EFFICIENT MESSAGE PASSING FOR PARTIAL DIFFERENTIAL EQUATION PRIORS

Anonymous authors
Paper under double-blind review

ABSTRACT

Prior information for real-world physical quantities is most elegantly expressed via partial differential equations (PDEs). In this paper, we propose a novel way to solve PDEs using probabilistic inference on a factor graph. In general, factor graphs provide a natural way to encode prior knowledge into a model as explicit factors; here, this knowledge is provided by a governing PDE, which narrows the solution space, while observed data further shape the posterior over the parameters. The approximate parameter posterior is inferred using message passing based on moment matching, without posterior sampling or global gradient-based optimization. We demonstrate our approach on the first-order advection and the second-order semi-linear Fisher-KPP equations, where it achieves predictive accuracy comparable to a standard baseline while providing structured predictive uncertainty. Moreover, the inferred posterior marginal means and uncertainty structure match more closely those obtained using Hamiltonian Monte Carlo than the evaluated mean-field variational inference baseline, while requiring up to 10× less training time in our experiments, with inference speed comparable to variational inference.

1 INTRODUCTION

Partial differential equations (PDEs) often describe physical processes in space and time. In many practical problems, a known parameterized equation is accompanied by a set of observations that provide additional information and further constrain the problem. Combining these observations with the PDE forms a machine learning problem with multiple objectives called *physics-informed learning*. The idea of solving this multiobjective problem by training a neural network was introduced in Raissi et al. (2019) and termed a physics-informed neural network. (PINN). There, the solution of an equation is represented by a neural network trained using both observed solution values and the governing PDE. We adopt the same setup in this work, but instead of solving it as a global optimization problem, we formulate the *PINN as a factor graph*, where the neural network learns from the observed data probabilistically, and the governing *PDE acts as prior physical knowledge* that constrains the solution space.

While effective for many problems, the standard PINN formulation yields only a point estimate of the network parameters; our work instead adopts a Bayesian formulation and infers a posterior distribution over these parameters. Existing Bayesian methods for PINNs (Yang et al., 2021; Zong et al., 2025; Raj et al., 2025) typically perform posterior inference through Hamiltonian Monte Carlo (HMC; Duane et al., 1987) or gradient-based variational inference (VI; Jordan et al., 1999). Sampling can be computationally expensive, while practical variational approximations may differ substantially from sampling-based references, especially in predictive uncertainty (Blei et al., 2017).

We approach this inference problem from a different angle and use *message passing*: by propagating information directly through the factor graph, we obtain an efficient, gradient-descent- and sampling-free framework for approximate Bayesian inference over the PDE solution. Our goal is not to establish message passing as a general replacement for Bayesian PINNs, but to test whether this inference perspective is viable in a physics-informed setting. We use HMC as a sampling-based reference and demonstrate that our approach yields similar results in prediction accuracy and posterior parameter estimates, while being more time-efficient.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Problem formulation In this work, we solve the PDE *forward problem*, i.e., predicting the future states of a parameterized physical system. For the shallow neural representation used throughout this work, the trainable parameters $\boldsymbol{\theta} := \boldsymbol{W}$ are treated as random variables with prior $p(\boldsymbol{\theta})$, and the learning objective is to infer their posterior distribution from both observed data and physical constraints. Given observations $\mathcal{D} = \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ and PDE constraints imposed at collocation points $\mathcal{C} = \{\boldsymbol{x}_m\}_{m=1}^M$ and modeled through PDE residual variable R, the target distribution is

$$p(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{C}) \propto p(\boldsymbol{\theta})\, p(\mathcal{D} \mid \boldsymbol{\theta})\, p(R = 0 \mid \mathcal{C}, \boldsymbol{\theta}). \tag{1}$$

Our message passing algorithm approximates this posterior with a Gaussian belief $q(\boldsymbol{\theta})$:

$$q(\boldsymbol{\theta}) \approx p(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{C}). \tag{2}$$

In this work, we focus on first-order linear PDE constraints of this form for a given coefficient vector $\boldsymbol{b}$:

$$\boldsymbol{b}^\top \nabla_{\boldsymbol{x}} f_U(\boldsymbol{x}; \boldsymbol{\theta}) = 0. \tag{3}$$

Our contributions We aim to answer the following question: Can approximate Bayesian inference on a factor graph solve PDEs from observed data and encoded prior physical knowledge using message passing? Our contributions are summarized as follows:

- **Probabilistic factor-graph formulation:** We introduce a novel perspective on PDE solving as probabilistic inference on a factor graph (Section 3). In this graph, a neural network is trained to satisfy the observed data, while incorporated as prior physical knowledge PDE narrows the solution space. Deterministic relations between variables are encoded as Dirac-delta distributions, and therefore messages are propagated with the Direct Message Approximation (DMA; Herbrich et al., 2026) framework; more details are in Definition 3 in Section 2.1. We derive the required factor messages for the first-order linear advection equation and the second-order semi-linear Fisher-KPP with a nonlinear reaction term.
- **Competitive PDE solution accuracy with uncertainty estimates:** In our experiments, the message passing approach recovers PDE solutions that closely match the exact solutions, while additionally providing uncertainty estimates (Section 4.1). We further observe that the predicted variance exhibits structured behavior across the evaluated PDE dynamics.
- **Posterior marginal inference close to HMC at substantially lower computational cost:** Compared with HMC and VI, our approach recovers similar posterior marginal means while avoiding posterior sampling and gradient-based variational optimization (Section 4.2). It is consistently faster than HMC and remains computationally competitive with VI, while its posterior uncertainty is closer in structure to the HMC reference.

2 PRELIMINARIES

To frame our approach, we first revisit the fundamental concepts and introduce notation for approximate message passing on factor graphs.

2.1 PROBABILISTIC INFERENCE ON FACTOR GRAPHS

Definition 1. A *factor graph* is a bipartite graph with two types of nodes - representing variables and functions of these variables. In the probabilistic inference context, it represents a joint density of random variables that can be computed as the product of all factors. Hidden variables of this probabilistic model are inferred with a *message passing* algorithm.

In this work, we use a *Gaussian approximation* for all information flow on the factor graph, i.e., all variables and messages are represented by Gaussian distributions. When arithmetically possible and computationally efficient, we compute the resulting moments exactly; in most cases, however, we approximate the resulting distribution by a Gaussian via moment-matching. As stated in Equation 2, our inferred posterior is approximate - we elaborate on it further in Appendix A.2.

Definition 2. *Message passing* generally refers to a framework in which nodes in a complex system exchange simple local messages to perform inference (MacKay, 2003). In the probabilistic inference

context, these nodes are random variables, factors encode statistical dependencies between them, and messages are distributions that communicate information about these variables. In particular, we use the *sum-product algorithm* (Kschischang et al., 2001), in which information is propagated through the factor graph by passing messages between variable and factor nodes: variable nodes collect information from their neighboring factors, while factor nodes combine this information according to the local dependency they represent and pass the resulting information to neighboring variables. For a general introduction to the sum-product algorithm and Gaussian arithmetic, refer to Appendix A.1.1. Throughout this work, we use the notation $m_{X_k \to f_i}(x_k)$ and $m_{f_i \to X_k}(x_k)$ to denote messages passed from a variable node X_k to a factor node f_i and from a factor node f_i to a variable node X_k, respectively.

Definition 3. *Direct Message Approximation* (DMA; Herbrich et al., 2026) is a framework that extends the sum-product algorithm. Instead of approximating the marginals, Gaussian messages from factors to variable, $m_{f_i \to X_k}(x_k)$ are directly approximated via moment-matching. In our PDE factor graph, most relations between variables are modeled as Dirac-delta factors, which make DMA the only admissible framework for such message passing; we therefore use this approach by default throughout our work. More details on the background and its relation to other frameworks, such as Expectation Propagation (Minka, 2001) and Variational Message Passing (Winn et al., 2005), can be found in the original paper.

Bayesian neural network We define a single-layer Bayesian neural network with a single non-linear output unit, trainable parameters $\boldsymbol{\theta} := \boldsymbol{W}$, an activation function $g(\cdot)$, and a basis feature vector function $\phi(\cdot)$. The input $\boldsymbol{x} \in \mathbb{R}^d$ is mapped to the basis feature vector $\phi(\boldsymbol{x}) \in \mathbb{R}^P$, where $P := \dim \phi(\boldsymbol{x})$, and $\boldsymbol{W} \in \mathbb{R}^P$. In our Bayesian formulation, the weights, preactivation, and network output are represented as Gaussian random variables, with

$$W_k \sim \mathcal{N}(W_k; \cdot, \cdot)\,, \quad z(\boldsymbol{x}) = \boldsymbol{W}^\top \phi(\boldsymbol{x}), \quad f_U(\boldsymbol{x}; \boldsymbol{\theta}) = g(z(\boldsymbol{x}))$$

where if the input $\boldsymbol{x}$ is clear from context, we write $z := z(\boldsymbol{x})$. We denote the mean and variance of the current Gaussian belief for W_k by μ_k and σ_k^2, respectively. Note that we define this shallow architecture as it is sufficient for our current experiments, and use the same architecture for all baselines to ensure a consistent comparison. For the one-dimensional spatiotemporal PDEs considered further, we write $\boldsymbol{x} = (s, t)$, where s denotes the spatial coordinate and t denotes time. For brevity, we sometimes write $f(\boldsymbol{x}; \boldsymbol{\theta}) := f_U(\boldsymbol{x}; \boldsymbol{\theta})$ when no ambiguity arises.

2.2 Differentiable activation function

The employed neural network must be differentiable up to the order of the differential equation being considered; i.e., for an n-th-order differential equation, the required partial derivatives of the network output up to order n must exist and be non-constant. Therefore, we have to choose the activation function $g(\cdot)$ and the feature map $\phi(\cdot)$ to be n-times differentiable. We provide more details on constructing $\phi(\cdot)$ in Appendix A.3.

In previous work on PINNs, the hyperbolic tangent is commonly used as an activation function (Raissi et al., 2019) - it is infinitely differentiable and analytic on all of $\mathbb{R}$ and therefore satisfies the requirement for a PDE of order n that g be n times differentiable almost everywhere, except on a null set. However, our message passing framework introduces an additional condition on the activation function g: we also require that for Gaussian X the moments of $g^{(k)}(X)$ and, where defined, its inverse $\left(g^{(k)}\right)^{-1}(X), k \in \{0, ..., n\}$ can be computed or accurately approximated; when the inverse is not uniquely defined, the corresponding backward message can instead be obtained through a suitable local approximation. This condition allows the activation and its derivatives to be propagated through the factor graph via moment matching.

The exponential function is a natural choice: it is infinitely differentiable, and exponentiated Gaussian variables have closed-form moments, as they follow a log-normal distribution. To keep these properties while allowing the activation function to also take negative values, we define

$$g(x) = \begin{cases} e^{\beta x} - 1, & x > 0, \\ -\alpha \left(e^{-\beta x} - 1\right), & x \leq 0, \end{cases} \tag{4}$$

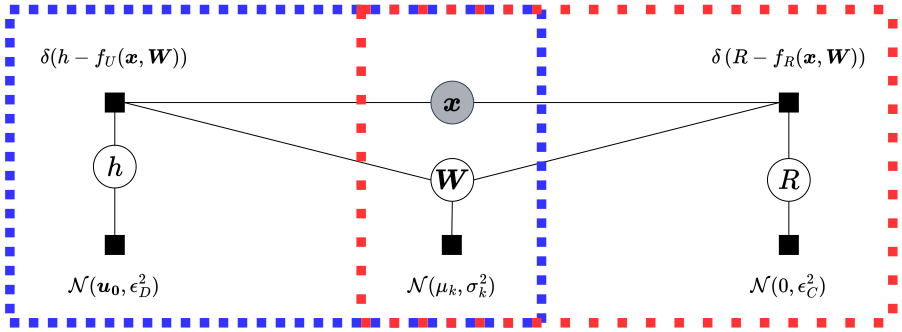

Figure 1: A generic factor graph for PDEs. White circle nodes refer to the hidden variables, the gray node to the observed data $\boldsymbol{x}$, and black squares to the functions (factors). Left subgraph (blue rectangle): data subgraph, where $f_U(\boldsymbol{x}, \boldsymbol{W})$ models a neural network that is trained to match the label data - only initial condition in our case. Right subgraph (red rectangle): PDE subgraph, that enforces a PDE on a subset of collocation points $\mathcal{C}$ with f_R.

where $0 < \alpha < 1$ and $\beta > 0$. The function is continuous and piecewise infinitely differentiable; its only non-differentiable point is $x = 0$, which has zero probability under a non-degenerate Gaussian preactivation. The corresponding Gaussian message approximations are derived in Appendix A.4, using Lemma 1.

Note that we do not claim any advantage of our message passing approach from the introduction of this activation function – in all baseline comparisons, we use the same activation function. Rather, we introduce it as a smooth exponential analog of the leaky ReLU, suitable for our message passing approach. In particular, for inputs close to zero, its two exponential branches are approximately linear, $e^{\beta x} - 1 \approx \beta x$ for $x > 0$ and $-\alpha(e^{-\beta x} - 1) \approx \alpha\beta x$ for $x \leq 0$, giving different slopes on the positive and negative sides as in the leaky ReLU.

3 Our Factor Graph

For clarity, we split the factor graph into two subgraphs: the *PDE subgraph* and the *data subgraph*. The PDE graph enforces the PDE at collocation points $\mathcal{C}$ from the domain with a PDE factor f_R (Section 3.1), while the data graph incorporates observed function values and updates weights with f_U to learn $\mathcal{D}$ (Section 3.2). In our PINN setting, these observations are given only at the initial condition, $t = 0$. These subgraphs are shown in Figure 1: the neural network parameters W are shared between the two subgraphs and get updated on both sides. All deterministic relations between variables are modeled as Dirac-delta distributions, $\delta(\cdot)$. For a fixed neural network architecture, the data subgraph remains unchanged regardless of the PDE being solved; only the PDE factor f_R in the PDE subgraph needs to be adapted to the specific PDE. In Section 3.3, we elaborate further on the schedule for message updates.

3.1 PDE subgraph

The PDE subgraph incorporates the governing equation through a *residual factor* $f_R(\boldsymbol{x}, \boldsymbol{W})$. At each collocation point $\boldsymbol{x} \in \mathcal{C}$, the residual $R(\boldsymbol{x}, \boldsymbol{W}) \sim \mathcal{N}\big(0, \epsilon_C^2\big)$ measures how well the network output satisfies the PDE and is constrained around zero. Thus, the residual factor connects the PDE constraint at a collocation point to the weight variables. Its specific form is determined by the operator of the desired PDE, while the generic message-passing procedure remains unchanged. We do not enforce the PDE residual R to be exactly zero; instead, we use $R \sim \mathcal{N}\big(\cdot; 0, \epsilon_C^2\big)$; an additional ablation study on the PDE residual prior in Appendix A.7.3 demonstrates that if we tighten the space too much, the solution can exhibit the failure mode.

Forward message Propagate the current weight distributions through the PDE operator and moment-match the resulting residual distribution,

$$m_{f_R \to R}(R) \approx \mathcal{N}(R;\, \mathbb{E}_{\boldsymbol{W}}[f_R(\boldsymbol{W}, \boldsymbol{x})],\, \mathrm{Var}_{\boldsymbol{W}}(f_R(\boldsymbol{W}, \boldsymbol{x}))) \,.$$

Backward message Propagate the residual R and all remaining weights $\boldsymbol{W}_{\setminus k}$ through the PDE operator and moment-match the resulting distribution,

$$m_{f_R \to W_k}(W_k) \approx \mathcal{N}\big(W_k;\, \mathbb{E}_{R,\boldsymbol{W}_{\setminus k}}[f_R(\boldsymbol{W}, \boldsymbol{x})],\, \mathrm{Var}_{R,\boldsymbol{W}_{\setminus k}}(f_R(\boldsymbol{W}, \boldsymbol{x}))\big)\,.$$

Note that in real equations, it is usually not feasible to solve this equation for W_k in a closed form; for this reason, we use a first-order Taylor expansion around parameter means. As an illustration, the residual of the advection equation is

$$c(\boldsymbol{W}, \boldsymbol{x}) := \boldsymbol{b}^\top \nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \boldsymbol{\theta}) = g'(z) \boldsymbol{W}^\top \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i}. \tag{5}$$

Then $f_R(\boldsymbol{W}, \boldsymbol{x}) = \delta\left(R - c(\boldsymbol{W}, \boldsymbol{x})\right)$ denotes the factor that connects the weight variables $\{W_k\}_k$ and the residual variable R, as shown in Figure 1. The derivations for the forward $m_{W \to f_R}(\boldsymbol{W})$ and the backward $m_{f_R \to W}$ are shown in the Appendix A.5.1. The same principle can be reused for other PDEs once the corresponding residual factor and its approximate messages have been derived; Appendix A.5.2 provides the Fisher-KPP equation as a second example with a different differential order and a nonlinear reaction term. Therefore, adapting the framework to a different PDE requires modifying the residual factor, while the general message passing procedure remains unchanged.

3.2 Data subgraph

The data subgraph forces the neural network to learn the observed values from the initial condition $(\boldsymbol{x}_n, y_n) \in \mathcal{D}$ with a factor f_U. With our simple architecture, we can compute forward and backward messages only using known Gaussian properties and the approximations of the activation function moments, derived in Appendix A.4.

Forward message Given the incoming messages to weights, we can compute the moments of $z_n := \boldsymbol{W}^\top \phi(\boldsymbol{x}_n) \sim \mathcal{N}(\cdot; \mu_{z_n}, \sigma^2_{z_n})$ exactly. The activation moments are derived in Appendix A.4, giving $f(\boldsymbol{x}_n; \boldsymbol{\theta}) \approx \mathcal{N}(\cdot; \mu_{h_n}, \sigma^2_{h_n})$.

Backward message To propagate the observation y_n back to the weights, we first moment-match the nonlinear joint distribution of z_n and $h_n = g(z_n)$ with a Gaussian distribution. This constitutes an additional Gaussian approximation in the backward update, while retaining the dependence between z_n and h_n through their covariance. Approximating the joint distribution of z_n and h_n by a Gaussian with the corresponding moments and conditioning on the observation $h_n = y_n$ gives

$$\mu_{z_n|y_n} = \mu_{z_n} + \frac{\mathrm{Cov}(z_n, h_n)}{\sigma^2_{h_n} + \epsilon^2_D}(y_n - \mu_{h_n}), \quad \sigma^2_{z_n|y_n} = \sigma^2_{z_n} - \frac{\mathrm{Cov}(z_n, h_n)^2}{\sigma^2_{h_n} + \epsilon^2_D},$$

where ϵ_D refers to the modeled noise in labels as shown in Figure 1 and the covariance can be evaluated using Stein's lemma, introduced in Equation 9, $\mathrm{Cov}(z_n, h_n) = \sigma^2_{z_n}\, \mathbb{E}[g'(z_n)]$.

Since $z_n = \sum_j \phi_j(\boldsymbol{x}_n) W_j$, we isolate W_k and moment-match its distribution:

$$\begin{aligned} W_k &= \frac{z_n - \sum_{j \neq k} \phi_j(\boldsymbol{x}_n) W_j}{\phi_k(\boldsymbol{x}_n)}, \quad \mu_{W_k} = \frac{\mu_{z_n|y_n} - \sum_{j \neq k} \phi_j(\boldsymbol{x}_n) \mu_j}{\phi_k(\boldsymbol{x}_n)}, \\ \sigma^2_{W_k} &= \frac{\sigma^2_{z_n|y_n} + \sum_{j \neq k} \phi_j(\boldsymbol{x}_n)^2 \sigma^2_j}{\phi_k(\boldsymbol{x}_n)^2}. \end{aligned} \tag{6}$$

Thus, the backward message is approximated as $m_{f_U \to W_k}(W_k) \approx \mathcal{N}(W_k; \mu_{W_k}, \sigma^2_{W_k})$.

3.3 Message passing schedule

Each subgraph alone can significantly narrow the solution space without satisfying the full model. As shown in Figure 5 (Appendix A.1.2), the data graph learns the initial condition but not its time evolution, but already significantly narrows the solution space. The PDE graph alone just propagates

	Advection				Fisher-KPP			
	Ours	VI	HMC	SGD	Ours	VI	HMC	SGD
RMSE ($\times 10^{-2}$)	4.89	19.5	4.85	**4.86**	**2.72**	72.7	2.73	13.7
Cov. 90 % (%)	**14.5**	11.1	27.1	–	**50.0**	0.0	68.3	–
CRPS ($\times 10^{-3}$)	**39.0**	160	37.9	39.2	**3.78**	718	3.65	112

Table 1: Predictive accuracy and uncertainty diagnostics against the exact solutions, aggregated over all runs. For the metric descriptions, refer to Appendix A.6. **RMSE**: mean per-run root mean square error of the predictive mean. **Cov. 90 %**: fraction of grid points for which the 90% predictive interval contains the exact solution. **CRPS**: continuous ranked probability score. HMC (grey) serves as the reference. Stochastic gradient descent (SGD) refers to the non-Bayesian PINN baseline (Section 4.1); its CRPS reduces to the mean absolute error. Best results among our method, VI, and SGD are highlighted in bold.

the prior over time without conditioning on the observed solution. A fixed point of either subgraph satisfies only $p(\boldsymbol{\theta} \mid \mathcal{D})$ or $p(\boldsymbol{\theta} \mid \mathcal{C})$, rather than the joint posterior. As introduced in Section 2.1, Gaussian messages combine additively in their natural parameters. We therefore interleave initial-condition and PDE updates, allowing both constraints to shape the posterior jointly and requiring empirically observed convergence to a fixed point of $p(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{C})$ rather than either subgraph alone; refer to more details and an ablation study on this in Appendix A.1.2.

4 Evaluation

Our evaluation asks three questions: (1) whether message passing recovers accurate PDE solutions, (2) how closely its posterior marginal statistics and predictive uncertainty agree with a sampling-based HMC reference, and (3) what computational cost this requires relative to HMC and mean-field VI. We use HMC as a reference for approximate posterior inference rather than as ground truth. We use the advection equation as a representative first-order PDE for quantitative comparison and uncertainty analysis; it has the form presented in Equation 3 with $\boldsymbol{b} = (b, 1)^\top$, and the residual function f_R is defined in Equation 5. One step further, we also evaluate our approach on the Fisher-KPP equation, a semi-linear second-order PDE, whose residual is given in Equation 12.

Experimental setup We evaluate on two PDEBench (Takamoto et al., 2022) datasets, which are generated by computing future states of the system from given initial conditions using known exact solutions of the PDEs. We take 1D advection across five wave speeds, $b \in \{0.1, 0.4, 1, 2, 7\}$, with ten trajectories per wave speed, and 1D reaction–diffusion across its full 16-setting parameter grid, $(\nu, \rho) \in \{0.5, 1, 2, 5\} \times \{1, 2, 5, 10\}$, with five trajectories per setting, all drawn from canonical seeded subsets. All the baselines we compare to share the same architecture, feature basis, labeled and collocation points, initialization, and random seeds as our model. All experiments, including the timing measurements, ran on a single CPU-only machine; more details on the experimental setup are provided in Appendix A.6.

4.1 Solution fidelity

Message passing compared to exact solutions We first compare our predictions against exact solutions, i.e., those generated by PDEBench datasets. Figure 2 shows the predictions for one of the advection trajectories in detail: across the evaluated wave speeds, the posterior mean follows the correct pattern of the ground-truth solution. At the same time, the posterior variance exhibits a structured dependence on the prediction horizon: it is lowest near the observed initial condition and generally increases as the solution is propagated further into the future. Figure 7(c) shows this pattern across the evaluated advection speeds, and related behavior is observed for Fisher-KPP in Figures 11 and 12. We interpret this as evidence that the inferred uncertainty responds to the available information and PDE dynamics.

Message passing compared to a baseline As a point-estimate reference, we adopt a non-Bayesian PINN baseline from Takamoto et al. (2022) and train it with Adam using the standard PINN loss

(SGD baseline). For all investigated trajectories, we observe that our approach achieves performance very close to that of the baseline model: Figure 6 for the advection and Figure 13 for the Fisher-KPP show that the mean squared error (MSE) and the PDE residual R are almost identical, while for the Fisher-KPP, our approach is even slightly better. In other words, this demonstrates that, given a nearly identical setup, our message passing training procedure yields results very close to those of a classical gradient-based training procedure. Table 1 summarizes the predictive results: our root mean square error (RMSE) is close to HMC and the baseline referred to as stochastic gradient descent (SGD) on advection and to HMC on Fisher-KPP, although 90 % coverage remains below nominal.

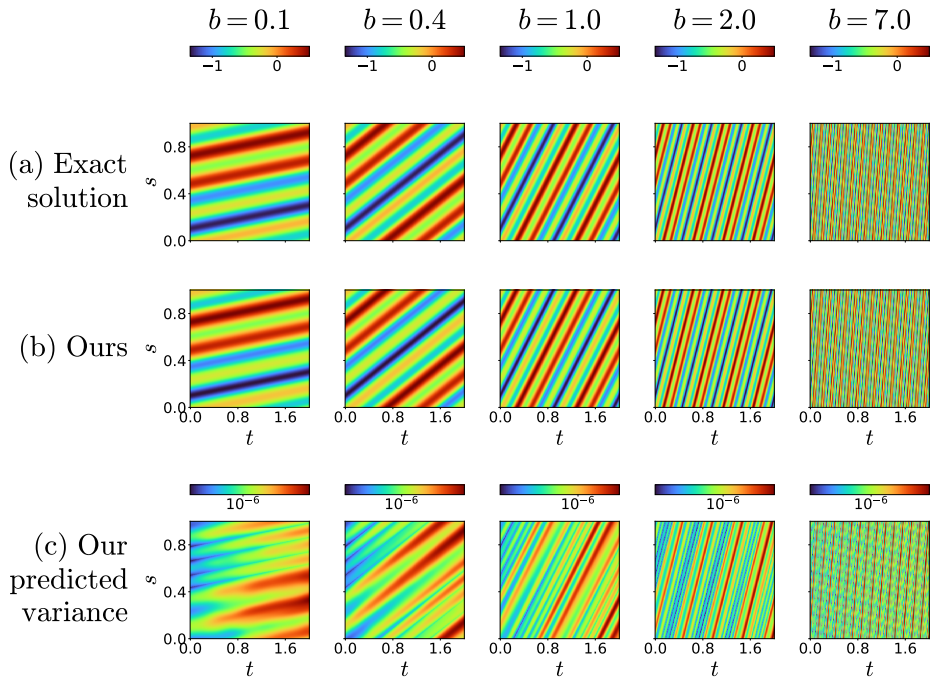

Figure 2: Comparison of the message passing approach (ours) and the baseline solution from PDEBench for the advection equation across different wave speeds, $b \in \{0.1, 0.4, 1, 2, 7\}$. All heatmaps are plotted in the coordinates of $\boldsymbol{x} = (s, t)$, where s denotes space and t denotes time. **(a)** the exact solution; **(b)** predicted mean values of our message passing solution, which preserve the propagation pattern across the evaluated wave speeds; **(c)** predicted variance values of our message passing solution, showing increasing uncertainty further in time from the observed initial condition.

4.2 Bayesian comparison

We compare our approach to HMC and VI in terms of the similarity of the inferred parameter posteriors and the training and inference time costs. To compare the predicted uncertainty, we use the continuous ranked probability score (CRPS) and 90 % coverage (see Appendix A.6). Table 1 shows CRPS close to HMC for both PDEs, with lower 90 % coverage, while VI is markedly worse.

Message passing compared to Hamiltonian Monte Carlo For the HMC baseline, we use Algorithm 1 from Yang et al. (2021), with a Metropolis-Hastings acceptance step and Gaussian likelihoods on the observed initial condition and the PDE residual. To ensure a fair comparison, we reimplemented HMC in Julia using the same architecture, priors, and random seeds as our message passing approach. Using HMC as a sampling-based reference, we find close agreement in posterior marginal means, as shown in Figures 3(a) and 15(a). The marginal standard deviations do not exactly match, but generally follow the same structure and are substantially closer to the HMC reference than those obtained with the mean-field VI baseline; Figures 7 and 18 quantify the remaining differences in predictive uncertainty. These comparisons concern marginal first- and second-order

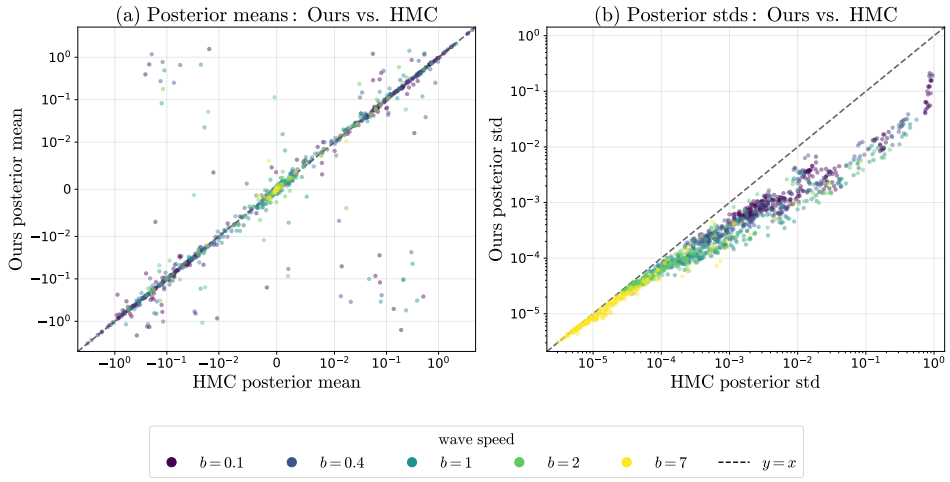

Figure 3: Comparison of our approach (ours) to the HMC implemented based on Yang et al. (2021) for the advection equation. We map posterior weights of HMC (x-axis) to our approach (y-axis) - points in **(a)** refer to the mean values, points in **(b)** refer to the standard deviations; our approximate approach converges to posterior weights close to the ones from HMC. For readability, 25 random weights per trajectory are shown.

statistics and do not establish recovery of the full joint HMC posterior. In the evaluated models, message passing is substantially cheaper computationally than HMC because it avoids posterior sampling (Figures 4 and 14).

Message passing compared to variational inference For the VI baseline, we use Algorithm 2 from Yang et al. (2021); again, we reimplement it in Julia to enable a fair comparison. Figure 8 for the advection and Figure 16 for the Fisher-KPP show that our approach still infers similar posterior means, but for posterior variances, VI yields different estimates from both HMC and our approach – Figures 9 and 17 additionally illustrate the difference between HMC and VI for both equations, respectively. In general, VI is known to often underestimate posterior variances (Blei et al., 2017); therefore, we regard the comparison with HMC as the more reliable reference. The computational efficiency of our approach compared to VI is shown in Figure 4 for the advection and in Figure 14 for the Fisher-KPP: our training is faster at every wave speed, while inference is equally fast, since both methods compute the predictive distribution in closed form.

5 Related Work

Sampling and variational inference The foundational Bayesian approach, Bayesian PINN (B-PINN; Yang et al. 2021), models neural network parameters as random variables and infers their posterior using either Hamiltonian Monte Carlo (HMC) or variational inference (VI). HMC is shown to provide more accurate posterior estimates; however, it requires computationally expensive posterior sampling. Moreover, VI learns an approximate posterior using stochastic gradient-based optimization; this is where our approach differs: we propose a more natural probabilistic solution based on message passing, in which inference is performed by directly propagating distributions through the model. Another line of work aims to improve HMC performance through a randomized loss function (Zong et al., 2025), but still relies on extensive posterior sampling. (Raj et al., 2025) proposes a more advanced VI approach, where a neural operator parameterizes the variational posterior and the corresponding evidence lower bound (ELBO) incorporates both data and PDE residual terms, but inference still relies on gradient-based optimization and Monte Carlo sampling.

Physics-informed priors In contrast to approaches that incorporate the PDE directly during posterior inference, the work of (Menicali et al., 2025) uses physics to construct the Bayesian neural network (BNN) prior. The prior means are first obtained by optimizing a PINN objective

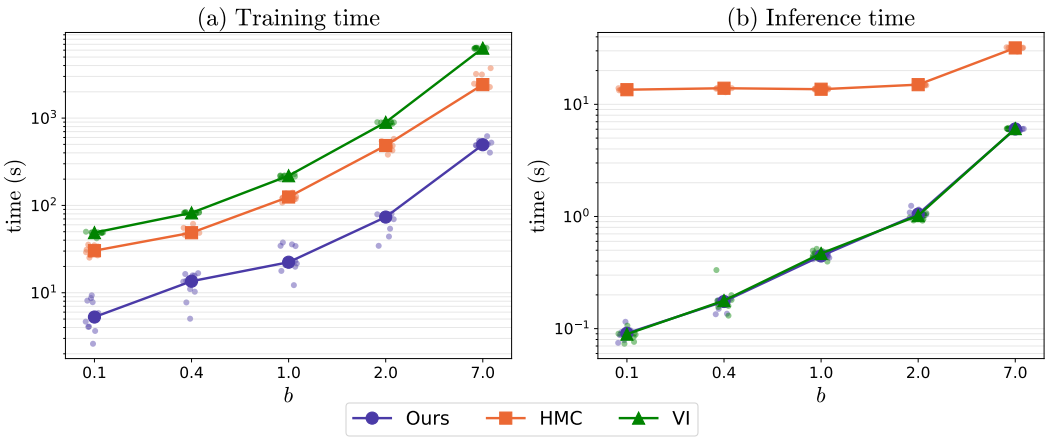

Figure 4: Computational cost of our approach (ours), HMC and VI for the advection equation across wave speeds b, ten trajectories each (points: trajectories; lines: medians). **(a)** Training time, cold start; **(b)** inference time of the predictive mean and variance on the full grid, steady state. Our training is up to $6.6\times$ faster than HMC and up to $13\times$ faster than VI; our closed-form inference is as fast as VI's and up to $150\times$ faster than HMC's sampling-based inference.

that includes the PDE, initial conditions, and boundary conditions. The resulting physics-informed prior is then updated with the observed data via Bayes by Backprop (Blundell et al., 2015), which performs variational inference via stochastic gradient-based optimization.

Gaussian processes A different line of work replaces the BNN with a Gaussian process (GP), placing a probability distribution directly over the PDE solution rather than over neural network parameters. Since derivatives of a GP are again jointly Gaussian under suitable kernels, the solution and its derivatives can be modeled jointly, which provides a natural way to incorporate differential constraints. (Paun et al., 2025) constructs such a physics-informed GP by conditioning on the PDE at a finite set of points and performs posterior inference using HMC. (Chkrebtii et al., 2016) jointly models the solution and its derivatives with a GP and sequentially updates their distributions using local differential-equation evaluations obtained from sampled states.

6 Conclusion

Summary Factor graphs allow different sources of prior knowledge to be incorporated directly into the probabilistic model through separate constraints. Building on this property, this work proposes a novel perspective on solving PDEs without sampling or gradient-descent optimization: formulating them as factor graphs for Bayesian neural networks that encode prior physical knowledge. Inference is performed via message passing, and because both subgraphs contain deterministic relations represented by Dirac delta factors, DMA is a suitable message passing framework for our factor graph. Our evaluation of the advection and Fisher-KPP equations demonstrates the feasibility of our approach, and the Bayesian comparison shows that it recovers posterior marginal means close to the HMC reference at substantially lower computational cost, while producing uncertainty estimates that are closer in structure to HMC than the evaluated mean-field VI baseline.

Discussion This work provides a first demonstration of Bayesian PDE learning through message passing on a factor graph. We deliberately focus on a shallow Bayesian neural network with a single nonlinear output unit and two representative PDEs. Within this controlled setting, we show that the same inference framework can recover PDE solutions and structured predictive uncertainty across equations with different differential and nonlinear structures. Our goal at this stage is to establish the feasibility of this new inference perspective rather than its full scalability. Extending the framework to higher-dimensional PDEs and deeper neural networks is therefore a promising direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools for editorial assistance and limited programming assistance. We have not used generative AI tools for generating research ideas, developing the proposed methodology, deriving results, conducting the literature review, interpreting experimental findings, or drawing conclusions. We have reviewed all AI-assisted work by reviewing the AI-assisted code implementations. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not raise any specific ethical concerns.

REPRODUCIBILITY STATEMENT

We provide all experimental details in Appendix A.6, including the hardware configuration, hyperparameters, and other implementation details. The reference implementation is available in the anonymous repository: `https://anonymous.4open.science/r/mp-pinn-release-5CFF/README.md`. It includes scripts for generating the experimental data and reproducing the paper's results.

REFERENCES

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Oksana A Chkrebtii, David A Campbell, Ben Calderhead, and Mark A Girolami. Bayesian solution uncertainty quantification for differential equations. 2016.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Ralf Herbrich, Rainer Schlosser, Jan Lemcke, Johann Ukrow, Anna Kazachkova, Nicolas Alder, Leonhard Hennicke, Theo Bardey, Nico Grimm, Luca Kleinschmidt, Philipp Kolbe, Cezary Kujath, Johanna Schlimme, and Karl Matti Schütz. Direct message approximation (DMA): A consistency-based framework for tractable approximate inference on factor graphs. *arXiv preprint arXiv:2609.29466*, 2026. URL https://arxiv.org/abs/2609.29466.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Agnan Kessy, Alex Lewin, and Korbinian Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018.

Aditi Krishnapriyan, Amir Gholami, Shandian Zhe, Robert Kirby, and Michael Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in neural information processing systems*, 34:26548–26560, 2021.

Frank R Kschischang, Brendan J Frey, and H-A Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Luca Menicali, David H Richter, and Stefano Castruccio. Bayesian neural networks with physics-informed priors with application to boundary layer velocity. *IEEE Transactions on Neural Networks and Learning Systems*, 36(10):18048–18061, 2025.

Thomas Peter Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

L Mihaela Paun, Mitchel J Colebank, and Dirk Husmeier. Physics-informed gaussian processes for nonlinear partial differential equations in a fuid-dynamics application. 2025.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Amogh Raj, Sakol Bun, Keerthana Srinivasa, Carol Eunice Gudumotou, and Arash Sarshar. Deep operator networks for bayesian parameter estimation in pdes. *Computer Physics Communications*, pp. 109853, 2025.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in neural information processing systems*, 35:1596–1611, 2022.

John Winn, Christopher M Bishop, and Tommi Jaakkola. Variational message passing. *Journal of Machine Learning Research*, 6(4):661, 2005.

Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

Yifei Zong, David Barajas-Solano, and Alexandre M Tartakovsky. Randomized physics-informed neural networks for bayesian data assimilation. *Computer Methods in Applied Mechanics and Engineering*, 436:117670, 2025.

A Appendix

A.1 Details on the message passing algorithm

This appendix provides the details needed to reproduce the inference procedure introduced in Section 2.1. Appendix A.1.1 summarizes the sum-product algorithm (Kschischang et al., 2001) and the Gaussian natural parameter arithmetic used by our implementation. Appendix A.1.2 then specifies the randomized update schedule used on the PDE factor graph from Section 3.

A.1.1 Sum-product algorithm

The sum-product algorithm performs inference by exchanging local messages between variable and factor nodes on a factor graph. Intuitively, a variable-to-factor message summarizes the information about a variable provided by all its neighboring factors except the receiving factor, while a factor-to-variable message combines the factor with the incoming information from all other variables connected to it. The marginal of a variable is then computed by combining all messages incoming to that variable. More specifically, given a set of random variables $\mathbf{X} = (X_1, \dots, X_K)$ and a set of factors $\{f_i\}_{i=1}^N$, marginals and messages follow the following rules:

$$p(\boldsymbol{\xi}) \propto \prod_{i=1}^{N} f_i\big(\boldsymbol{\xi}_{\mathrm{ne}(f_i)}\big)\,, \tag{7a}$$

$$m_{X_k \to f_i}(x_k) \propto \prod_{f_j \in \mathrm{ne}(X_k)\setminus\{f_i\}} m_{f_j \to X_k}(x_k), \tag{7b}$$

$$m_{f_i \to X_k}(x_k) \propto \int f_i\big(\boldsymbol{\xi}_{\mathrm{ne}(f_i)}\big) \prod_{X_j \in \mathrm{ne}(f_i)\setminus\{X_k\}} m_{X_j \to f_i}(x_j)\, d\boldsymbol{\xi}_{\mathrm{ne}(f_i)\setminus\{X_k\}}, \tag{7c}$$

where $m_{X_k \to f_i}(x_k)$ is a message from a variable node X_k to a factor node f_i and $m_{f_i \to X_k}(x_k)$ is a message from a factor node f_i to a variable node X_k.

In our implementation, all Gaussian messages and marginals are represented in terms of their natural parameters. For a Gaussian density with mean μ and variance σ^2, we define the precision as $\kappa := \frac{1}{\sigma^2}$ and the precision-weighted mean as $\tau := \frac{\mu}{\sigma^2}$, and denote its natural parameterization by $\mathcal{G}(\cdot; \tau, \kappa)$. Consider two Gaussian densities over the same variable x,

$$p(x) = \mathcal{G}(x; \tau_p, \kappa_p), \qquad q(x) = \mathcal{G}(x; \tau_q, \kappa_q).$$

Then, the multiplication of Gaussian densities corresponds to the addition of their natural parameters,

$$p(x)q(x) \propto \mathcal{G}(x; \tau_p + \tau_q, \kappa_p + \kappa_q),$$

while division corresponds to subtraction,

$$\frac{p(x)}{q(x)} \propto \mathcal{G}(x; \tau_p - \tau_q, \kappa_p - \kappa_q),$$

provided that the resulting precision is positive.

We use these properties for each message update in the employed sum-product algorithm. Besides the convenience, these operations of summation and subtraction are also less computationally expensive than multiplication or division that classic parametrization would require.

A.1.2 Randomized message passing schedule

Algorithm 1 summarizes the randomized message passing schedule for the PDE factor graph defined in Section 3.1. The updates follow the sum-product rules introduced in Section 2.1 and summarized in Equation 7. At the beginning of each epoch, all data and PDE residual factors are placed in a randomized order. Each factor update consists of three stages: (1) a forward update that propagates the incoming weight messages through the factor to obtain an approximate Gaussian message for the network output F or PDE residual R; (2) an observation update that combines this message with the

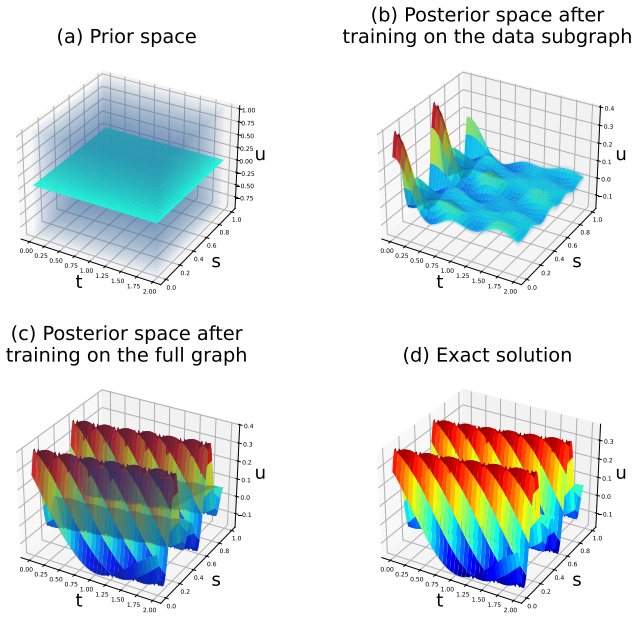

Figure 5: Solution spaces for the advection equation ($b = 0.1$) for prior and posterior parameters shown as predictive mean (colorful) $\pm\sigma$ (gray shade). **(a)** prior space before any training; **(b)** posterior space with parameters trained only on initial condition - the function fits perfectly for $t = 0$, but does not learn any time dependencies and already narrows the space too much; **(c)** posterior space with parameters trained with a mixed schedule of initial condition data passes and PDE passes - the function learns both spatial and time patterns correctly; **(d)** exact solution of the PDE.

data observation or zero-residual constraint; and (3) a backward update that propagates the resulting information to the weights and replaces the factor's previously cached messages.

The randomized schedule is motivated in Section 3.3. The belief of each weight W_k combines the prior, data, and PDE messages as

$$\begin{aligned} q(W_k) &\propto q^{(0)}(W_k)\, q^{\mathcal{D}}(W_k)\, q^{\mathcal{C}}(W_k) \\ &\propto \exp\left\{-\frac{1}{2}\left(\kappa_k^{(0)} + \kappa_k^{\mathcal{D}} + \kappa_k^{\mathcal{C}}\right) W_k^2 + \left(\tau_k^{(0)} + \tau_k^{\mathcal{D}} + \tau_k^{\mathcal{C}}\right) W_k\right\}, \end{aligned}$$

where $(\kappa_k^{(0)}, \tau_k^{(0)})$, $(\kappa_k^{\mathcal{D}}, \tau_k^{\mathcal{D}})$, and $(\kappa_k^{\mathcal{C}}, \tau_k^{\mathcal{C}})$ denote the natural parameters contributed by the prior, data subgraph, and PDE subgraph, respectively. Updating either subgraph separately may lead to a fixed point that satisfies only the data constraints or only the PDE constraints, while substantially narrowing the weight beliefs before the other subgraph is considered. Interleaving individual data and PDE-residual factors allows both sources of information to influence the beliefs throughout inference. Our ablation study illustrates this behavior in Figure 5: converging the data subgraph separately produces overly concentrated intermediate beliefs, whereas the interleaved schedule empirically reaches a fixed point that reflects both the observed data and the PDE constraints. This observation is empirical and does not constitute a general convergence guarantee for loopy message passing.

A.2 Theoretical properties

Our target distribution is the Bayesian posterior in Equation 1. For a nonlinear neural network, the messages induced by the data and PDE factors are generally non-Gaussian; therefore, posteriors computed by our algorithm are not exact. Instead, each message is moment-matched to a Gaussian, producing *an approximate belief* $q(\boldsymbol{\theta})$. After this convergence, these beliefs are not guaranteed to match the exact posterior, i.e., $q^\star(\boldsymbol{\theta}) \not\equiv p(\boldsymbol{\theta} \mid \mathcal{D}, \mathcal{C})$ - the equality holds only if all factors

Algorithm 1 Randomized message passing schedule on the PDE factor graph

Require: Factors $\mathcal{F} = \mathcal{F}_{\mathcal{D}} \cup \mathcal{F}_{\mathcal{C}}$, prior beliefs $q^{(0)}(W_k)$, maximum number of epochs $E_{\max}$ (default: 300), and convergence tolerance δ (default: 10^{-5})

Initialize every cached factor-to-weight message with zero natural parameters:
$\big(\kappa_{i\to k}, \tau_{i\to k}\big) \leftarrow (0,0)$ for every $f_i \in \mathcal{F}$ and $W_k \in \mathrm{ne}(f_i)$
Initialize the weight beliefs: $q(W_k) \leftarrow q^{(0)}(W_k)$ for every W_k
for $e = 1, \ldots, E_{\max}$ **do**
  Draw a fresh random permutation π_e of $\mathcal{F}$
  $\Delta_e \leftarrow 0$
  for f_i in the order π_e **do**
    for $W_k \in \mathrm{ne}(f_i)$ **do**
      Extract the factor-excluded incoming message as given in Equation 7:
$$m_{W_k \to f_i}(W_k) \propto \frac{q(W_k)}{m_{f_i \to W_k}(W_k)}$$
    end for
    Set the forward inputs to $\bar{m}_{ik}(W_k) \leftarrow m_{W_k \to f_i}(W_k)$ for every $W_k \in \mathrm{ne}(f_i)$
    (1) Forward update: propagate $\{\bar{m}_{ik}\}_{W_k \in \mathrm{ne}(f_i)}$ through f_i and moment-match the resulting output message $m_{f_i \to F_i}$ if $f_i \in \mathcal{F}_{\mathcal{D}}$, or $m_{f_i \to R_i}$ if $f_i \in \mathcal{F}_{\mathcal{C}}$
    (2) Observation update: combine the forward message with the corresponding Gaussian observation – observed data point y_i for a data factor of zero-mean residual for a PDE residual factor:
$$q(F_i) \propto m_{f_i \to F_i}(F_i)\, \mathcal{N}\big(F_i; y_i, \epsilon_D^2\big), \qquad f_i \in \mathcal{F}_{\mathcal{D}},$$
$$q(R_i) \propto m_{f_i \to R_i}(R_i)\, \mathcal{N}\big(R_i; 0, \epsilon_C^2\big), \qquad f_i \in \mathcal{F}_{\mathcal{C}}.$$
    (3) Backward update: use the updated output belief $q(F_i)$ or $q(R_i)$ to compute the Gaussian messages $\widetilde{m}_{f_i \to W_k}$ for every $W_k \in \mathrm{ne}(f_i)$
    for $W_k \in \mathrm{ne}(f_i)$ **do**
      $q_{\mathrm{old}}(W_k) \leftarrow q(W_k)$
      Replace the cached message and update the weight belief:
$$q(W_k) \leftarrow m_{W_k \to f_i}(W_k)\, \widetilde{m}_{f_i \to W_k}(W_k)$$
      $m_{f_i \to W_k} \leftarrow \widetilde{m}_{f_i \to W_k}$
      Update the maximum belief change:
$$\Delta_e \leftarrow \max\{\Delta_e,\, D_{\mathrm{KL}}(q_{\mathrm{old}}(W_k) \,\|\, q(W_k))\}$$
    end for
  end for
  if $\Delta_e < \delta$ **then**
    break
  end if
end for
return $\{q(W_k)\}_k$

produce only Gaussians, and messages therefore can be computed exactly. We therefore evaluate the resulting predictive mean and uncertainty empirically; refer to Section 4.2 for this evaluation. We also empirically observe that interleaving data and PDE updates is important for obtaining solutions that satisfy both sources of information; this behavior is not claimed as a convergence guarantee.

A.3 Differentiable and decorrelated basis

Inspired by Fourier series, for each coordinate x_i, $i = 1, \ldots, d$, we define

$$\psi_i(x_i) = \Big[1,\ \big(\sin(2\pi k x_i), \cos(2\pi k x_i)\big)_{k=1}^{K_i}\Big],$$

where K_i determines the number of frequencies. We combine these coordinate-wise bases using the Kronecker product, $\phi(\boldsymbol{x}) = \bigotimes_{i=1}^{d} \psi_i(x_i)$, which produces all combinations of the coordinate-wise basis functions.

In the factor graph, the basis enters inference only through the factors corresponding to the data observations and PDE constraints. If different basis components behave similarly at the data and PDE

points, their corresponding weights are not individually well-identified, and hence the posterior exhibits strong correlations among these weights. These correlations could be represented explicitly by maintaining a full multivariate Gaussian distribution over the weights; however, this would substantially increase the algorithm's cost. Instead, inspired by the principal component analysis (PCA) decorrelation transform (Kessy et al., 2018), we *rotate the basis* so that components with similar effects are replaced by new components that more distinctly affect the constraints, reducing correlations between their corresponding weights. We evaluate the basis components entering the data and PDE constraints as for advection, with $\boldsymbol{r}_{\mathrm{adv}}(\boldsymbol{x}) := \sum_{i=1}^{d} b_i \, \partial_{x_i} \phi(\boldsymbol{x})$,

$$\boldsymbol{A}_{\mathcal{D}} = \begin{bmatrix} \phi(\boldsymbol{x}_1)^\top \\ \vdots \\ \phi(\boldsymbol{x}_N)^\top \end{bmatrix}, \qquad \boldsymbol{A}_{\mathcal{C}} = \begin{bmatrix} \boldsymbol{r}_{\mathrm{adv}}(\boldsymbol{x}_1)^\top \\ \vdots \\ \boldsymbol{r}_{\mathrm{adv}}(\boldsymbol{x}_M)^\top \end{bmatrix}.$$

We aim to evaluate the similarities of the basis components. We construct a weighted Gram matrix that contains non-normalized cosine similarities between the components:

$$\boldsymbol{G} = \frac{1}{\epsilon_D^2} \boldsymbol{A}_{\mathcal{D}}^\top \boldsymbol{A}_{\mathcal{D}} + \frac{1}{\epsilon_C^2} \boldsymbol{A}_{\mathcal{C}}^\top \boldsymbol{A}_{\mathcal{C}}.$$

We compute the eigenvalue decomposition of this symmetric matrix as $\boldsymbol{G} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^\top$ The eigenvectors define a rotation of the original basis, for which $\boldsymbol{V}^\top \boldsymbol{G} \boldsymbol{V} = \boldsymbol{\Lambda}$. Since $\boldsymbol{\Lambda}$ is diagonal, the cosine similarity between any two distinct rotated components is zero. Let $\widetilde{\boldsymbol{V}}$ contain the selected eigenvectors, then we redefine the basis as $\widetilde{\phi}(\boldsymbol{x}) = \widetilde{\boldsymbol{V}}^\top \phi(\boldsymbol{x})$ and, correspondingly, $\widetilde{\boldsymbol{r}}_{\mathrm{adv}}(\boldsymbol{x}) = \widetilde{\boldsymbol{V}}^\top \boldsymbol{r}_{\mathrm{adv}}(\boldsymbol{x})$. The corresponding modification for the nonlinear Fisher-KPP residual is given in Appendix A.3.1.

A.3.1 Adaptation to the Fisher-KPP equation

For the Fisher-KPP equation (Appendix A.5.2), the data matrix $\boldsymbol{A}_{\mathcal{D}}$ is unchanged. Its residual, however, is nonlinear in the weights: the reaction term $-\rho f(1-f)$ and the quadratic term arising from the chain rule (Appendix A.5.2) define no weight-independent basis components. We therefore build $\boldsymbol{A}_{\mathcal{C}}$ from the linear part of the operator,

$$\boldsymbol{r}(\boldsymbol{x}) = \frac{\partial \phi(\boldsymbol{x})}{\partial t} - \nu \, \frac{\partial^2 \phi(\boldsymbol{x})}{\partial s^2},$$

which dominates the correlations the rotation is meant to remove: the diffusion term acts on spatial frequency k as $\nu(2\pi k)^2$, while the reaction term is bounded. These linearized rows enter only $\boldsymbol{G}$, i.e., only the choice of basis; the residual function f_R itself uses the full nonlinear expression of Equation 12 in the rotated basis $\widetilde{\phi}$.

A.4 Derivations for the activation function

This section derives the forward and backward messages for the activation function $g(\cdot)$ in Equation 4 and for its derivative. Because the activation is piecewise exponential, the required moments reduce to conditional log-normal moments. Lemma 1 gives these moments. Appendix A.4.1 applies them to g, while Appendix A.4.2 treats g'.

Lemma 1. *Conditional moments of the lognormal distribution can be computed exactly. Let $X \sim \mathcal{N}(\cdot; \mu_X, \sigma_X^2)$. Then, for any $\lambda \in \mathbb{R}$,*

$$\mathbb{E}\left[e^{\lambda X} \mathbf{1}_{\{X>0\}}\right] = e^{\lambda \mu_X + \frac{1}{2}\lambda^2 \sigma_X^2} \, \Phi\left(\frac{\mu_X + \lambda \sigma_X^2}{\sigma_X}\right),$$

$$\mathbb{E}\left[e^{\lambda X} \mathbf{1}_{\{X\leq 0\}}\right] = e^{\lambda \mu_X + \frac{1}{2}\lambda^2 \sigma_X^2} \, \Phi\left(-\frac{\mu_X + \lambda \sigma_X^2}{\sigma_X}\right).$$

Proof. By definition, the expectation of a function of X is the integral of that function with respect to the density of X. Since the indicator functions restrict X to $X > 0$ or $X \leq 0$, respectively, the corresponding integration domains are $(0, \infty)$ and $(-\infty, 0]$. We can therefore rewrite the required expressions as

$$
\begin{aligned}
\mathbb{E}\big[e^{\lambda X}\mathbf{1}_{\{X>0\}}\big] &= \frac{1}{\sigma_X\sqrt{2\pi}}\int_0^\infty \exp\!\left(\lambda x - \frac{(x-\mu_X)^2}{2\sigma_X^2}\right)dx \\
&= e^{\lambda\mu_X+\frac{1}{2}\lambda^2\sigma_X^2}\,\frac{1}{\sigma_X\sqrt{2\pi}}\int_0^\infty \exp\!\left(-\frac{(x-\mu_X-\lambda\sigma_X^2)^2}{2\sigma_X^2}\right)dx \\
&= e^{\lambda\mu_X+\frac{1}{2}\lambda^2\sigma_X^2}\,\Phi\!\left(\frac{\mu_X+\lambda\sigma_X^2}{\sigma_X}\right), \\
\mathbb{E}\big[e^{\lambda X}\mathbf{1}_{\{X\le 0\}}\big] &= \frac{1}{\sigma_X\sqrt{2\pi}}\int_{-\infty}^0 \exp\!\left(\lambda x - \frac{(x-\mu_X)^2}{2\sigma_X^2}\right)dx \\
&= e^{\lambda\mu_X+\frac{1}{2}\lambda^2\sigma_X^2}\,\frac{1}{\sigma_X\sqrt{2\pi}}\int_{-\infty}^0 \exp\!\left(-\frac{(x-\mu_X-\lambda\sigma_X^2)^2}{2\sigma_X^2}\right)dx \\
&= e^{\lambda\mu_X+\frac{1}{2}\lambda^2\sigma_X^2}\,\Phi\!\left(-\frac{\mu_X+\lambda\sigma_X^2}{\sigma_X}\right).
\end{aligned}
$$

□

A.4.1 ACTIVATION FUNCTION MESSAGES

Forward message For the defined $g(x)$, the forward moments follow directly from Lemma 1:

$$
\begin{aligned}
\mathbb{E}[g(X)] &= \mathbb{E}\big[e^{\beta X}\mathbf{1}_{\{X>0\}}\big] - \mathbb{P}(X>0) - \alpha\left(\mathbb{E}\big[e^{-\beta X}\mathbf{1}_{\{X\le 0\}}\big] - \mathbb{P}(X\le 0)\right), \\
\mathbb{E}[g(X)^2] &= \mathbb{E}\big[e^{2\beta X}\mathbf{1}_{\{X>0\}}\big] - 2\mathbb{E}\big[e^{\beta X}\mathbf{1}_{\{X>0\}}\big] + \mathbb{P}(X>0) \\
&\quad + \alpha^2\left(\mathbb{E}\big[e^{-2\beta X}\mathbf{1}_{\{X\le 0\}}\big] - 2\mathbb{E}\big[e^{-\beta X}\mathbf{1}_{\{X\le 0\}}\big] + \mathbb{P}(X\le 0)\right).
\end{aligned}
$$

Backward message Given $\alpha > 0$ and $\beta > 0$ in Equation 4, the sign of $Y = g(X)$ uniquely identifies the corresponding branch: $Y > 0$ implies $X > 0$, while $Y \le 0$ implies $X \le 0$. Since each branch is strictly monotonic, $g^{-1}(Y)$ has a closed form:

$$
g^{-1}(Y) = \begin{cases} \dfrac{1}{\beta}\log(1+Y), & Y > 0, \\ -\dfrac{1}{\beta}\log\!\left(1-\dfrac{Y}{\alpha}\right), & Y \le 0. \end{cases} \tag{8}
$$

For any positive random variable Z with mean μ_Z and variance σ_Z^2, we approximate Z by a log-normal distribution with matching first two moments, i.e., $Z \sim \mathrm{LogNormal}\Big(Z; \mu_{\log Z}, \sigma^2_{\log Z}\Big)$. By definition of the log-normal distribution, $\mu_Z = \exp\Big(\mu_{\log Z} + \frac{1}{2}\sigma^2_{\log Z}\Big)$ and $\sigma_Z^2 = \Big(\exp(\sigma^2_{\log Z}) - 1\Big)\exp\Big(2\mu_{\log Z} + \sigma^2_{\log Z}\Big)$. Solving for $\mu_{\log Z}$ and $\sigma^2_{\log Z}$, we get $\sigma^2_{\log Z} = \log\Big(1 + \frac{\sigma_Z^2}{\mu_Z^2}\Big)$ and $\mu_{\log Z} = \log(\mu_Z) - \frac{1}{2}\sigma^2_{\log Z}$. Therefore, we have $\log(Z) \sim \mathcal{N}\Big(\cdot; \mu_{\log Z}, \sigma^2_{\log Z}\Big)$ under the log-normal approximation. We apply this property to get the moments of $\log(1+Y)$ and $\log\big(1-\frac{Y}{\alpha}\big)$ and compute final moments of the function from Equation 8.

A.4.2 ACTIVATION DERIVATIVE MESSAGES

The derivative of $g(z)$ is defined as

$$
g'(z) = \begin{cases} \beta e^{\beta z}, & z > 0, \\ \alpha\beta e^{-\beta z}, & z \le 0. \end{cases}
$$

Forward message Using again Lemma 1, the forward moments are computed directly as

$$
\begin{aligned}
\mathbb{E}[g'(Z)] &= \beta\mathbb{E}\big[e^{\beta Z}\mathbf{1}_{\{Z>0\}}\big] + \alpha\beta\mathbb{E}\big[e^{-\beta Z}\mathbf{1}_{\{Z\le 0\}}\big], \\
\mathbb{E}[g'(Z)^2] &= \beta^2\mathbb{E}\big[e^{2\beta Z}\mathbf{1}_{\{Z>0\}}\big] + \alpha^2\beta^2\mathbb{E}\big[e^{-2\beta Z}\mathbf{1}_{\{Z\le 0\}}\big].
\end{aligned}
$$

Backward message Given $g'(Z) = D$, for the positive branch, $Z > 0$ implies $D > \beta$, while for the negative branch, $Z \leq 0$ implies $D \geq \alpha\beta$. Solving each branch for Z gives

$$(g'_+)^{-1}(D) = \frac{1}{\beta}\log\left(\frac{D}{\beta}\right), \qquad D > \beta,$$

$$(g'_-)^{-1}(D) = -\frac{1}{\beta}\log\left(\frac{D}{\alpha\beta}\right), \quad D \geq \alpha\beta.$$

Unlike the forward transformation, D does not uniquely identify the branch, since its ranges may overlap. For this reason, the residual-factor updates are derived without requiring an inverse of g'; see Appendices A.5.1 and A.5.2.

A.5 Derivations for the residual factor

In this section, we show the derivations of the residual factor f_R for the advection equation (Appendix A.5.1) and the Fisher-KPP equation (Appendix A.5.2). All derivations are performed for the shallow neural architecture used throughout this work, defined in Section 3.

A.5.1 Advection equation

The advection equation has a generic form given by Equation 3. The physical meaning of the advection equation is the transport of a quantity by a given velocity field. For the shallow architecture defined in Section 3, the chain rule gives

$$f'_R(\boldsymbol{W}, \boldsymbol{x}) = \boldsymbol{b}^\top \nabla_{\boldsymbol{x}} f_U(\boldsymbol{x}; \boldsymbol{\theta}) = g'(z)\boldsymbol{W}^\top \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i}, \qquad z = \boldsymbol{W}^\top \phi(\boldsymbol{x}).$$

Forward message Starting from Equation 5, we compute the residual expectation as

$$\mathbb{E}[c(\boldsymbol{W}, \boldsymbol{x})] = \mathbb{E}[g'(z)\boldsymbol{W}^\top] \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i} = \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \cdot \left(\mathbb{E}[g'(z)]\mathbb{E}[\boldsymbol{W}^\top] + \mathrm{Cov}(g'(z), \boldsymbol{W}^\top)\right).$$

The covariance term $\mathrm{Cov}(g'(z), \boldsymbol{W}^\top)$ is computed as follows. Stein's lemma states that for two jointly Gaussian variables η_1 and η_2 and a function $\chi(.)$, the following equality holds:

$$\mathrm{Cov}(\eta_1, \chi(\eta_2)) = \mathrm{Cov}(\eta_1, \eta_2)\mathbb{E}[\chi'(\eta_2)]. \tag{9}$$

Since W_k and z are jointly Gaussian, we set $\eta_1 = W_k$, $\eta_2 = z$, and replace χ by our g'. Hence,

$$\mathrm{Cov}(W_k, g'(z)) = \mathrm{Cov}(W_k, z)\mathbb{E}[g''(z)] = \mathrm{Cov}\left(W_k, \sum_j W_j \phi_j(\boldsymbol{x})\right)\mathbb{E}[g''(z)]$$

$$= \sum_{j=1}^{P} \phi_j(\boldsymbol{x})\,\mathrm{Cov}(W_k, W_j)\mathbb{E}[g''(z)] = \phi_k(\boldsymbol{x})\sigma_k^2\mathbb{E}[g''(z)].$$

As $c(\boldsymbol{W}, \boldsymbol{x})$ is a non-linear function of correlated Gaussians, it does not have a closed-form variance expression, and we use the first-order Taylor expansion for $c(\boldsymbol{W}, \boldsymbol{x})$ to get the variance approximation. The generic formula for this expansion is

$$F(X, Y) \approx F\big(\mathbb{E}[X], \mathbb{E}[Y]\big) + \left.\frac{\partial F}{\partial X}\right|_{(\mathbb{E}[X], \mathbb{E}[Y])} (X - \mathbb{E}[X]) + \left.\frac{\partial F}{\partial Y}\right|_{(\mathbb{E}[X], \mathbb{E}[Y])} (Y - \mathbb{E}[Y]). \tag{10}$$

For simplicity, let $r(\boldsymbol{x}) := \boldsymbol{W}^\top \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i}$. Then, $c(\boldsymbol{W}, \boldsymbol{x}) = g'(z)r(\boldsymbol{x})$ and we can approximate it as

$$
\begin{aligned}
c(\boldsymbol{W}, \boldsymbol{x}) &= g'(z) r(\boldsymbol{x}) \\
&\approx g'(\mathbb{E}[z])\mathbb{E}[r(\boldsymbol{x})] + g''(\mathbb{E}[z])\mathbb{E}[r(\boldsymbol{x})]\,(z - \mathbb{E}[z]) \\
&\quad + g'(\mathbb{E}[z])\,(r(\boldsymbol{x}) - \mathbb{E}[r(\boldsymbol{x})]) \\
&= g'(\mathbb{E}[z]) r(\boldsymbol{x}) + g''(\mathbb{E}[z])\mathbb{E}[r(\boldsymbol{x})]\,(z - \mathbb{E}[z])\,.
\end{aligned} \tag{11}
$$

Thus, the second moment approximation is

$$
\begin{aligned}
\mathbb{E}[(c(\boldsymbol{W}, \boldsymbol{x}))^2] \approx\ & g'(\mathbb{E}[z])^2 \mathbb{E}[r(\boldsymbol{x})^2] \\
& + \mathbb{E}[r(\boldsymbol{x})]^2 g''(\mathbb{E}[z])^2 \operatorname{Var}(z) \\
& + 2g'(\mathbb{E}[z])\mathbb{E}[r(\boldsymbol{x})] g''(\mathbb{E}[z]) \operatorname{Cov}(r(\boldsymbol{x}), z),
\end{aligned}
$$

where the covariance term can be computed straightforwardly.

Backward message In the backward update, we send a message to each W_k. Since $z = \boldsymbol{W}^\top \phi(\boldsymbol{x}) = \sum_j W_j [\phi(\boldsymbol{x})]_j$ depends on W_k, the residual is nonlinear in W_k and cannot be solved for W_k directly; we therefore use the same first-order approximation as in the forward direction, defined in Equation 11. We solve Equation 10 for W_k, assuming all moments and derivatives are already evaluated and therefore constitute fixed coefficients. Then, the coefficient before W_k is

$$
A_k = \left(\sum_{i=1}^{d} b_i \left[\frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right]_k \right) g'(\mathbb{E}[z]) + [\phi(\boldsymbol{x})]_k \mathbb{E}[r(\boldsymbol{x})] g''(\mathbb{E}[z]).
$$

Then, collecting all other terms, we can express W_k as

$$
\begin{aligned}
W_k \approx \frac{1}{A_k} \Bigg[& R - g'(\mathbb{E}[z]) \sum_{j \neq k} \left(\sum_{i=1}^{d} b_i \left[\frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right]_j \right) W_j \\
& - \mathbb{E}\left[\boldsymbol{W}^\top \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right] g''(\mathbb{E}[z]) \left(\sum_{j \neq k} [\phi(\boldsymbol{x})]_j W_j - \mathbb{E}[z] \right) \Bigg].
\end{aligned}
$$

Given that the incoming residual message is independent of the incoming weights and $\mathbb{E}[R] = 0$, moment matching gives the following results:

$$
\begin{aligned}
\mathbb{E}[W_k] \approx \frac{1}{A_k} \Bigg[& - g'(\mathbb{E}[z]) \sum_{j \neq k} \left(\sum_{i=1}^{d} b_i \left[\frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right]_j \right) \mu_j \\
& - \mathbb{E}\left[\boldsymbol{W}^\top \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right] g''(\mathbb{E}[z]) \left(\sum_{j \neq k} [\phi(\boldsymbol{x})]_j \mu_j - \mathbb{E}[z] \right) \Bigg], \\
\operatorname{Var}(W_k) \approx \frac{1}{A_k^2} \Bigg[& \epsilon_C^2 + g'(\mathbb{E}[z])^2 \sum_{j \neq k} \left(\sum_{i=1}^{d} b_i \left[\frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right]_j \right)^2 \sigma_j^2 \\
& + \mathbb{E}\left[\boldsymbol{W}^\top \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right]^2 g''(\mathbb{E}[z])^2 \sum_{j \neq k} [\phi(\boldsymbol{x})]_j^2 \sigma_j^2 \\
& + 2g'(\mathbb{E}[z]) \mathbb{E}\left[\boldsymbol{W}^\top \sum_{i=1}^{d} b_i \frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right] g''(\mathbb{E}[z]) \sum_{j \neq k} \left(\sum_{i=1}^{d} b_i \left[\frac{\partial \phi(\boldsymbol{x})}{\partial x_i} \right]_j \right) [\phi(\boldsymbol{x})]_j \sigma_j^2 \Bigg].
\end{aligned}
$$

A.5.2 FISHER-KPP EQUATION

The Fisher-KPP equation is a reaction–diffusion PDE that extends the idea of transport and describes how a quantity spreads through diffusion while also growing locally. It is expressed through the following PDE:

$$f'_R(t,s) := \frac{\partial f}{\partial t} - \nu \frac{\partial^2 f}{\partial s^2} - \rho f(1-f), \tag{12}$$

where $\nu > 0$ is a given diffusion coefficient, $\rho > 0$ is the logistic growth rate, and $f = f_U(\boldsymbol{x}; \boldsymbol{\theta})$ denotes the inferred population density.

Forward message The expectation follows by taking the expectation of each term in Equation 12:

$$\begin{aligned}\mathbb{E}[c(t,s)] = \mathbb{E}\left[g'(z)\boldsymbol{W}^\top \frac{\partial \phi}{\partial t}\right] - \nu \mathbb{E}\left[g'(z)\boldsymbol{W}^\top \frac{\partial^2 \phi}{\partial s^2}\right] \\ - \nu \mathbb{E}\left[g''(z)\left(\boldsymbol{W}^\top \frac{\partial \phi}{\partial s}\right)^2\right] - \rho \mathbb{E}[g(z)] + \rho \mathbb{E}[g(z)^2].\end{aligned}$$

Using Stein's lemma from Equation 9, for any deterministic vector $\boldsymbol{a}$,

$$\begin{aligned}\mathbb{E}[g'(z)\boldsymbol{W}^\top \boldsymbol{a}] &= \sum_k a_k \, \mathbb{E}[W_k g'(z)] \\ &= \sum_k a_k \left(\mathbb{E}[W_k]\mathbb{E}[g'(z)] + \mathrm{Cov}(W_k, g'(z))\right) \\ &= \sum_k a_k \left(\mathbb{E}[W_k]\mathbb{E}[g'(z)] + \phi_k(\boldsymbol{x}) \, \mathrm{Var}(W_k)\mathbb{E}[g''(z)]\right).\end{aligned}$$

Substituting $\frac{\partial \phi}{\partial t}$ and $\frac{\partial^2 \phi}{\partial s^2}$ for $\boldsymbol{a}$, we can compute the first two terms. For the next quadratic derivative term, let

$$\zeta_s = \boldsymbol{W}^\top \frac{\partial \phi}{\partial s}.$$

Its first two moments are

$$\mathbb{E}[\zeta_s] = \sum_k \mathbb{E}[W_k] \frac{\partial \phi_k}{\partial s}, \qquad \mathbb{E}[\zeta_s^2] = \mathbb{E}[\zeta_s]^2 + \sum_k \mathrm{Var}(W_k) \left(\frac{\partial \phi_k}{\partial s}\right)^2.$$

Since z and ζ_s depend on the same weights, they are generally correlated. For simplicity, we introduce an additional moment-factorization approximation by neglecting the dependence between $g''(z)$ and ζ_s^2:

$$\mathbb{E}[g''(z)\zeta_s^2] \approx \mathbb{E}[g''(z)] \, \mathbb{E}[\zeta_s^2].$$

Thus, we approximate the residual mean as

$$\begin{aligned}\mathbb{E}[c(t,s)] \approx \sum_k \frac{\partial \phi_k}{\partial t} \left(\mathbb{E}[W_k]\mathbb{E}[g'(z)] + \phi_k \, \mathrm{Var}(W_k)\mathbb{E}[g''(z)]\right) \\ - \nu \sum_k \frac{\partial^2 \phi_k}{\partial s^2} \left(\mathbb{E}[W_k]\mathbb{E}[g'(z)] + \phi_k \, \mathrm{Var}(W_k)\mathbb{E}[g''(z)]\right) \\ - \nu \mathbb{E}[g''(z)] \left[\left(\sum_k \mathbb{E}[W_k] \frac{\partial \phi_k}{\partial s}\right)^2 + \sum_k \mathrm{Var}(W_k) \left(\frac{\partial \phi_k}{\partial s}\right)^2\right] \\ - \rho \mathbb{E}[g(z)] + \rho \mathbb{E}[g(z)^2].\end{aligned}$$

To approximate the variance, as in the derivations for the advection equation in Section A.5.1, we use a first-order approximation of the complete residual around the expected weights, as in Equation 10. Writing $R = f_R(\boldsymbol{x}, \boldsymbol{W})$,

$$R \approx R(\mathbb{E}[\boldsymbol{W}]) + \sum_k \left.\frac{\partial R}{\partial W_k}\right|_{\boldsymbol{W} = \mathbb{E}[\boldsymbol{W}]} \left(W_k - \mathbb{E}[W_k]\right).$$

Since the weights are represented by independent Gaussian messages,

$$\mathrm{Var}(R) \approx \sum_k \left(\frac{\partial R}{\partial W_k}\bigg|_{\boldsymbol{W}=\mathbb{E}[\boldsymbol{W}]} \right)^2 \mathrm{Var}(W_k).$$

Differentiating the residual gives

$$\begin{aligned}\frac{\partial R}{\partial W_k} = g'(z)\frac{\partial \phi_k}{\partial t} + g''(z)\phi_k \boldsymbol{W}^\top \frac{\partial \phi}{\partial t} \\ - \nu g'(z)\frac{\partial^2 \phi_k}{\partial s^2} - \nu g''(z)\phi_k \boldsymbol{W}^\top \frac{\partial^2 \phi}{\partial s^2} \\ - \nu g'''(z)\phi_k \left(\boldsymbol{W}^\top \frac{\partial \phi}{\partial s}\right)^2 \\ - 2\nu g''(z)\left(\boldsymbol{W}^\top \frac{\partial \phi}{\partial s}\right)\frac{\partial \phi_k}{\partial s} \\ + \rho \phi_k g'(z)\big(2g(z)-1\big).\end{aligned}$$

The moments of $g(z)$ and $g'(z)$ are given in Appendices A.4.1 and A.4.2, respectively; the computation of $g''(z)$ and $g'''(z)$ changes only by the corresponding constant factors.

Backward message In the backward update, we apply the same local linearization used for the advection equation in Appendix A.5.1. The coefficient A_k is defined by evaluating the residual derivative at the expected weights:

$$A_k = \frac{\partial R}{\partial W_k}\bigg|_{\boldsymbol{W}=\mathbb{E}[\boldsymbol{W}]}.$$

For notation simplicity, define

$$\begin{aligned}\mathbb{E}[z] = \mathbb{E}[\boldsymbol{W}]^\top \phi, \quad \mathbb{E}[\zeta_t] = \mathbb{E}[\boldsymbol{W}]^\top \frac{\partial \phi}{\partial t}, \\ \mathbb{E}[\zeta_{ss}] = \mathbb{E}[\boldsymbol{W}]^\top \frac{\partial^2 \phi}{\partial s^2}, \quad \mathbb{E}[\zeta_s] = \mathbb{E}[\boldsymbol{W}]^\top \frac{\partial \phi}{\partial s}.\end{aligned}$$

Then, we can expand A_k as

$$\begin{aligned}A_k = g'(\mathbb{E}[z])\frac{\partial \phi_k}{\partial t} + g''(\mathbb{E}[z])\phi_k \mathbb{E}[\zeta_t] \\ - \nu g'(\mathbb{E}[z])\frac{\partial^2 \phi_k}{\partial s^2} - \nu g''(\mathbb{E}[z])\phi_k \mathbb{E}[\zeta_{ss}] \\ - \nu g'''(\mathbb{E}[z])\phi_k \mathbb{E}[\zeta_s]^2 - 2\nu g''(\mathbb{E}[z])\mathbb{E}[\zeta_s]\frac{\partial \phi_k}{\partial s} \\ + \rho \phi_k g'(\mathbb{E}[z])\left(2g(\mathbb{E}[z])-1\right).\end{aligned}$$

Then, W_k can be approximated as

$$W_k \approx \mathbb{E}[W_k] + \frac{1}{A_k}\left[R - \mathbb{E}[R] - \sum_{j\neq k} A_j \left(W_j - \mathbb{E}[W_j]\right)\right].$$

Given target $R \sim \mathcal{N}(\cdot; 0, \epsilon_C^2)$, we substitute $\mathbb{E}[R]$ and $\mathrm{Var}(R)$ accordingly and moment match the W_k approximation in the same way as for the backward message for the advection equation in Appendix A.5.1.

A.6 EXPERIMENTAL DETAILS

Data Trajectories are drawn once from a fixed-seed permutation of each PDEBench (Takamoto et al., 2022) setting; the trajectory used during development is excluded. Advection uses ten trajectories across five speeds each; the time cost of these experiments is mostly dominated by the HMC and VI references. Fisher-KPP uses five trajectories in each of its 16 settings. Its stored trajectories freeze at non-physical values once the field homogenizes (a float32 artifact of the dataset generator), so Fisher-KPP errors are measured against a float64 re-solve of the PDE from the same initial condition.

Shared setup We use 1024 labeled points at $t = 0$ and $48K_t$ stratified collocation points, where K_t is the number of temporal Fourier frequencies in the basis. For each trajectory, all methods share the basis, rotation, factor locations, and seeds, thereby targeting the same posterior. Message passing visits the factors in a shuffled order each epoch and stops when the largest per-factor Kullback–Leibler (KL) divergence falls below 10^{-5}, or after 300 epochs.

Uncertainty metrics Each method yields a Gaussian predictive $\mathcal{N}(\cdot; \mu_j, \sigma_j^2)$ of the network output at every point (x_j, t_j) of the evaluation grid: our marginal moments, or the sample mean and variance over posterior draws for HMC and VI. With the exact solution u_j and $z_j = (u_j - \mu_j)/\sigma_j$, the coverage of the nominal 90 % interval and the closed-form Gaussian CRPS (Gneiting & Raftery, 2007) are

$$\mathrm{Cov}_{90} = \frac{1}{G}\sum_{j=1}^{G} \mathbb{1}\left[\,|z_j| \le \Phi^{-1}(0.95)\right], \quad \mathrm{CRPS} = \frac{1}{G}\sum_{j=1}^{G} \sigma_j\Big[z_j\big(2\Phi(z_j) - 1\big) + 2\varphi(z_j) - \tfrac{1}{\sqrt{\pi}}\Big],$$

where Φ and φ are the standard normal cumulative density function and density, $\Phi^{-1}(0.95) \approx 1.645$, and the averages run over the G grid points of all runs. For a point prediction ($\sigma_j \to 0$) CRPS reduces to the absolute error $|u_j - \mu_j|$.

Reaction continuation (Fisher-KPP) The logistic reaction adds a spurious posterior mode at $f \equiv 0$, which message passing and gradient-based training reach from their default initializations at a strong reaction. This is a common failure mode for learning PDEs Krishnapriyan et al. (2021). All methods therefore train in four passes at reaction coefficient $\lambda\rho$, $\lambda \in \{0.25, 0.5, 0.75, 1\}$, each warm-started from the previous one, within unchanged iteration budgets. This roughly doubles message passing training time; reported times include it.

Baselines The SGD baseline, VI and the HMC warm start use Adam (learning rate 3×10^{-4}, at most $30{,}000$ iterations, split evenly over the passes on Fisher-KPP), halving the rate on loss plateaus (less than $0.1\,\%$ improvement over 500 iterations) up to four times and stopping at the next plateau. VI is mean-field Gaussian with eight reparameterized samples per iteration.

HMC reference Four chains of $1{,}000$ iterations use a dense Gauss–Newton mass matrix at the maximum a posteriori (MAP) estimate (the polished Adam warm start; on Fisher-KPP, the better of the warm starts from zero and from the message passing mean). The step size is adapted by dual averaging to acceptance 0.8 over 250 discarded warm-up iterations, and the number of leapfrog steps per proposal is drawn uniformly from $\{1, \ldots, 20\}$. Every coordinate of all 50 advection and 80 Fisher-KPP reference runs reaches split-$\hat{R} < 1.05$ and bulk effective sample size > 100.

Width ablation For this ablation, the wider model is $f_H(\boldsymbol{x}) = \sum_{h=1}^{H} a_h g(z_h)$ with $z_h = \boldsymbol{W}_h^\top \phi(\boldsymbol{x})$ and fixed, distinct readout weights a_h. Models with $H \in \{2, 3, 4\}$ hidden units keep everything else of the single-unit model. They run on all 50 advection runs and, on Fisher-KPP, on all 80 runs at $H = 2$ and on 33 ($H = 3$) and 19 ($H = 4$) runs covering all 16 settings.

Hardware and timing All experiments ran on one 8-core AMD EPYC (Genoa) central processing unit (CPU) with 16 GB random-access memory (RAM) and no graphics processing unit (GPU) (Julia 1.12.6). Reported training times are cold-start and include just-in-time compilation; steady-state inference times are the minimum over seven repetitions (three for HMC) after a warm-up call on the otherwise idle machine.

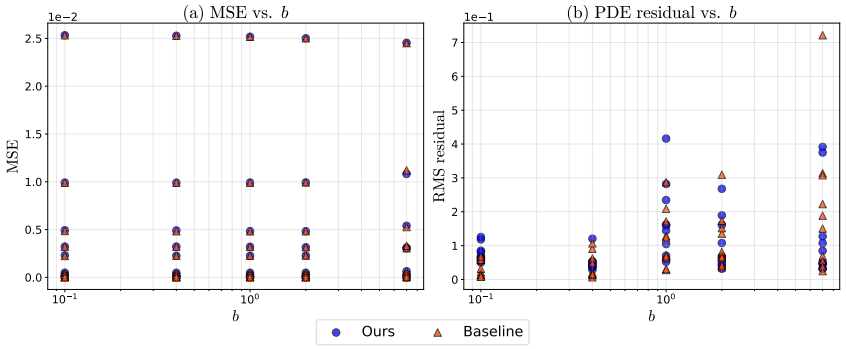

Figure 6: Comparison of our message passing approach (ours) and the SGD baseline for the advection equation; each point refers to one trajectory. **(a)** mean squared error (MSE) of the predictive mean against the exact solution; **(b)** root mean square (RMS) of the PDE residual R. Both methods reach errors of the same order at every wave speed.

A.7 Additional Evaluations

We provide additional experimental evaluations for the advection and Fisher-KPP equations and the PDE residual prior.

A.7.1 Advection equation

This appendix includes additional results for the advection equation that are mentioned in Section 4 before. Figure 6 compares inferred posteriors between our approach and the non-Bayesian PINN baseline. Figure 7(a) demonstrates a monotonic pattern in the uncertainty increase over time; it also shows that the rate at which uncertainty increases depends on the wave speed b, indicating that the inferred uncertainty responds to changes in the PDE dynamics as well as to the distance from the observed initial condition.

Figures 8 and 9 support the claim from Section 3 that our inferred means are close to those from HMC and VI, but our inferred variance has a more similar structure to HMC than VI does.

Figure 10 provides an ablation study on wider architectures with the increasing number of neurons. Among other results, it demonstrates that increasing dimensionality does not substantially increase training time.

A.7.2 Fisher-KPP equation

This appendix presents additional results for the Fisher-KPP equation that are discussed in the main text but included here due to space limitations. Figures 11 and 12 show predicted means and variances for two different random trajectories; as in the advection results, the means reflect dynamics close to the exact solutions. For one of the initial conditions, the prediction error increases with the growth rate ρ, but configurations with high growth rates are already known to be problematic and are described in Krishnapriyan et al. (2021) as a source of failure modes. The predicted variance shows the following pattern: it is especially large when a physical process is active, but otherwise it grows over time as we move further from the observed initial condition.

Figure 13 investigates a broader scope of the trajectories: it compares our approach with the non-Bayesian PINN baseline introduced in Section 4.1 across different growth rates ρ and demonstrates that both methods exhibit similar performance in terms of MSE and PDE residual, while our method even exhibits slightly lower residual.

Figure 14 presents the time-efficiency results: our approach is significantly faster than both HMC and VI, and comparable to VI for inference. VI is the slowest during training because it does not meet the convergence criterion within the allocated training budget, as shown by the quantitative results in Table 1.

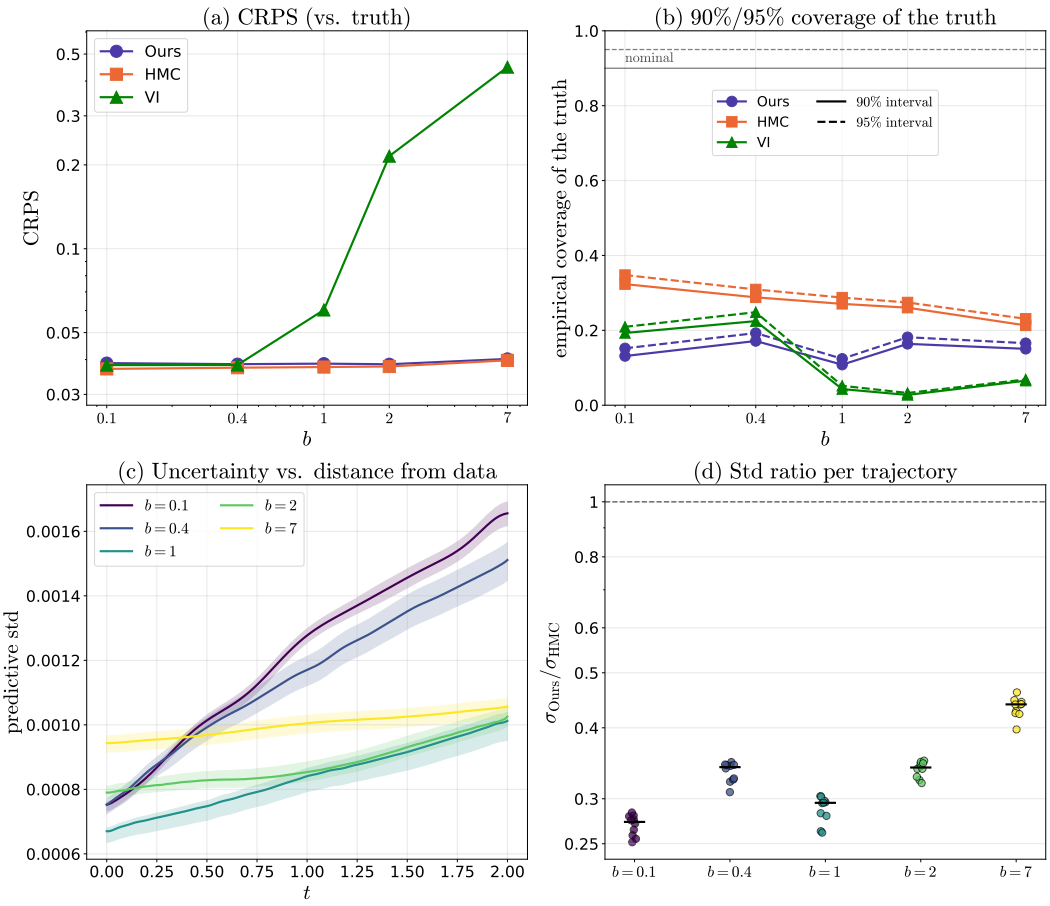

Figure 7: Calibration and uncertainty structure for the advection equation, ten trajectories per wave speed b. **(a)** CRPS against the exact solution; **(b)** empirical coverage of the exact solution by the nominal 90 % (solid) and 95 % (dashed) predictive intervals; **(c)** our mean predictive standard deviation over time t (band: minimum to maximum over trajectories), growing with the distance from the observed initial condition at a rate that depends on b; **(d)** per-trajectory median ratio of our predictive standard deviation to that of HMC (bar: median over trajectories), roughly constant within each b. Ours and HMC share accuracy and the uncertainty pattern, whereas VI degrades for $b \geq 1$.

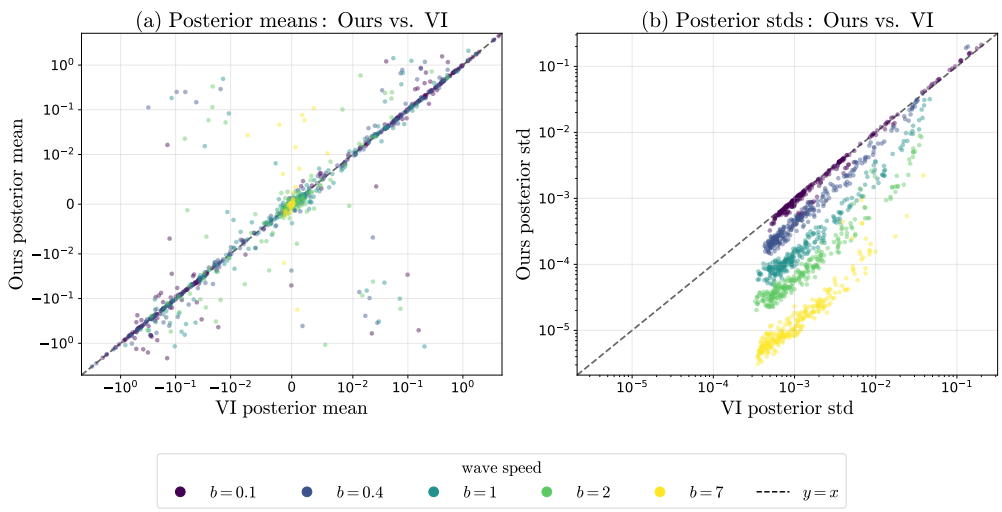

Figure 8: Comparison of our approach (ours) to VI, implemented based on Yang et al. (2021), for the advection equation. We map posterior weights of VI (x-axis) to our approach (y-axis) - points in **(a)** refer to the mean values, points in **(b)** refer to the standard deviations; the means largely agree, while the VI standard deviations deviate from ours increasingly with the wave speed b. For readability, 25 random weights per trajectory are shown.

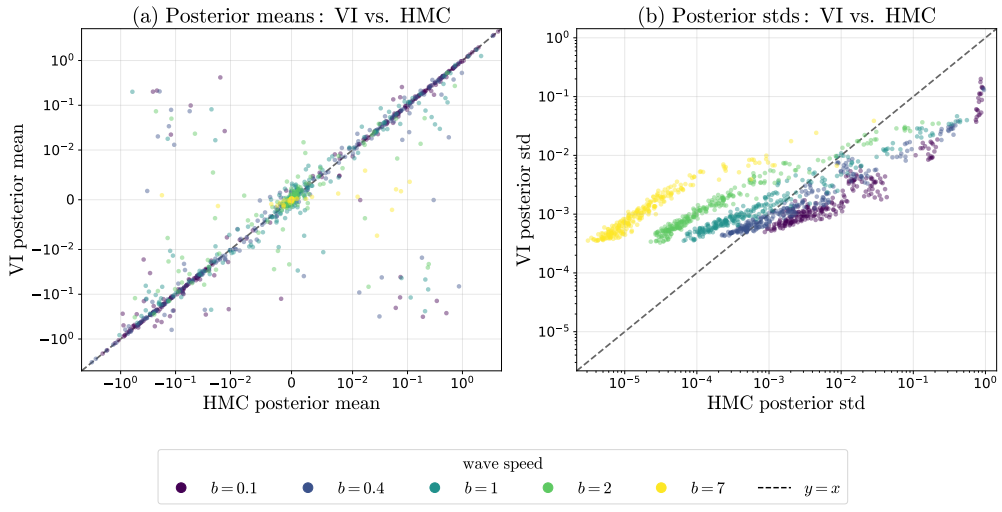

Figure 9: Comparison of VI to HMC, both implemented based on Yang et al. (2021), for the advection equation. We map posterior weights of HMC (x-axis) to VI (y-axis) - points in **(a)** refer to the mean values, points in **(b)** refer to the standard deviations; VI underestimates the standard deviations at $b \leq 0.4$ and overestimates them at $b \geq 2$, so it is further from HMC than our approach (Figure 3). For readability, 25 random weights per trajectory are shown.

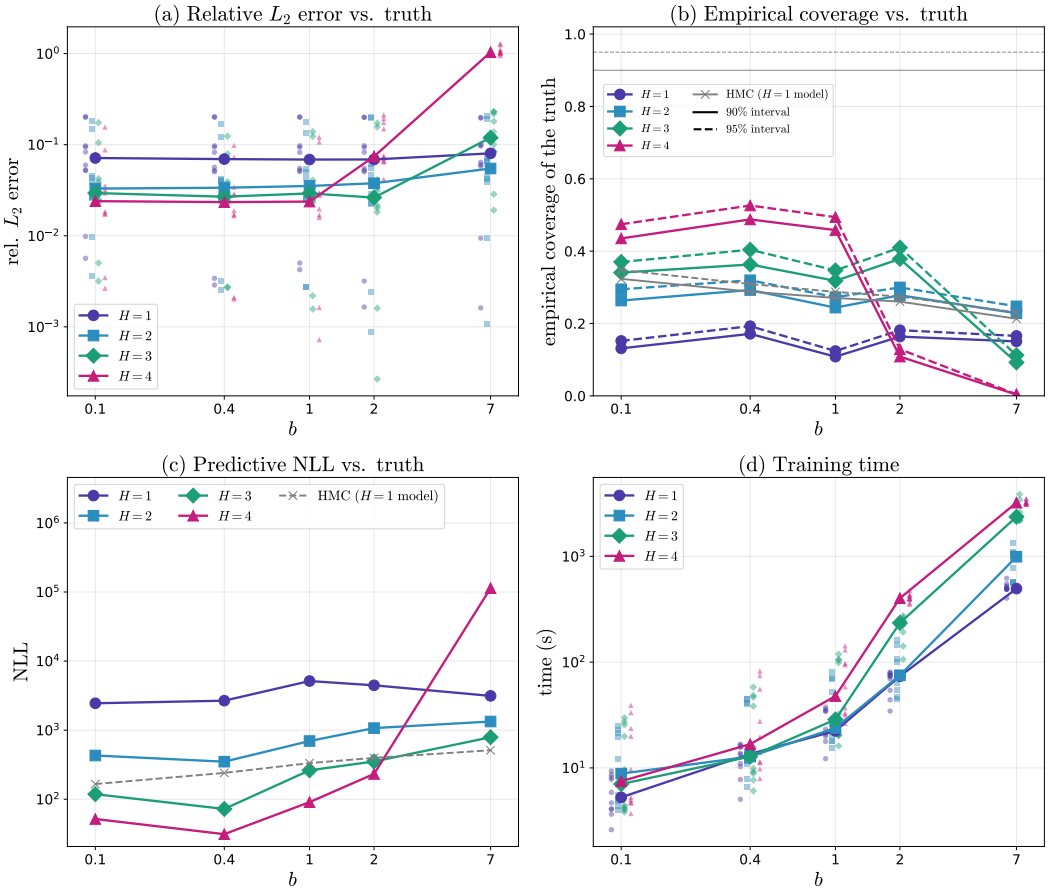

Figure 10: Neural layer width ablation for the advection equation: $H \in \{1, 2, 3, 4\}$ neurons in the hidden layer. **(a)** Relative L_2 error of the predictive mean (points: trajectories; lines: medians); **(b)** empirical coverage of the exact solution by the nominal 90 % (solid) and 95 % (dashed) predictive intervals; **(c)** predictive negative log-likelihood (NLL) of the exact solution, mean over trajectories; **(d)** training time (points: trajectories; lines: medians). HMC (grey) refers to the $H = 1$ model. Wider layers reduce the error and improve calibration for $b \leq 1$ at a higher training cost; at larger b, $H \geq 3$ degrades, and $H = 4$ fails at $b = 7$.

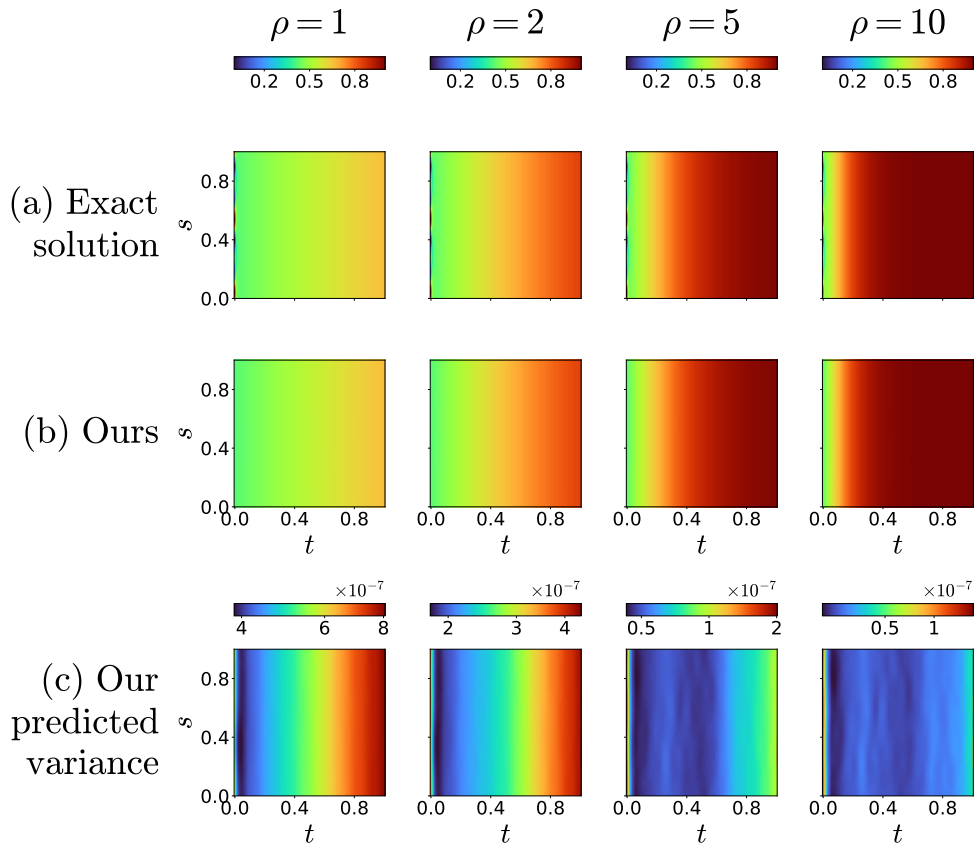

Figure 11: Our message passing approach for the Fisher-KPP equation on one PDEBench trajectory at $\nu = 1$ across reaction strengths $\rho \in \{1, 2, 5, 10\}$, in the coordinates $\boldsymbol{x} = (s, t)$. **(a)** the exact solution; **(b)** predicted mean values of our message passing solution, which reproduce the reaction-driven saturation; **(c)** predicted variance values of our message passing solution.

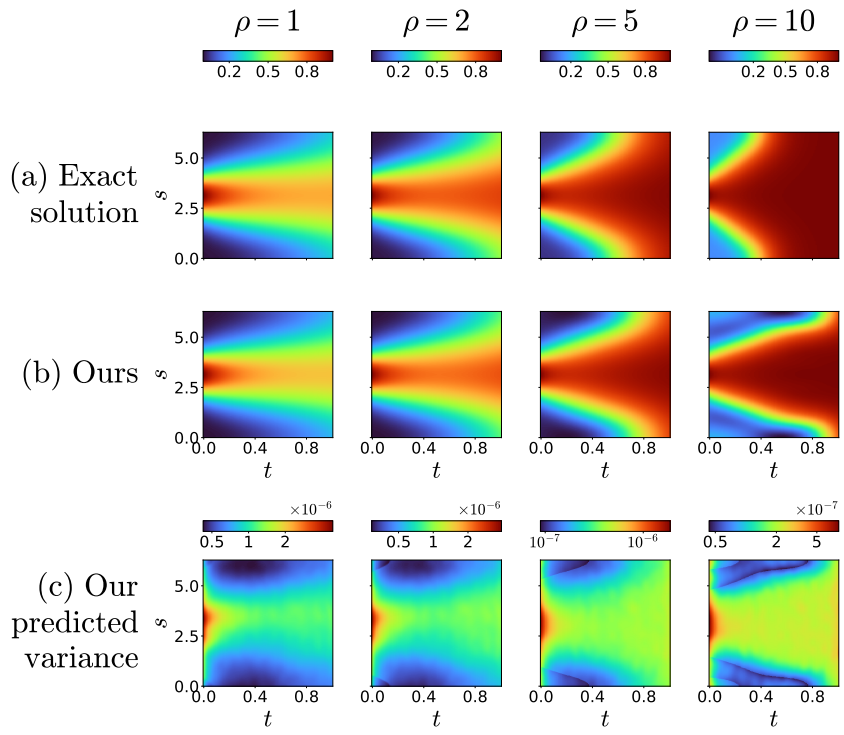

Figure 12: Our message passing approach for the Fisher-KPP equation with a Gaussian initial condition, $f(s,0) = \exp\big(-(s-\pi)^2/(2(\pi/4)^2)\big)$ on $s \in [0, 2\pi)$, at $\nu = 1$ across reaction strengths $\rho \in \{1, 2, 5, 10\}$, in the coordinates $\boldsymbol{x} = (s, t)$. **(a)** the exact solution; **(b)** predicted mean values of our message passing solution; **(c)** predicted variance values of our message passing solution.

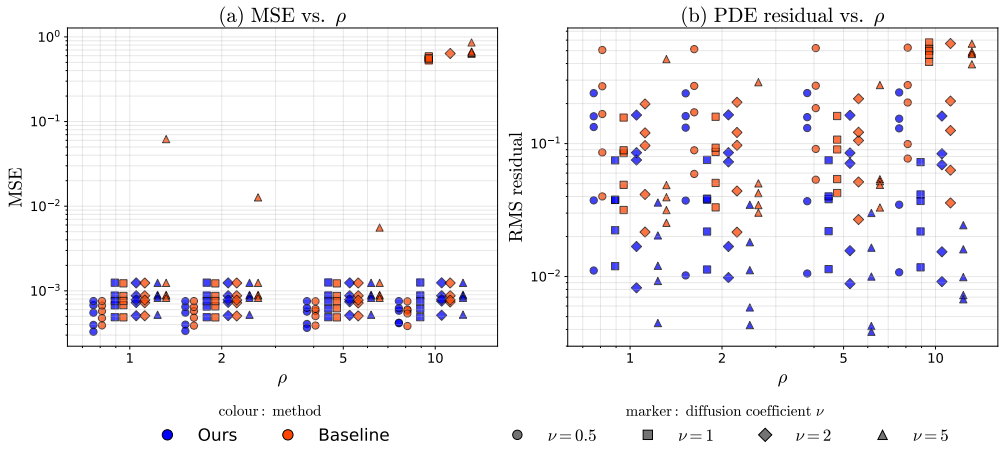

Figure 13: Comparison of our message passing approach (ours) and the SGD baseline for the Fisher-KPP equation across reaction strengths $\rho \in \{1, 2, 5, 10\}$ (log scale), with the diffusion coefficient $\nu \in \{0.5, 1, 2, 5\}$ shown by the marker, five trajectories per setting; each point refers to one trajectory. **(a)** MSE of the predictive mean against the exact solution; **(b)** root mean square (RMS) of the PDE residual R. Our errors are of the same order in all settings, while the baseline collapses to the trivial solution $f \equiv 0$ on most runs at $\rho = 10$, $\nu \geq 1$.

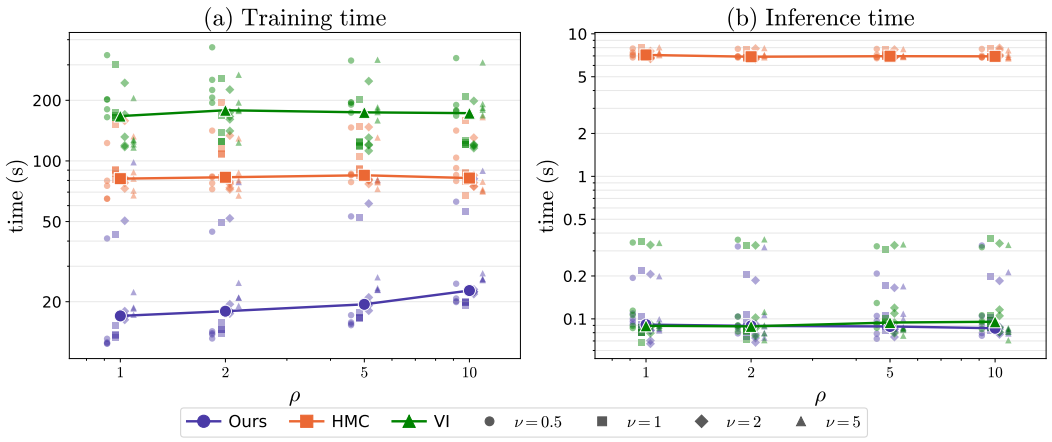

Figure 14: Computational cost of our approach (ours), HMC and VI for the Fisher-KPP equation across reaction strengths ρ, with the diffusion coefficient ν shown by the marker (points: runs; lines: medians). **(a)** Training time, cold start; **(b)** inference time of the predictive mean and variance on the full grid, steady state. Our training is about $4\times$ faster than HMC and $9\times$ faster than VI; our closed-form inference is as fast as VI's and about $80\times$ faster than HMC's sampling-based inference.

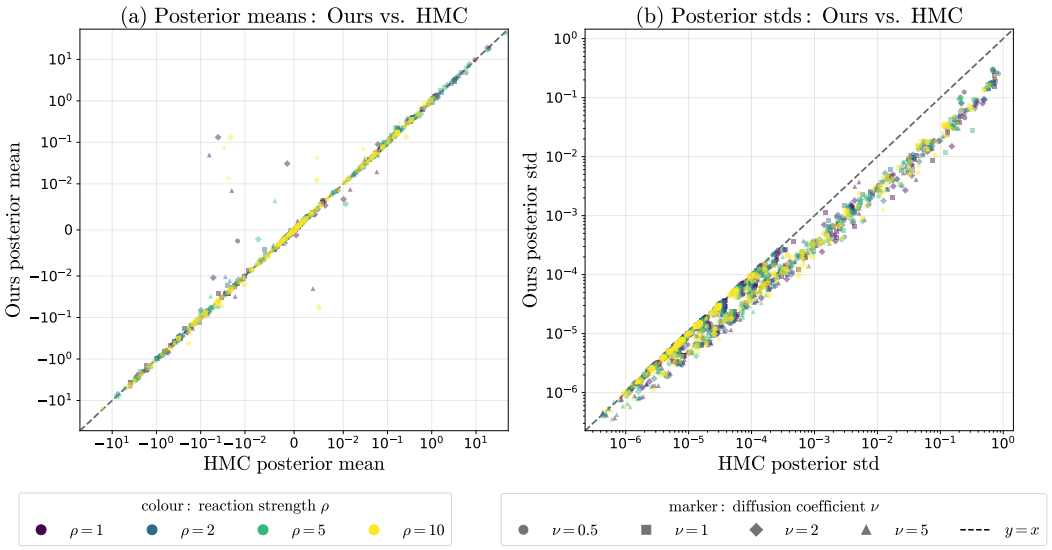

Figure 15: Comparison of our approach (ours) to HMC, implemented based on Yang et al. (2021), for the Fisher-KPP equation (color: ρ; marker: ν). We map posterior weights of HMC (x-axis) to our approach (y-axis) - points in **(a)** refer to the mean values, points in **(b)** refer to the standard deviations; the means agree closely, and our standard deviations are slightly smaller than, but proportional to, those of HMC. For readability, 25 random weights per trajectory are shown.

Posterior comparison against HMC and VI is shown in Figures 15 and 16. As with the advection results, we also observe that our approach infers means that are very close to those produced by HMC and VI, while our uncertainties are much closer to HMC uncertainties. Figure 17 additionally shows that VI standard deviations strongly decline from the HMC ones. Overall, it exhibits the same pattern as in the advection results, discussed in Section 4.2.

Figure 18 compares the uncertainty calibration across these Bayesian methods. Unlike the advection results, the ratio of our predicted uncertainty to the HMC one is not as constant, though it does not change much. Calibration results also show similar patterns between our method and HMC.

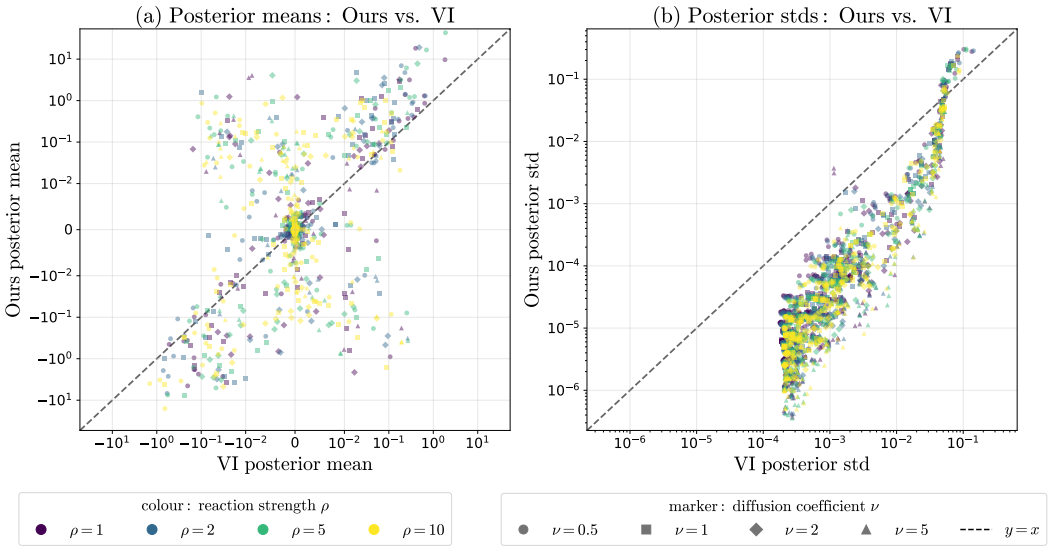

Figure 16: Comparison of our approach (ours) to VI, implemented based on Yang et al. (2021), for the Fisher-KPP equation (colour: ρ; marker: ν). We map posterior weights of VI (x-axis) to our approach (y-axis) - points in **(a)** refer to the mean values, points in **(b)** refer to the standard deviations; VI reaches neither our means nor our standard deviations, since it does not converge to the solution within the training budget. For readability, 25 random weights per trajectory are shown.

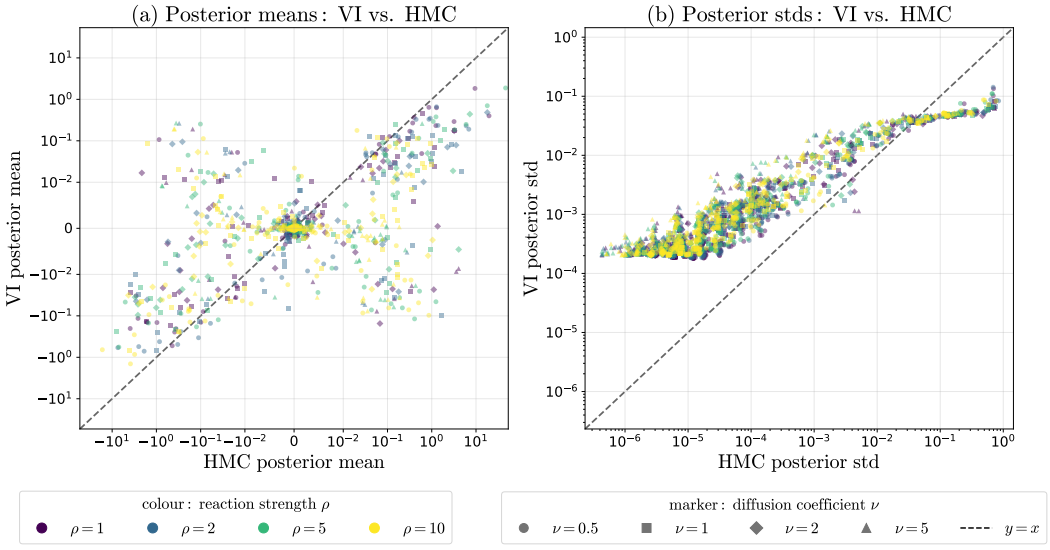

Figure 17: Comparison of VI to HMC, both implemented based on Yang et al. (2021), for the Fisher-KPP equation (colour: ρ; marker: ν). We map posterior weights of HMC (x-axis) to VI (y-axis) - points in **(a)** refer to the mean values, points in **(b)** refer to the standard deviations; VI neither recovers the HMC means nor its standard deviations, which it overestimates by up to two orders of magnitude. For readability, 25 random weights per trajectory are shown.

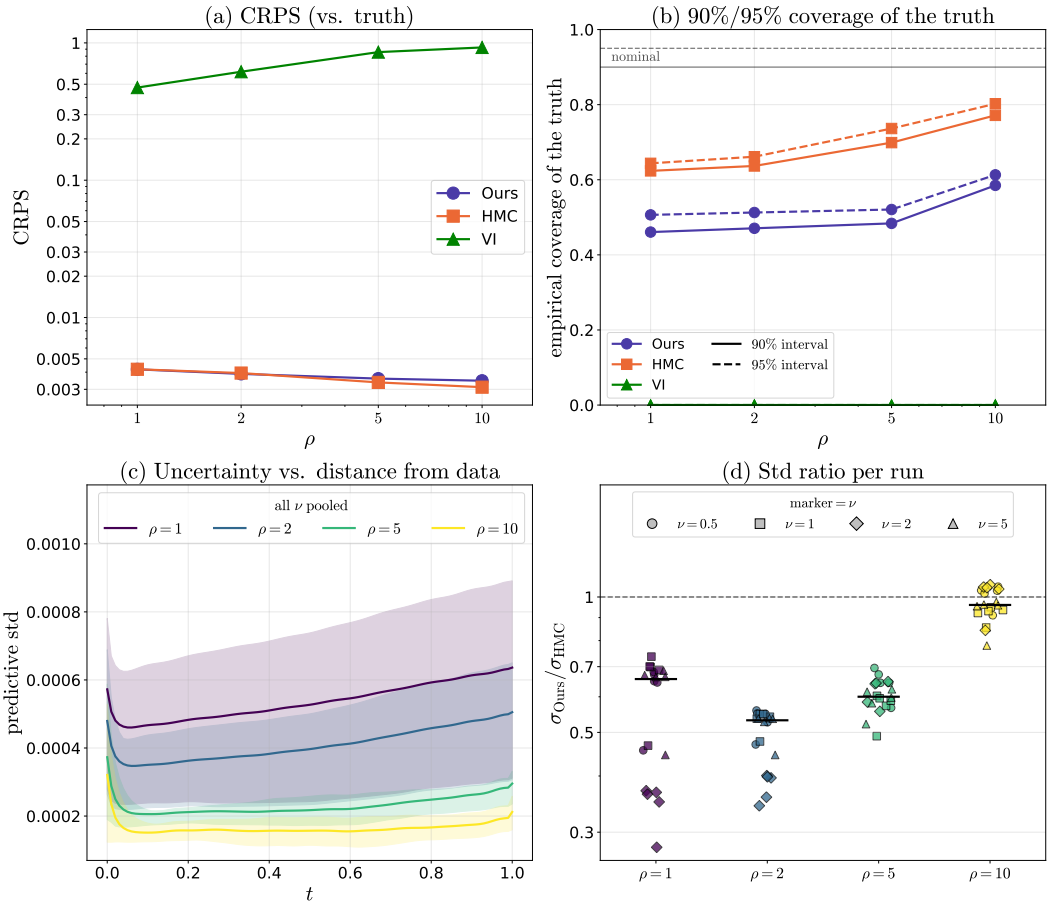

Figure 18: Calibration and uncertainty structure for the Fisher-KPP equation across reaction strengths ρ, pooled over the diffusion coefficients ν. **(a)** CRPS against the exact solution; **(b)** empirical coverage of the exact solution by the nominal 90 % (solid) and 95 % (dashed) predictive intervals; **(c)** our mean predictive standard deviation over time t (band: minimum to maximum over runs); **(d)** per-run median ratio of our predictive standard deviation to that of HMC (bar: median over runs; marker: ν). Ours closely follows HMC in CRPS but is under-covered relative to it, with median ratios between 0.53 and 0.96; VI fails in all settings.

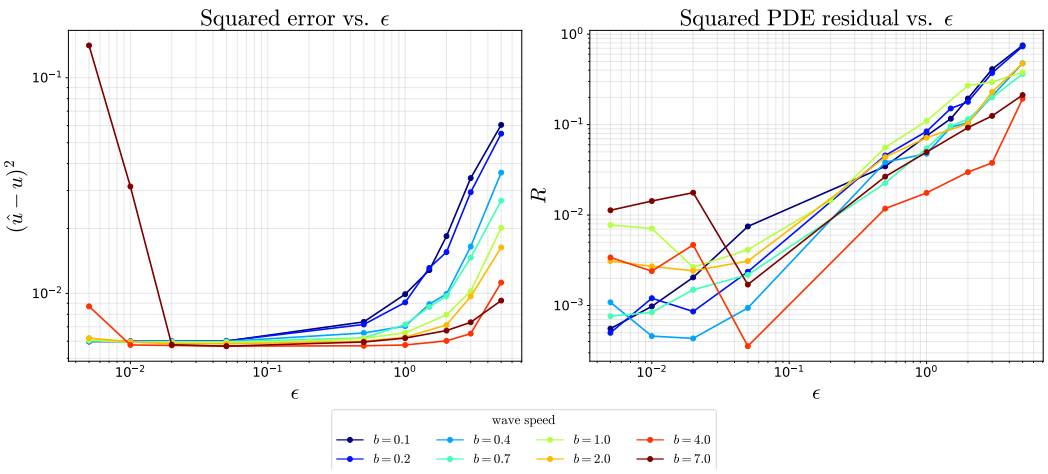

Figure 19: Effect of the PDE tolerance ϵ_C for the advection equation, ten trajectories per wave speed b (lines: medians; bands: interquartile ranges; dotted: the default $\epsilon_C = 0.05$). **(a)** MSE of the predictive mean against the exact solution; **(b)** mean squared PDE residual R. Tightening ϵ down to about 0.1 lowers both; below that the error no longer improves, and at $b = 7$ with $\epsilon = 0.005$ the model collapses to a constant solution, i.e., exhibits a failure mode.

A.7.3 PDE RESIDUAL PRIOR

Figure 19 studies the effect of the PDE tolerance ϵ_C on the advection equation. Decreasing ϵ_C enforces the PDE constraint more strongly and generally reduces both the prediction error and the PDE residual. However, when the constraint becomes too strong, especially for higher wave speeds ($b \geq 4$), the solution can collapse towards a constant function. Such a solution satisfies the advection equation with a very small residual but does not preserve the propagated initial condition. Thus, ϵ_C controls the balance between enforcing the PDE and retaining information from the observed initial condition.