field assumption underestimates weight correlations, degrading predictive variance in spatially structured fields.

III. PROBLEM FORMULATION

A. Ambient Magnetic Field

First, the magnetic vector field is described as a function $\mathbf{B}(\mathbf{x}) : \mathbb{R}^3 \to \mathbb{R}^3$ where $\mathbf{x} \in \mathbb{R}^3$ is the point of interest in space and $\mathbf{B}(\mathbf{x})$ denotes the three-dimensional magnetic field vector at location $\mathbf{x}$. The field is described as curl-free if it follows the fundamental relation in vector fields:

$$\nabla \times \mathbf{B} = \mathbf{0} \tag{1}$$

The field $\mathbf{B}$ can be alternatively described as a scalar potential where the gradients give us the expanded field

$$\mathbf{B} = -\nabla\varphi(\mathbf{x}), \quad \varphi : \mathbb{R}^3 \to \mathbb{R} \tag{2}$$

B. Workspace and Map

The workspace is defined as $\mathcal{W} \subset \mathbb{R}^3$. The map built is an interpolated field map within the bounds of the collected data $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^N$, where $\mathbf{y}_k$ is the sensed magnetic field at $\mathbf{x}_k$ and N is the number of data points. However, it is noted that the mobile robot's motion is restricted to 2D because a ground robot performs the mapping.

We exploit (1) and (2) for magnetic fields; the scalar potential is inferred instead of predicting the three individual fields:

$$\begin{aligned} \varphi &= f(\mathbf{x}) \\ \mathbf{B} &= -\nabla f(\mathbf{x}) \end{aligned} \tag{3}$$

where $f(\cdot)$ is the learned functional predictor.

Problem statement: Given a sparse set of magnetometer measurements $\mathcal{D} = \{(\mathbf{x}_k, \mathbf{y}_k)\}_{k=1}^N$ collected over a workspace $\mathcal{W} \subset \mathbb{R}^3$, the goal is to learn a continuous functional predictor $f : \mathbb{R}^3 \to \mathbb{R}$ that infers the scalar magnetic potential φ at any query location $\mathbf{x}^* \in \mathcal{W}$, such that the derived field $\mathbf{B} = -\nabla\varphi(\mathbf{x})$ satisfies the physical constraints in (1) and (2) and approximates $\mathbf{y}^* \approx \mathbf{B}(\mathbf{x}^*)$.

Rather than producing a point estimate, f is parameterized as a Bayesian Neural Network (BNN) that maintains a posterior distribution $p(\varphi(\mathbf{x}^*)|\mathcal{D})$ that induces a posterior over the magnetic field $\mathbf{B}(\mathbf{x}^*) = -\nabla_{\mathbf{x}^*}\varphi(\mathbf{x}^*)$, yielding a mean estimate $\mathbb{E}[\mathbf{B}(\mathbf{x}^*)]$ and uncertainty $\mathrm{Var}[\mathbf{B}(\mathbf{x}^*)]$ at each query point. The formulation must remain valid under four conditions: (i) **domain expansion**, where $\mathcal{W}$ grows as the platform explores previously unmapped regions; (ii) **continuous updating**, where the posterior is revised sequentially as new measurements arrive through particle reweighting and resampling, avoiding cold-start retraining; (iii) **quantified uncertainty**, where calibrated predictive uncertainty is produced by the functional predictor and validated under controlled simulation conditions; and (iv) **predictive performance with minimal data**, where the posterior predictive performance requires minimal data.

IV. Bayesian Expandable Ambient Magnetic Mapping (BEAMM)

Bayesian expandable ambient magnetic mapping (BEAMM) utilizes a physics-constrained Bayesian neural network to model the ambient magnetic field, with deployable inference enhanced by Hamiltonian Monte Carlo, Kalman-Bayesian covariance approximation, and sequential particle resampling.

A. Bayesian Neural Network (BNN)

A BNN [30], [31] places a prior $p(\theta)$ over network weights θ, yielding a posterior $p(\theta \mid \mathcal{D})$ and predictive distribution $p(\mathbf{y}^* \mid \mathbf{x}^*, \mathcal{D})$ via (4)–(7):

$$\theta \sim p(\theta) \tag{4}$$

$$y = \Phi_\theta(x) + \epsilon \tag{5}$$

$$p(\theta \mid \mathcal{D}) = \frac{p(Y|X,\theta)p(\theta)}{p(Y \mid X)} \tag{6}$$

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int_\theta p(\mathbf{y}^* \mid \mathbf{x}^*, \theta)p(\theta \mid \mathcal{D})d\theta \tag{7}$$

Both the posterior and predictive distributions are high-dimensional and non-convex (multi-modal), which are intractable in closed form [32], [33].

B. Physics Constrained Bayesian Neural Network

Rather than imposing physics as a soft constraint [34], [35], BEAMM parametrizes the field in terms of a scalar potential, enforcing curl-free constraints as a hard structural constraint. Following the definitions in Sec. III-A, we model the three vector fields as a scalar potential φ, forcing the functional predictor to predict curl-free fields, defining a modified predictive distribution as:

$$p(\varphi^*|\mathbf{x}^*, \mathcal{D}) = \int_\theta p(\varphi^* \mid \mathbf{x}^*, \theta)p(\theta \mid \mathcal{D})d\theta \tag{8}$$

The gradient of the output φ^* will give the recovered three-dimensional field:

$$\mathbf{B}(\mathbf{x}^*) = -\nabla_{\mathbf{x}^*}\varphi(\mathbf{x}^*) \tag{9}$$

For the specific activation of each neuron, a *Softplus* activation was used over a traditional *ReLU* (rectified linear unit) due to the requirement of differentiability of φ to recover the magnetic field $\mathbf{B}$, making *Softplus* the natural choice due to the boundedness, smoothness, and differentiability of the function.

C. Predictive and Posterior Approximation

We use a Hamiltonian Monte Carlo (HMC) sampler [32] to draw samples $s \in \{1, 2, \ldots, S\}$ from the target posterior distribution $p(\theta \mid \mathcal{D})$ by augmenting the parameter space with momentum variables $\mathbf{r} \sim \mathcal{N}(0, \mathbf{M})$ where $\mathbf{M}$ is a mass matrix. The Hamiltonian dynamics guide the system:

$$H(\theta, \mathbf{r}) = U(\theta) + K(\mathbf{r}) \tag{10}$$

where $U(\theta)$ is the negative log posterior $\mathcal{L}(\theta) = -\log p(\mathcal{D} \mid \theta) - \log p(\theta)$, which serves as the potential energy term in the Hamiltonian system, and $K(\mathbf{r})$ is the kinetic term. The continuous sampler evolves constrained by Hamilton's equations, which are numerically estimated through the leapfrog integrator:

$$\begin{aligned} \mathbf{r}_{t+\frac{1}{2}} &= \mathbf{r}_t - \frac{\tau}{2}\nabla_\theta U(\theta_t), \\ \theta_{t+1} &= \theta_t + \tau \mathbf{M}^{-1}\mathbf{r}_{t+\frac{1}{2}}, \\ \mathbf{r}_{t+1} &= \mathbf{r}_{t+\frac{1}{2}} - \frac{\tau}{2}\nabla_\theta U(\theta_{t+1}) \end{aligned} \tag{11}$$

where τ is the step size.

The no-U-turn sampler (NUTS) [36] employs dual averaging to adapt the leapfrog step size τ automatically, and terminates trajectory integration to adaptively set trajectory length, eliminating the need to tune parameters manually.

$$(\theta^+ - \theta^-) \cdot \mathbf{r}^- < 0 \quad \text{or} \quad (\theta^+ - \theta^-) \cdot \mathbf{r}^+ < 0 \tag{12}$$

where θ^+ and θ^- are the forward and backward trajectory endpoints, respectively.

Although NUTS provides well-calibrated samples, its per-step cost motivates a hybrid strategy that is discussed next.

D. Efficient Training

As retaining calibration quality while reducing sampling cost is paramount, we combine a Kalman-style covariance approximation with an accurate *maximum a posteriori* (MAP) estimate to form an informed prior. [37] introduced a Kalman-style system for updating the weight posterior (Kalman-BNN), providing an efficient computational framework for single-epoch continual learning.

1) Kalman Bayesian Neural Network: The training system utilizes the Kalman-BNN strictly as a covariance initializer; it is insufficient as a standalone posterior approximation. The Kalman-BNN applies Bayes' rule (6) layer-wise via a joint Gaussian filtering formulation for any input-output pair $(\mathbf{x}, \mathbf{y})$, yielding the posterior in (13).

The updated weights from the joint Gaussian under the Kalman smoothing step are then used to draw samples from the posterior distribution:

$$p(\theta \mid \mathcal{D}) \sim \mathcal{N}(\mu_w^+, \mathbf{C}_w^+) \tag{13}$$

where $+$ indicates an updated distribution, μ_w is the mean, and $\mathbf{C}_w$ is the covariance of the posterior.

As the Kalman-BNN update is expressed in terms of scalar-potential targets rather than vector-field components, relative scalar-potential targets can be recovered from a reference point $\mathbf{x}_0$ via the gradient theorem:

$$\varphi(\mathbf{x}) - \varphi(\mathbf{x}_0) = -\int_{\mathbf{x_0}}^{\mathbf{x}} \mathbf{B}(\mathbf{s})\ d\mathbf{s} \tag{14}$$

following the curl-free assumption in (1).

2) Informed Prior via Stochastic Gradient Descent and Kalman Smoothing: While the Kalman-BNN provides an efficient single-epoch approximation for efficient learning, its posterior mean is often too inaccurate to serve directly as an informed NUTS prior [37]. Conversely, a stochastic-gradient MAP estimate provides a higher-quality posterior mode, but collapses uncertainty around that mode, yielding an overconfident initialization [38]. To resolve this, we construct an informed hybrid Gaussian prior whose mean is the MAP estimate and whose covariance is the diagonal Kalman-BNN posterior variance:

$$p(\theta) = \mathcal{N}(\boldsymbol{\mu}_{\mathrm{MAP}}, \mathbf{C}_w^+) \tag{15}$$

where $\boldsymbol{\mu}_{\mathrm{MAP}} = \arg\max_\theta \left[\log p(\mathcal{D} \mid \theta) + \log p(\theta)\right]$ and $\mathbf{C}_w^+$ is the diagonal posterior defined in (13). If the diagonal entries of $\mathbf{C}_w^+$ are strictly positive, then (15) is a valid Gaussian prior over θ. The prior centers NUTS in a high-probability region while preserving the per-parameter uncertainty, reducing burn-in, and improving posterior exploration. A diagonal matrix with strictly positive entries $\sigma_{w_i}^2 > 0$ is symmetric positive definite by construction, guaranteeing a proper Gaussian density.

The hybrid prior in (15) constitutes an empirical Bayes construction, as the prior components are derived from the observed dataset $\mathcal{D}$, leading to a mildly overconfident uncertainty estimate in data-rich regions, while the primary benefits of reducing NUTS burn-in and improved posterior mode localization remain valid. This overconfidence is directly reflected in the coverage results shown in Sec. VI-A, where a coverage gap is attributable to the stated overconfidence. In practice, the diagonal structure of $\mathbf{C}_w^+$ also provides a convenient diagonal prior covariance for the NUTS integrator.

E. Continual Learning

Without an explicit prior over the parameter distribution, batched updates cause catastrophic forgetting [39], [40]; moreover, storing all accumulated data scales as $\mathcal{O}(ND)$ [31] where D is the dimensionality of observations. The HMC posterior samples are treated as weighted particles representing the current approximation to $p(\theta \mid \mathcal{D}_{1:t})$. As new data batches arrive, these particles are sequentially reweighted using the likelihood of the new batch, and resampled when a particle-degeneracy threshold is reached. This method is called a sequential Monte Carlo (SMC) [41], [42], [43].

Algorithm 1 Algorithm for continual learning using the hybrid prior NUTS setup

Input: $\mathcal{D}_t = \{\mathbf{x}_t, \mathbf{y}_t\}, \mathbf{x}^*$
Output: $\mathbb{E}(\mathbf{B}_t)$, $\mathrm{Var}(\mathbf{B}_t)$

```
Initialize θ samples using (15) and NUTS (11).
Use (7) to predict over x* to get y* distribution.
Use (9) to recover the ambient field B.
Sequentially update the posterior sample set:
for t = 2, 3 ..., T do
    Using (13) and MAP estimation update the hybrid prior (15).
    Using U(θ) in NUTS, explore through (11) for D_t to find θ_t samples.
    Re-weight the posterior samples using (16).
    if ESS < S/2 then
        Resample the weights using residual sampling [44]
    Use the (7) to predict over x* to get y* distribution.
    Use (9) to recover the ambient field B.
```

Let $p(\theta \mid \mathcal{D}_{1:t})$ be the posterior over network parameters after observing batches $\mathcal{D}_{1:t}$, approximated by the weighted particle set $\left\{\theta^{(i)}, w_t^{(i)}\right\}_{i=1}^{S}$. For a new batch of data $\mathcal{D}_{t+1}$, the updated posterior $p(\theta \mid \mathcal{D}_{1:t+1})$ is approximated by reweighting:

$$w_{t+1}^{(i)} \propto w_t^{(i)} p(\mathcal{D}_{t+1} \mid \theta^{(i)}) \tag{16}$$

In practice, particle degeneracy is monitored by the effective sample size

$$\mathrm{ESS} = \frac{(\sum_i w_i)^2}{\sum_i w_i^2} \tag{17}$$

and residual resampling [44] is triggered when $\mathrm{ESS} < \frac{S}{2}$. An overview of the entire system is given in Algorithm 1.

V. EXPERIMENTAL SETUP

The simulated experiments consist of running ablation and cold-start behavior studies on a curl-free dataset generated using the architecture described by [20] and direct calibration trends using datasets generated by [22] with two state-of-the-art physics-constrained algorithms *G-Hilbert* [20], and *G-DTC* [22]. The hardware experiments to validate deployability and expandable mapping were conducted on a ground robot equipped with a QUSpin QTFM magnetometer, which is a hybrid scalar-vector optically pumped magnetometer with a 140 Hz sampling rate and a sensitivity of $< 0.1\ \mathrm{nT}/\sqrt{\mathrm{Hz}}$.

BEAMM's tunable network parameters are given in Table I and are consistent across all experiments. 50 samples were quantitatively selected via simulated ablation studies to optimize efficiency and posterior quality. Performance is measured by mean squared error (MSE) and root-mean squared error (RMSE). Calibration is evaluated using mean standardized log loss (MSLL) and $1\sigma/2\sigma$ coverage.

TABLE I: BEAMM network parameters

Parameter	Value
Hidden Neurons	128
Number of NUTS Samples	50
Initial NUTS Step Size (τ)	3×10^{-6}
NUTS exploration steps per sample	30
Number of Hidden Layers	2
Activation Function	*Softplus*

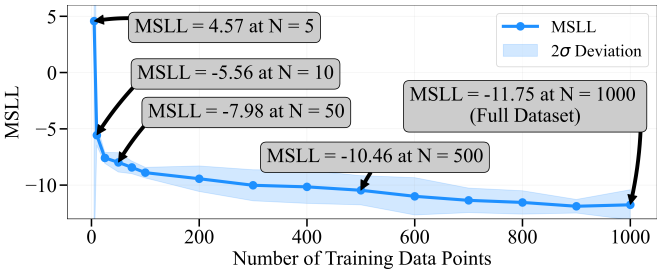


Fig. 2: Predictive performance on a subset of the training points $N \subset \mathbb{N}$ is shown as the change in *mean standardized log loss* (MSLL) as training data points increase, with points from the upper graphs highlighted with callouts.

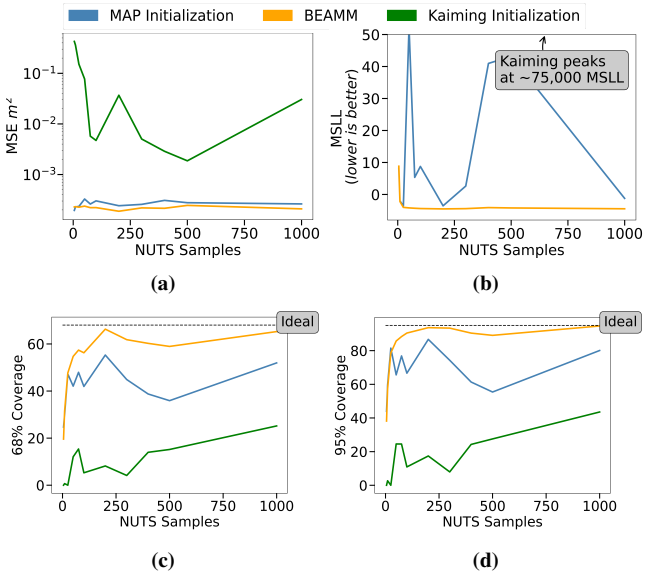


Fig. 3: Ablative study using three different priors: 1) Kaiming initialization [47], 2) Stochastic gradient descent, and 3) BEAMM. (a) The mean squared error (MSE), (b) mean standardized log loss (MSLL), (c) the 1σ standard deviation coverage (68%), and (d) 2σ (95%) standard deviation coverage.

A. Simulation Setup

For ablative studies to validate each part of the architecture, a simulated dataset was generated using the G-Hilbert architecture with the following parameters: 1) $\sigma^2_{\text{noise}} = 0.01$, 2) $\sigma^2_{\text{SE}} = 1.0$, 3) $l_{\text{SE}} = 1.0$, 4) $\sigma^2_{\text{const}} = 0.3$, and 5) $m = 4096$. For comparative performance evaluation of the algorithms, [22] provided three simulated datasets: *Simu1D*, *Simu2D*, and *Simu3D*. These datasets are generated through the use of a physics-constrained (see Sec. III-A) zero-mean Gaussian process in a $3 \times 3 \times 3$ m^3 box with each dataset consisting of 1000 noisy training data points, and a shared 100 point test data set. As the names suggest, each dataset is constrained to 1D,2D, and 3D, respectively, providing varied coverage for the network to train on.

B. Hardware

For hardware experiments, the mapping and interpolation system was run on a Clearpath TurtleBot4 with ROS 2 [45] in a lab outfitted with motion-capture cameras to provide accurate position and rotational measurements. To account for computational requirements, such as the number of CUDA cores for Torch-based inference, an NVIDIA Jetson Orin was connected to the robot's hardware to run neural inference more efficiently.

The robot was programmed to drive along a square spiral path that extended in both x, y by a factor of 0.15 m. A Stanley controller was used to follow the path and maintain good tracking [46].

VI. RESULTS AND DISCUSSION

A. Simulation

The comparative performance between three GP-simulated datasets designed by [22] is given in Table II. The proposed BEAMM architecture performs similarly to G-DTC and G-Hilbert. Across datasets, BEAMM performs reasonably competitively on MSE and is around 1.9 times worse on the *Simu3D* dataset and MSLL. Despite evaluation on a GP-generated dataset that inherently favors the baselines, BEAMM's weight generalization enables deployment at locations that deviate significantly from training points.

TABLE II: Comparative results for BEAMM against G-DTC and G-Hilbert on simulated datasets.

Dataset	Method	**MSE** (μT^2)	**MSLL**	**Prediction Time** (ms)
Simu1D	G-DTC	7.7×10^{-5}	-11.8	2.2
	G-Hilbert	7.8×10^{-5}	-12.3	28
	BEAMM	$\mathbf{1.84 \times 10^{-4}}$	$\mathbf{-11.5}$	**8.2**
Simu2D	G-DTC	1.9×10^{-4}	-11.6	8.7
	G-Hilbert	1.9×10^{-4}	-11.7	28
	BEAMM	$\mathbf{3.8 \times 10^{-4}}$	$\mathbf{-10.2}$	**7.3**
Simu3D	G-DTC	4.9×10^{-4}	-11.1	43
	G-Hilbert	5.1×10^{-4}	-10.9	28
	BEAMM	$\mathbf{9.5 \times 10^{-4}}$	$\mathbf{-9.7}$	**11.2**

Table III ablates the physics constraint described in Sec. IV-B and the hybrid prior introduced in Sec. IV-D.2. Noting PBP's 100% coverage across both σ bands reflects severe underconfidence attributable to the breakdown of its mean-field Gaussian approximation under the correlated weight structure, which is consistent with known limitations of analytical propagation through non-linear activations [29].

In Fig. 2, the methodology can cold-start with minimal training data, with the MSLL dropping from 4.57 at $N = 5$ to -5.56 at $N = 10$, representing a significant improvement in both MSE and variance. Fig. 3 shows that BEAMM converges $50 - 100$ samples faster than the baseline Adam and random (Kaiming [47]) initializations on both number

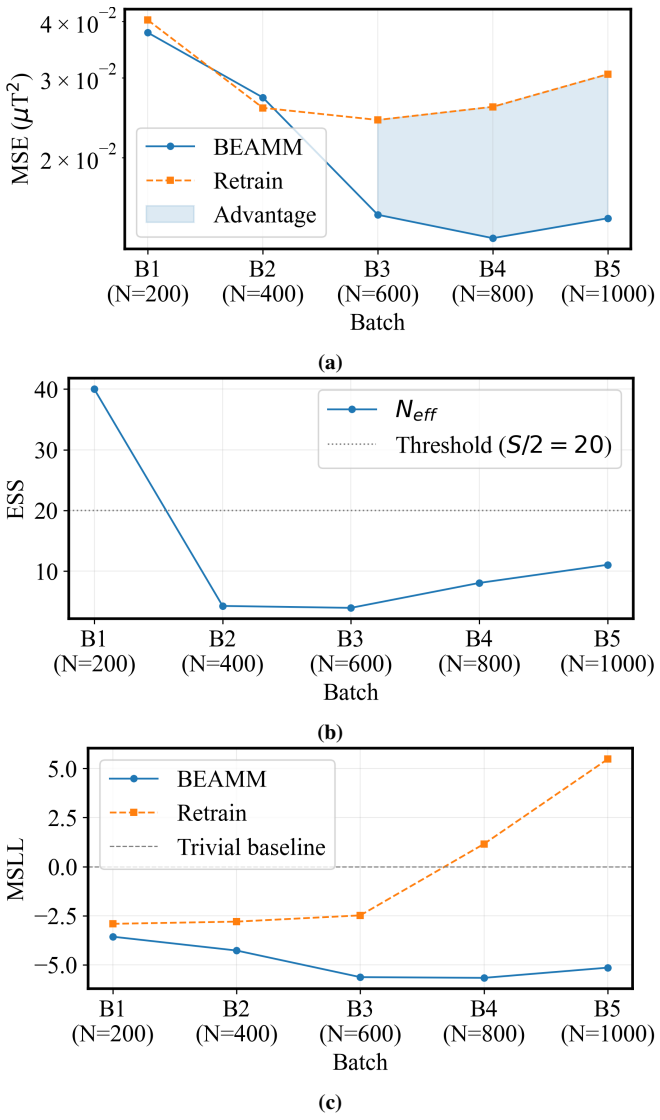


Fig. 4: Ablation study on the SMC posterior consistency. A dataset was split into 5 batches and sequentially run through BEAMM, reporting: (a) MSE, (b) ESS, and (c) MSLL.

of samples versus MSE and MSLL, and 68% and 95% coverage, respectively.

Finally, as a validation of SMC consistency, an ablation study using a batched split of the dataset generated using G-Hilbert is shown in Fig. 4 with BEAMM outperforming retrain. BEAMM maintains a fully weighted posterior set, resampling only when the ESS threshold is met, whereas retrain discards all prior posterior information on each batch. BEAMM resampling is triggered on each batch because each new batch exposes a new area within the bounded simulated cube, necessitating a sample refresh. Overall, BEAMM outperforms the retrain baseline in both accuracy and calibration, converging to the MSLL given in Table III.

B. Hardware

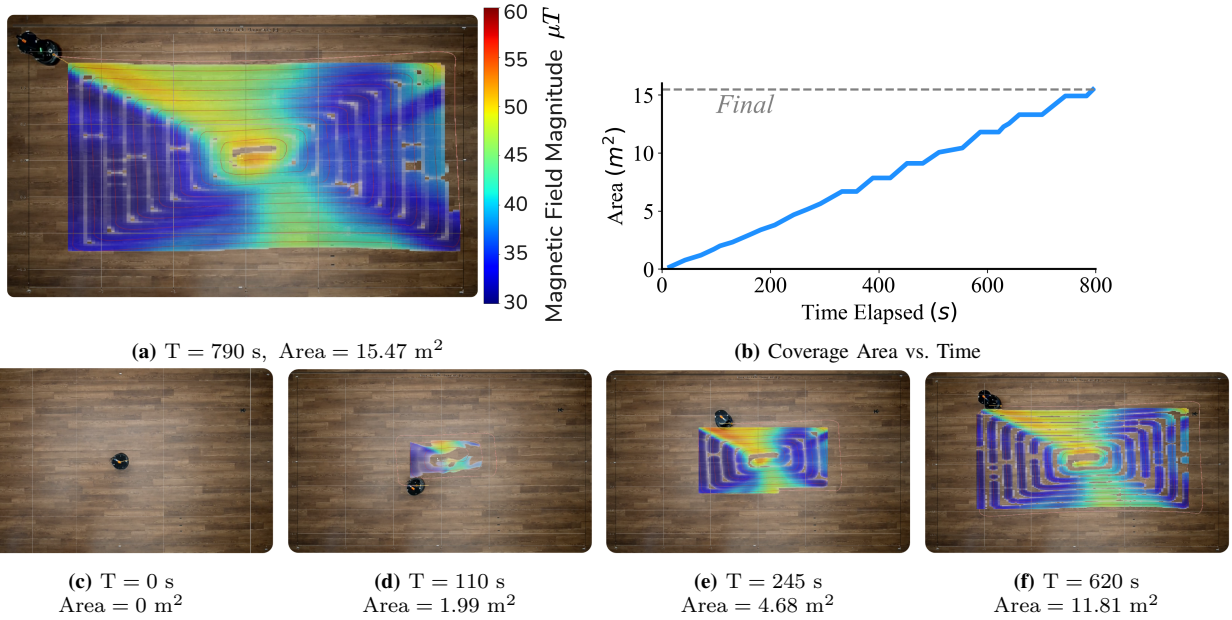
The hardware experiment demonstrates online operational capability and mapping accuracy. Uncertainty calibration is validated extensively in simulation (Tables II and III), but

TABLE III: Component-wise ablation on simulated data. $+\varphi$: curl-free scalar potential constraint (Sec. IV-B). +MAP: informed prior via MAP. Separate BNN is a BNN that predicts three fields instead of the scalar potential.

Variant	**MSE** (μT^2)	**MSLL**	**68%**	**95%**
Uninformed Priors				
Separate BNN	5.59×10^{-2}	22.95	25.9	47.7
BNN $+\varphi$	5.10×10^{-2}	-0.23	39.0	67.6
PBP [29]	9.93×10^{-2}	21.69	100	100
MC Dropout [28]	1.71×10^{-1}	-1.34	54.6	83.3
MAP-Initialized Priors				
Separate BNN +MAP	3.68×10^{-2}	-1.76	39.4	68.3
BNN $+\varphi$ +MAP	1.65×10^{-2}	-3.69	44.6	75.4
BEAMM	$\mathbf{1.17 \times 10^{-2}}$	$\mathbf{-5.75}$	**60.8**	**87.8**

is not assessed here due to insufficient spatial density of the ground-truth survey for a statistically representative coverage evaluation. A 375-point dense grid survey (3.75% of the 10000-point prediction grid) provides ground truth for error calculation for the map shown in Fig. 5. Using the training data collected from the hardware experiment, G-Hilbert and G-DTC were run on the system to evaluate its performance relative to BEAMM. While G-Hilbert and G-DTC cannot be applied to systems with changing domains, they provide a reliable baseline that can be trained on and used to predict on the entire training set. The results are shown in Table IV. G-Hilbert and G-DTC are run offline after the fact, whereas BEAMM was run continuously on hardware. The comparison is inherently asymmetric, as a strictly fair online comparison would require refitting the GP basis functions, which the methods do not support by design.

Observing Table IV, despite operating online without prior domain knowledge, BEAMM achieves total field RMSE within 0.6 μT of G-Hilbert, which operates offline with full trajectory access. BEAMM outperforms G-DTC across most hardware metrics. When examining the individual fields, BEAMM tends to perform favorably in B_y and B_z, with far superior performance in B_z and comparable performance in B_y. BEAMM's online performance B_x is by far the worst among the other algorithms; however, it is comparable to G-Hilbert in the offline version. The performance of B_z and B_y yields a practical total-field RMSE. The anomalously large B_z errors for G-Hilbert and G-DTC are attributed to the robot's 2D motion constraining vertical-field sampling, resulting in poor GP conditioning for the B_z component. Interestingly, offline BEAMM also performs poorly on B_z. This issue is attributed to the incremental prediction grid used in the method. At each SMC step, it only covers the local region where the density of vertical-field sampling is highest. In contrast, batched grid systems evaluate the entire region. BEAMM's significantly superior Bz performance versus G-Hilbert, offline BEAMM and G-DTC (RMSE 1.93 μT versus 72.23 μT, 62.94 μT and 68.16 μT respectively) demonstrates the advantage of the BNN's weight generalization and SMC updating over the fixed-basis-function approach in undersampled dimensions.



(a) T = 790 s, Area = 15.47 m^2

(b) Coverage Area vs. Time

(c) T = 0 s
Area = 0 m^2

(d) T = 110 s
Area = 1.99 m^2

(e) T = 245 s
Area = 4.68 m^2

(f) T = 620 s
Area = 11.81 m^2

Fig. 5: BEAMM magnetic field interpolation over 790 seconds. (a) Final map with field magnitude in μT. (b) Bounding box coverage vs. time, computed via $(x_{\max} - x_{\min}) \times (y_{\max} - y_{\min})$ demonstrating continuous domain expansion. (c) — (f) Sequential snapshots at $T = 0,\ 110,\ 245,$ and 620 s. Transparency encodes predictive uncertainty. G-Hilbert and G-DTC require prior fixed domain coverage and cannot be evaluated incrementally.

BEAMM's 24.39 s per update latency sits comfortably between G-Hilbert's (3.01 s due to the Hilbert space approximation) and BEAMM (offline at 3.32 s), and G-DTC (161.02 s, limited by Local Bayesian Committee Machine cost). As mapping operates independently of navigation, BEAMM per-update latency is consistent with the robot's spiral mapping trajectory.

Figure 5 shows the coverage area, computed as $(x_{\max} - x_{\min}) \times (y_{\max} - y_{\min})$, versus time, growing monotonically from 0 m^2 to 15.47 m^2. The coverage bounding-box metric is a conservative upper bound derived from the network's predicted positional inputs. This continuous domain-expansion capability directly addresses the fixed-domain limitations of GP-based methods [20], [22], enabling online deployment without reinitialization of basis functions.

VII. CONCLUSIONS

BEAMM demonstrated five experimentally validated contributions: continuous domain expansion without basis function reinitialization; hard-physics constraint validation; deployability through live robot mapping; efficient posterior exploration via an informed hybrid prior; and cold-start capability from minimal initial observations. The findings indicate that BEAMM offers a scalable and practical framework for magnetic mapping in cases where conventional GP-based methods are challenging.

A. Limitations

Despite its competitive performance, BEAMM inherits several limitations from Bayesian neural network approaches. First, the learned representations and network parameters remain difficult to interpret, making model debugging and hyperparameter tuning time-consuming and more difficult than in kernel-based methods. Second, uncertainty estimation requires a Monte Carlo-based sampler, which introduces additional computational overhead that may be prohibitive at high update rates or on resource-constrained platforms, as evidenced by the per-update latency in Table IV. This latency is suitable for offline concurrent map building, but would require further optimization for deployment in real-time SLAM pipelines. Finally, as with other data-driven approaches, performance depends on the representativeness of the training data as evidenced by Table IV. Consequently, calibrated uncertainty quantification is validated in simula-

TABLE IV: Hardware results for BEAMM, G-Hilbert, and G-DTC compared against dense grid-survey ground truth. G-Hilbert and G-DTC are evaluated offline; BEAMM is evaluated online. Predict time for BEAMM is reported per update step of the SMC.

Method	Vector	**MSE** (μT^2)	**RMSE** (μT)	**Online**	**Per Update Latency** (s)
G-Hilbert	B_x	265.93	16.3	×	3.01
	B_y	298.17	17.27		
	B_z	5217.26	72.23		
	B	24.19	4.92		
G-DTC	B_x	233.63	15.29	×	161.02
	B_y	124.81	11.17		
	B_z	4645.17	68.16		
	B	95.98	9.80		
BEAMM (Offline)	B_x	225.49	15.01	×	3.32
	B_y	364.04	19.08		
	B_z	3961.45	62.94		
	B	24.50	4.95		
BEAMM	B_x	537.92	23.19	✓	24.39 per step
	B_y	366.73	19.15		
	B_z	3.73	1.93		
	B	30.18	5.49		

tion only (Tables II and III); hardware uncertainty calibration remains an open item for future work. Future work will investigate computationally efficient approximate inference methods, including closed-form Bayesian updates as shown in GPs. Additionally, extension to 3D aerial platforms and investigation of sensor fusion with IMU data represent promising directions for improving real-world deployment performance.

REFERENCES

[1] J. Huang *et al.*, "Indoor Positioning Systems of Mobile Robots: A Review," *Robotics*, vol. 12, no. 2, p. 47, Apr. 2023.

[2] A. Canciani and J. Raquet, "Absolute Positioning Using the Earth's Magnetic Anomaly Field," *Navigation*, vol. 63, no. 2, pp. 111–126, 2016.

[3] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in *Proc. Ubiquitous Position. Indoor Navig. Location Based Serv.*, Oct. 2010, pp. 1–10.

[4] J. Shockley, "Ground Vehicle Navigation Using Magnetic Field Variation," Ph.D. dissertation, Air Force Institute of Technology, Sep. 2012.

[5] A. Canciani and J. Raquet, "Airborne Magnetic Anomaly Navigation," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 67–80, Feb. 2017.

[6] A. J. Canciani, "Magnetic Navigation on an F-16 Aircraft Using Online Calibration," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 1, pp. 420–434, Feb. 2022.

[7] J. Quintas *et al.*, "AUV path planning, navigation, and control using geophysical data," in *Proc. OCEANS*, Jun. 2019, pp. 1–9.

[8] J. Jung *et al.*, "Navigation of Unmanned Surface Vehicles Using Underwater Geophysical Sensing," *IEEE Access*, vol. 8, pp. 208 707–208 717, 2020.

[9] J. M. Quinn *et al.*, "The joint US/UK 1990 epoch world magnetic model," Tech. Rep. NASA-CR-190424, Oct. 1991.

[10] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robot. Auton. Syst.*, vol. 57, no. 10, pp. 1028–1035, Oct. 2009.

[11] B. Li *et al.*, "How feasible is the use of magnetic field alone for indoor positioning?" in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Nov. 2012, pp. 1–9.

[12] M. Angermann *et al.*, "Characterization of the indoor magnetic field for applications in Localization and Mapping," in *Proc. Int. Conf. Indoor Position. Indoor Navig.*, Nov. 2012, pp. 1–9.

[13] M. Frassl *et al.*, "Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Nov. 2013, pp. 913–920.

[14] A. Penumarti *et al.*, "Real-Time Uncertainty-Aware Motion Planning for Magnetic-Based Navigation," *IEEE Access*, vol. 14, pp. 23 824–23 834, 2026.

[15] A. Penumarti *et al.*, "Neural Network-Based Magnetometer Sensor Model for Virtual Magnetic-Based Navigation Testbed," in *Proc. AIAA SciTech Forum*, ser. AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan. 2025.

[16] M. Montemerlo *et al.*, "FastSLAM: a factored solution to the simultaneous localization and mapping problem," in *Proc. AAAI Conf. Artif. Intell.* USA: American Association for Artificial Intelligence, Jul. 2002, pp. 593–598.

[17] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[18] A. Kemppainen, I. Vallivaara, and J. Röning, "Magnetic field SLAM exploration: Frequency domain Gaussian processes and informative route planning," in *Proc. Eur. Conf. Mobile Robots*, Sep. 2015, pp. 1–7.

[19] T. N. Lee and A. J. Canciani, "MagSLAM: Aerial simultaneous localization and mapping using Earth's magnetic anomaly field," *Navigation*, vol. 67, no. 1, pp. 95–107, Mar. 2020.

[20] A. Solin *et al.*, "Modeling and Interpolation of the Ambient Magnetic Field by Gaussian Processes," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1112–1127, Aug. 2018.

[21] M. Kok and A. Solin, "Scalable Magnetic Field SLAM in 3D Using Gaussian Process Maps," in *Proc. Int. Conf. Inf. Fusion*. Cambridge, United Kingdom: IEEE, Jul. 2018, pp. 1353–1360.

[22] I. Abdul-Raouf *et al.*, "Large scale mapping of indoor magnetic field by local and sparse gaussian processes," in *Proc. Conf. Robot Learn.*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, 2024, pp. 2104–2119.

[23] C. Menzen, M. Fetter, and M. Kok, "Large-scale magnetic field maps using structured kernel interpolation for Gaussian process regression," in *Proc. Int. Conf. Inf. Fusion*, Jun. 2023, pp. 1–7.

[24] S. Pollok *et al.*, "Magnetic field prediction using generative adversarial networks," *J. Magn. Magn. Mater.*, vol. 571, p. 170556, Apr. 2023.

[25] T. Khatib *et al.*, "Robotic Information Gathering via Deep Generative Inpainting," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, Oct. 2023, pp. 3130–3137.

[26] B. Jacob *et al.*, "E-PINNs: Epistemic Physics-Informed Neural Networks," Mar. 2025, arXiv:2503.19333 [cs].

[27] L. Yang, X. Meng, and G. E. Karniadakis, "B-PINNs: Bayesian Physics-Informed Neural Networks for Forward and Inverse PDE Problems with Noisy Data," *J. Comput. Phys.*, vol. 425, p. 109913, Jan. 2021.

[28] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proc. Int. Conf. Mach. Learn.* Proc. Mach. Learn. Res., Oct. 2016.

[29] J. M. Hernandez-Lobato and R. Adams, "Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks," in *Proc. Int. Conf. Mach. Learn.* PMLR, Jun. 2015, pp. 1861–1869.

[30] R. M. Neal, *Bayesian Learning for Neural Networks*, ser. Lecture Notes in Statistics, P. Bickel *et al.*, Eds. New York, NY: Springer, 1996, vol. 118.

[31] L. V. Jospin *et al.*, "Hands-on Bayesian Neural Networks – a Tutorial for Deep Learning Users," *IEEE Comput. Intell. Mag.*, vol. 17, no. 2, pp. 29–48, May 2022.

[32] R. M. Neal, "MCMC Using Hamiltonian Dynamics," in *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

[33] C. Blundell *et al.*, "Weight Uncertainty in Neural Network," in *Proc. Int. Conf. Mach. Learn.* PMLR, Jun. 2015, pp. 1613–1622.

[34] Y. Liu, R. Ni, and A. H. Qureshi, "Physics-Informed Neural Mapping and Motion Planning in Unknown Environments," *IEEE Trans. Robot.*, vol. 41, pp. 2200–2212, 2025.

[35] P. W. Livermore *et al.*, "Reconstructions of Jupiter's magnetic field using physics-informed neural networks," *Mon. Not. R. Astron. Soc.*, vol. 533, no. 4, pp. 4058–4067, Sep. 2024.

[36] M. D. Hoffman and A. Gelman, "The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo," *J. Mach. Learn. Res.*, vol. 15, no. 47, pp. 1593–1623, 2014.

[37] P. Wagner, X. Wu, and M. F. Huber, "Kalman Bayesian Neural Networks for Closed-Form Online Learning," *Proc. AAAI Conf. Artif. Intell.*, vol. 37, no. 8, pp. 10 069–10 077, Jun. 2023.

[38] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Sep. 2003.

[39] J. Kirkpatrick *et al.*, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[40] P. Carreno, D. Kulic, and M. Burke, "Adapting Neural Models with Sequential Monte Carlo Dropout," in *Proc. Conf. Robot Learn.* PMLR, Mar. 2023, pp. 1542–1552.

[41] A. Doucet, N. Freitas, and N. Gordon, Eds., *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer, 2001.

[42] F. Saad *et al.*, "Sequential Monte Carlo Learning for Time Series Structure Discovery," in *Proc. Int. Conf. Mach. Learn.* PMLR, Jul. 2023, pp. 29 473–29 489.

[43] A. Millard *et al.*, "Investigating Batch Inference in a Sequential Monte Carlo Framework for Neural Networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, May 2026, pp. 5036–5040.

[44] R. Douc and O. Cappe, "Comparison of resampling schemes for particle filtering," in *Proc. Int. Symp. Image Signal Process. Anal.*, Sep. 2005, pp. 64–69.

[45] S. Macenski *et al.*, "Robot Operating System 2: Design, architecture, and uses in the wild," *Sci. Robot.*, vol. 7, no. 66, p. eabm6074, May 2022.

[46] G. M. Hoffmann *et al.*, "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing," in *Proc. Amer. Control Conf.*, Jul. 2007, pp. 2296–2301.

[47] K. He *et al.*, "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1026–1034.