Combining Flow Matching and Transformers for Efficient Solution of Bayesian Inverse Problems

Daniil Sherki[a], **Ivan Oseledets**[b], **Ekaterina Muravleva**[a]

[a] *Sberbank of Russia, AI4S Center, Moscow, Russian Federation* DSSherki@sberbank.ru
Skolkovo Institute of Science and Technology Daniil.Sherki@skoltech.ru

[b] *Artificial Intellidence Research Institute, Moscow, Russian Federation*
Skolkovo Institute of Science and Technology

[a] *Sberbank of Russia, AI4S Center, Moscow, Russian Federation*
Skolkovo Institute of Science and Technology

1. Introduction

This paper presents a new approach to solving Bayesian inverse problems using generative artificial intelligence models. Bayesian inverse problems are found in various applications in physics, mathematics, and other natural sciences.

Our method directly generates the posterior distribution, avoiding intermediate steps to provide a more accurate representation of uncertainty, a critical aspect of Bayesian inference. We propose a model architecture capable of accommodating an arbitrary number of observations, making it adaptable to diverse problem settings.

2. Related works

The table 1 provides methods comparing for solving Bayesian inverse problems. Deep learning methods for solving Bayesian inverse problems exhibit distinct strengths and limitations. MDGM [1] leverages a VAE-based convolution neural network with MCMC for multiscale inference, excelling in high-dimensional PDE-based problems but lacking exact likelihood estimation and flexibility for arbitrary observations. MCGAN [2] combines MCMC with GANs for high-fidelity posterior sampling but suffers from computational complexity, lack of explainability, and fixed observation models. PI-INN [3] employs physics-informed flow-based models, enabling exact likelihood estimation and end-to-end training but struggles with variable observation sizes due to architectural constraints. In contrast, CFM-Tr integrates conditional flow matching with transformers, offering exact likelihood estimation, end-to-end training, and adaptability to arbitrary observations.

3. Methodology

Consider a forward model defined as:

$$d = \mathcal{F}(m, e) + \eta,$$

where m is the model parameters, sampled from their prior distribution, e is the experimental conditions or design, also sampled, η is random noise, sampled from a predefined noise distribution.

Using this process, we construct a dataset comprising samples (d, m, e), which are drawn from the **conditional distribution** $\rho(d, m, e)$. The primary objective is to address the inverse problem: given d and e, infer the model parameters m. Since m is not uniquely determined by d and e, it is characterized by the conditional distribution $\rho(m|d, e)$. Our goal is to estimate this conditional distribution.

To achieve this, we employ a conditional flow matching (CFM) framework from original work [4]. This involves first sampling an unconditional prior distribution for m (denoted as m_0). Next, we define a conditional interpolation path between (m_0, d, e) and (m, d, e), where (d, m, e) is a sample from the dataset. The interpolation path is given by:

$$m_t = (1 - t)m_0 + t \cdot m, \quad t \in [0, 1]$$

In the CFM approach, we learn a velocity field $v_\theta(m_t, t, d, e)$ that minimizes the following objective:

$$\mathbb{E}_{t, m_0, (m,d,e) \sim \rho} \left[\| v_\theta(m_t, t, d, e) - (m - m_0) \|^2 \right] \to \min_\theta$$

Here, v_θ represents a learnable function parameterized by θ, which predicts the velocity field given inputs (m_t, t, d, e). The input dimension corresponding to m_t, t, d, and e, while the output dimension matches the dimensionality of m.

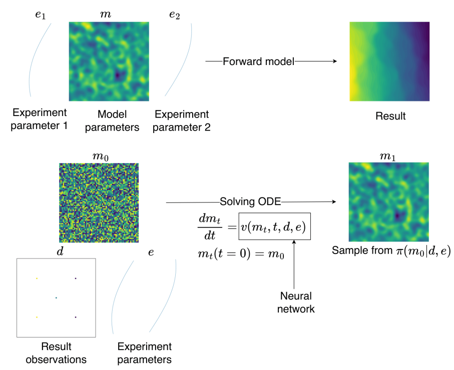

Fig. 1: Scheme of using the forward model and pre-trained flow matching model for solving Bayesian inverse problem

The figure 1 illustrates schematically the inference method of a trained flow matching neural net-

Method	Base model	Exact likelihood estimation	No middle-man Training	Arbitrary number of observations
MDGM	VAE based on CNN	×	✓	×
MCGAN	MCMC + GAN	×	×	×
PI-INN	PI + flow-based model	✓	✓	×
CFM-Tr (ours)	CFM + Transformer	✓	✓	✓

Table 1: Comparison of methods for solving Bayeisan Inverse problems. *MDGM use the PDE solution as a holistic observation; the problem was not formulated as the recovery of the forward model from a small number of observations

work. In this example, the forward problem involves solving a two-dimensional coefficient equation m, where the experimental parameters e represent the boundary conditions, and the observations d correspond to the values of the solution to the equation at arbitrary points.

4. Numerical Experiments

The Bayesian inverse problem cases were adopted from the article [5] and consist of the following: experiments on the reconstruction of parameters in a system of ODEs, used to model the spread of a disease (based on the SEIR model) within a population, as detailed in Appendix B. The second example involves the reconstruction of a coefficient in a PDE for Darcy Flow using a limited number of solution points from the PDE. Technical details and mathematical derivations are provided in Appendix C.

5. Results and Discussions

Table 2 presents the results of numerical experiments for our proposed method using the relative error metric for differential equations solution, generated and ensembled in 10 times.

The true solution of the ODE system and the reconstructed parameter distribution, obtained using only 4 data points, are illustrated in Figure 2. The results of the restoration of the 2D coefficients of the PDE (Permeability field inverse problem) are shown in the Figure 3.

Table 2: The relative inference error of the trained model for SEIR problem (Appendix B) and Permeability Field (Appendix C)

N	SEIR Problem	Permeability Field
2	$10.88\% \pm 2.39\%$	$28.84\% \pm 3.43\%$
3	$3.31\% \pm 1.47\%$	$16.23\% \pm 1.53\%$
4	$2.80\% \pm 1.37\%$	$17.80\% \pm 1.99\%$
5	$2.15\% \pm 0.99\%$	$16.86\% \pm 1.76\%$
6	$1.97\% \pm 0.91\%$	$7.21\% \pm 1.26\%$
7	$1.59\% \pm 0.75\%$	$7.48\% \pm 1.23\%$
8	$1.48\% \pm 0.71\%$	$2.75\% \pm 0.60\%$

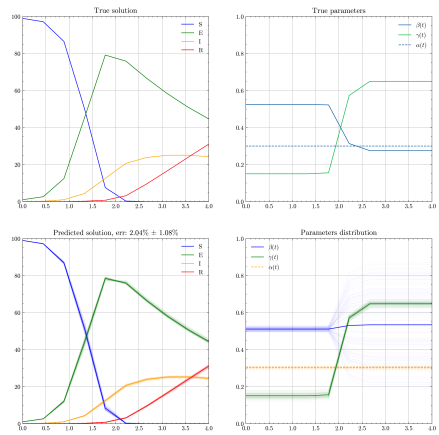

Fig. 2: Probabilistic solutions to the inverse problem for $\mathbf{m}_{\mathbf{true}} = [0.4, 0.3, 0.3, 0.1, 0.15, 0.6]$

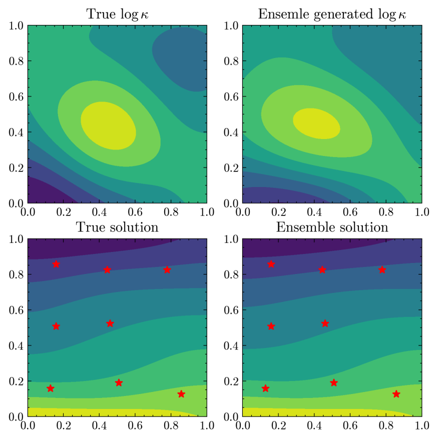

Fig. 3: PDE coefficient and solution: true (left) and reconstructed using Flow-matching (right)

6. Conclusions

Thus, our method is universal and can be adapted to a large number of problems in a short time if the problem is reduced to the standard Bayesian Inverse Problem formulation, since it can learn complex nonlinear distributions. Also a great advantage is the possibility of using an input that is not fixed in terms of the number of observations, where increasing the number of observed points increases the accuracy of the method in terms of recovering the solution from the generated parameters.

References

[1] Yingzhi Xia and Nicholas Zabaras. Bayesian multiscale deep generative model for the solution of high-dimensional inverse problems. *Journal of Computational Physics*, 455:111008, April 2022.

[2] Nikolaj T. Mücke, Benjamin Sanderse, Sander Bohté, and Cornelis W. Oosterlee. Markov chain generative adversarial neural networks for solving bayesian inverse problems in physics applications, 2022.

[3] Xiaofei Guan, Xintong Wang, Hao Wu, Zihao Yang, and Peng Yu. Efficient bayesian inference using physics-informed invertible neural networks for inverse problems, 2023.

[4] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023.

[5] Karina Koval, Roland Herzog, and Robert Scheichl. Tractable optimal experimental design using transport maps, 2024.

Appendix A. Flow Matching Algorithm

Algorithm 1: Conditional Flow Matching Training Algorithm

Input: Dataset of paired samples (x_1, e, d), neural network model $\mathbf{v}_\theta(t, x, e, d)$, Conditioning data e and d, time $t \sim$ Uniform$(0, 1)$.
Output: Trained conditional flow field $\mathbf{v}_\theta(t, x, e, d)$.
for *each minibatch of samples* (x_1, e, d) **do**
  $t \sim \mathcal{U}(0, 1)$ `// Sample t`
  $x_0 \sim$ prior distribution
  $x_t \leftarrow t \cdot x_1 + (1 - t) \cdot x_0$
  $u_t \leftarrow x_1 - x_0$ `// Compute the target velocity`
  $v_t \leftarrow \mathbf{v}(t, x_t, e, d)$ `// Predict the velocity`
  $\mathcal{L}(\theta) \leftarrow \mathbb{E}\left[(v_t - u_t)^2\right]$ `// Compute the loss`
  Update θ using the optimizer and $\nabla\mathcal{L}(\theta)$
end
return $\mathbf{v}_\theta(t, x, e, d)$

Appendix B. SEIR disease model problem statement

SEIR (Susceptible-Exposed-Infected-Recovered) model is a mathematical model used to mathematically simulate the spread of infectious diseases. In this case study we simulate a real situation where, during the spread of a disease, we measure the number of infected and dead people at random times and use this information to try to recover the control parameters of the ODE system.

In this experiment, we adopt the susceptible-exposed-infected-removed (SEIR) model as presented by Koval et al., a widely used framework for modeling the transmission dynamics of infectious diseases. Assuming a constant population size, the SEIR model is described by the following system of ordinary differential equations:

$$\frac{dS}{dt} = -\beta(t)SI, \frac{dE}{dt} = \beta(t)SI - \alpha E$$

$$\frac{dI}{dt} = \alpha E - \gamma(t)I, \frac{dR}{dt} = \gamma(t)I$$

where the variables $S(t)$, $E(t)$, $I(t)$, $R(t)$ are used to denote the fractions of susceptible, exposed, infected and removed individuals at time t, respectively, and are initialized with $S(0) = 99$, $E(0) = 1$, $I(0) = R(0) = 0$.

The parameters to be estimated are $\beta(t)$, α, γ^r, $\gamma^d(t)$, where the constants α and γ^r denote the rate of susceptibility to exposure and infection to recovery, respectively. To simulate the effect of policy changes or other time-dependent factors (e.g., quarantine and overcrowding of hospitals), the rates at which exposed individuals become infected and infected individuals perish are assumed to be time-dependent and parametrized as follows:

$$\beta(t) = \beta_1 + \frac{\tanh(7(t - \tau))}{2}(\beta_2 - \beta_1)$$

$$\gamma(t) = \gamma^r + \gamma^d(t)$$

$$\gamma^d(t) = \gamma_1^d + \frac{\tanh(7(t - \tau))}{2}(\gamma_2^d - \gamma_1^d)$$

i.e., the rates transition smoothly from some initial rate (β_1 and γ_1^d) to some final rate (β_2 and γ_2^d) around time $\tau > 0$.

In the following, we fix $\tau = 2.1$ and an overall time interval of $[0, 4]$. The experiment consists of choosing such four time intervals $e = [a_1, a_2, a_3, a_4] \sim \mathcal{U}[1, 3]$ to measure the number of infected and deceased individuals $d_i = [I_{e_i}, R_{e_i}]$ for $i \in [1, 4]$ ($d \in \mathbf{R}^{2\times4}$) for optimal inference of the 6 rates $\mathbf{m} = [\beta_1, \alpha, \gamma^r, \gamma_1^d, \beta_2, \gamma_2^d]$. After training MLP and solving the flow matching problem, we learned to smoothly transition from the distribution $\mathcal{U}[0, 1]^6$ to the distribution $\hat{\mathbf{m}} \sim \rho(\mathbf{m}|\mathbf{e}, \mathbf{d})$.

- $e = [a_1, a_2, a_3, a_4] \sim \mathcal{U}[1, 3]$ random times, when measurements are performed
- $d_i = [I_{e_i}, R_{e_i}]$ for $i \in [1, 4]$ ($d \in \mathbf{R}^{2\times4}$) the number of infected and deceased individuals
- $m = [\beta_1, \alpha, \gamma^r, \gamma_1^d, \beta_2, \gamma_2^d]$ is ODE model parameters.

Appendix C. Permeability field inversion problem statement

$$-\nabla \cdot (\kappa \nabla u) = 0$$

with boundary conditions

$$u(x=0,y) = f(y,e_1) = \exp\left(-\frac{1}{2\sigma_w}(y-e_1)^2\right)$$

$$u(x=1,y) = g(y,_2) = -\exp\left(-\frac{1}{2\sigma_w}(y-e_2)^2\right)$$

was solved using the finite element (FE) method with second-order Lagrange elements on a mesh of size $h = \frac{1}{64}$ in each coordinate direction, where κ is a custom 2D matrix.

This is a common problem, for example in the oil industry, when there are a small number of wells and pressure observations across them, and from these data one needs to reconstruct the permeability field of an oil field.

In this example, the inverse problem consists of estimating the spatially-dependent diffusivity field κ, given measurements of the pressure u at some pre-determined locations $(x_i, y_i) \in \Omega$. To ensure κ is nonnegative, we impose a Gaussian prior on the log diffusivity, $m = \log(\kappa) \sim N(0, C_{pr})$, with covariance operator C_{pr} defined using a squared-exponential kernel

$$c(x,z) = \sigma_v^2 \exp\left[\frac{-\|x-z\|^2}{2\ell^2}\right] \quad \text{for } x, z \in \Omega,$$

with $\sigma_v = 1$ and $\ell^2 = 0.1$. Employing a truncated Karhunen-Loève expansion of the unknown diffusivity field yields the approximation

$$m(x, \mathbf{m}) \approx \sum_{i=1}^{n_m} m_i \sqrt{\lambda_i}\phi_i(x),$$

where λ_i and $\phi_i(x)$ denote the i-th largest eigenvalue and eigenfunction of C_{pr}, respectively, and the unknown coefficients $m_i \sim N(0,1)$. The Karhunen-Loève expansion is truncated after $n_m = 16$ modes, resulting in an approximation that captures 99 percent of the weight of C_{pr}.

The input consists of a vector of values d of arbitrary length and two corresponding vectors of coordinates x, y. The final input is a matrix $\mathbf{D} = (\mathbf{d}, \mathbf{x}, \mathbf{y})^T$ with shape $(n, 3)$. e vector parametrizing smooth functions for boundary condition with shape $(2,)$ and m vector of 16 parameters $(16,)$.

Appendix D. Transformer model architecture

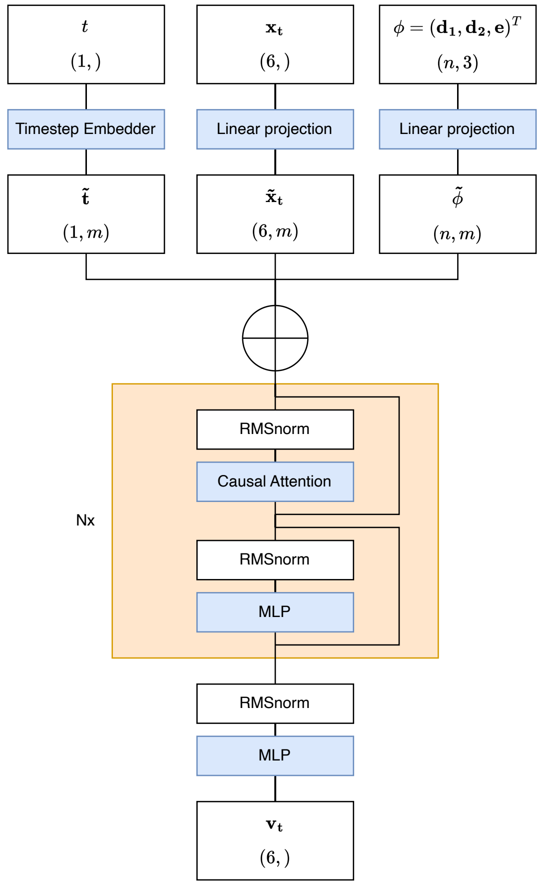

Fig. A1: Transformer model architecture for SEIR model case(n is number of observation points, m is a embedding size)