Neural Control of Parametric Solutions for Evolution PDEs

Xiaojing Ye

Department of Mathematics and Statistics, Georgia State University

Joint work with Nathan Gaby (GSU) and Haomin Zhou (GT)

This research is partially supported by National Science Foundation

Motivations

1. Partial Differential Equations (PDEs) are central to modeling phenomenon from science and engineering, to finance and economics.
2. PDEs lack explicit closed form solutions.
3. Many of the traditional numerical methods for PDEs suffer from the so-called "curse-of-dimensionality".
4. Use neural-networks as reduced order models to approximate the solutions of PDEs.
5. While effective, if any initial data of the PDE changes (boundary values, initial values) neural-network based methods require expensive retraining.

Evolution PDEs

Consider the following class of Initial Value Problem (IVP) with an evolution PDE:

$$\text{(PDE)} \quad \begin{cases} \partial_t u(x,t) = F[u](x,t), & \forall\, x \in \Omega, t \in [0,T] \\ B[u](x,t) = 0, & \forall\, x \in \partial\Omega, t \in [0,T] \\ u(x,0) = g(x), & \forall\, x \in \bar{\Omega} \end{cases}$$

- F is a potentially nonlinear differential operator of u.
- B is the boundary conditions operator.
- g is the initial value of u.

Related Work

A rather incomplete list of related work:

- Classical methods: Finite Difference (Thomas '13), Finite Element (Johnson '12) etc.
- NNs for PDE:
  - Strong form: PINN (Raissi, Perdikaris, Karniadakis '19), nPINN (Pang, D'Elia, Parks, Karniadakis '20), fPINN (Pang, Lu, Karniadakis '19) etc.
  - Variational form: Deep Ritz (Yu, E '18) etc.
  - Weak form: Weak adversarial net (Zang, Bao, Ye, Zhou '19, '20) etc.
  - Faynman-Kac: Backward SDE (Beck, E, Jentzen '17, Han, E, Jentzen '17, '18) etc.
- Solution operator of PDE:
  - Green's function: NN approx Green's function (Boullé, Kim, Shi Townsend '22, Teng, Zhang, Wang, Ju '22) etc.
  - Operator learning: DeepONet (Lu, Jin, Karniadakis '19), FNO (Li, Kovachki, Azizzadenesheli, Liu, Bhattacharya, Stuart, Anandkumar '20) etc.

Problems to Tackle

1. Seek a method to solve the IVP for different initial values without the need to retrain.
2. The method should be able to apply to high-dimensional problems.
3. Simple to implement and generalizable to nonlinear PDEs.
4. A rigorous error estimate of the approximate solution.

Neural Networks (NNs)

We first parameterize the solution of IVPs using reduced-order models, such as Neural Networks (NNs).

Many structures exist for NNs (e.g. Feedforward, CNN, RNN, ResNet, NF, NODE, etc.)

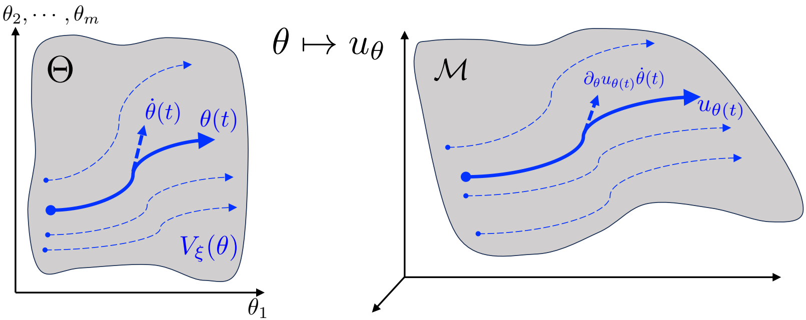
We require:

- A NN $\theta \mapsto u_\theta \in C(\bar{\Omega})$ where $\theta \in \mathbb{R}^m$ are the parameters of u_θ.
- $u_\theta : \bar{\Omega} \to \mathbb{R}$ is smooth with respect to θ.

Parameter Submanifold

To establish motivation, realize the following:

- Let u_θ be an NN with parameter $\theta \in \Theta \subset \mathbb{R}^m$. Θ is parameter space.
- Then $\theta \mapsto u_\theta$ defines a set of functions by $\mathcal{M} := \{u_\theta : \Omega \to \mathbb{R} \mid \theta \in \Theta\}$.
- We call $\mathcal{M}$ the parameter submanifold determined by the architecture of u_θ.
- So a curve $\theta(t) \in \Theta$ corresponds to a trajectory $u_{\theta(t)}$ on $\mathcal{M}$.



Control Vector Field in Parameter Space

- For any initial g, let $u^g(\cdot, t)$ denote the solution of the IVP.
- Under sufficient regularity conditions, there is a curve $\theta(t)$ in the parameter space Θ such that $u_{\theta(t)}(\cdot) \in \mathcal{M}$ tracks $u^g(\cdot, t)$, i.e.,

$$u_{\theta(t)}(\cdot) \approx u^g(\cdot, t), \qquad \forall\, t.$$

- We want to learn a control V in Θ such that it can steer $\theta(t)$ to obtain such close tracking of $u^g(\cdot, t)$ from starting point θ_0.
- If such a vector field V is continuous then we need solve

$$\text{(ODE)} \qquad \begin{cases} \dot{\theta}(t) = V(\theta(t)) \\ \theta(0) = \theta_0 \end{cases}$$

  to generate the desired $\theta(t)$.
- Note that this control vector field V is universal for all g.

Proposed Method

Suppose then we have another NN $V_\xi : \Theta \to \mathbb{R}^m$ which is a vector field defined over Θ.

What requirement should V_ξ satisfy to steer $\theta(t)$ such that $u_{\theta(t)}$ tracks $u^g(\cdot, t)$?

- Since $u_{\theta(t)}(\cdot) \approx u^g(\cdot, t)$ which solves the PDE, we need

$$\partial_t u_{\theta(t)}(x) = F[u_{\theta(t)}](x)$$

- We also have

$$\partial_t u_{\theta(t)}(x) = \partial_\theta u_{\theta(t)}(x) \cdot \dot{\theta}(t) = \partial_\theta u_{\theta(t)}(x) \cdot V_\xi(\theta(t))$$

- Therefore, it suffices to have for any $x \in \Omega$ and $\theta \in \Theta$ that

$$F[u_\theta](x) = \partial_\theta u_\theta(x) \cdot V_\xi(\theta), \quad \forall x \in \Omega, \theta \in \Theta$$

Proposed Method

Design u_θ such that $B[u_\theta] = 0$ for all $\theta \in \Theta$. Then seek V_ξ by solving

$$\min_\xi \ell(\xi) := \int_\Theta \int_\Omega \left|\partial_\theta u_\theta(x) \cdot V_\xi(\theta) - F[u_\theta](x)\right|^2 dx d\theta.$$

We form the empirical loss by sampling θ_i uniformly in Θ and for each θ_i sampling x_j uniformly in Ω:

$$\min_\xi \hat{\ell}(\xi) := \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left|\partial_\theta u_{\theta_i}(x_j) \cdot V_\xi(\theta_i) - F[u_{\theta_i}](x_j)\right|^2$$

Algorithms

Train V_ξ:

Algorithm 1. Training neural control field V_ξ

Input: Reduced-order model structure u_θ and parameter set Θ. Control vector field structure V_ξ. Error tolerance ε.

Output: Optimal control parameter ξ.

1: Sample $\{\theta_k\}_{k=1}^K$ uniformly from Θ.
2: Form empirical loss $\hat{\ell}(\xi)$.
3: Minimize $\hat{\ell}$ with respect to ξ using any optimizer (e.g. SGD) until $\hat{\ell}(\xi) \leq \varepsilon$.

Use V_ξ to solve IVP:

Algorithm 2. Solution operator of IVP using trained V_ξ

Input: Initial value g and tolerance ε_0. Reduced-order model u_θ and trained neural control V_ξ.

Output: Trajectory $\theta(t)$ such that $u_{\theta(t)}$ approximate the solution of the IVP (PDE).

1: Compute initial parameter θ_0 such that $\|u_{\theta_0} - g\|_2 \leq \varepsilon_0$.
2: Use any ODE solver to compute $\theta(t)$ to solve (ODE) with approximate field V_ξ and initial $\theta(0) = \theta_0$.

Numerical Results: 10D Transport Equation

Consider the following 10D transport equation:

$$\begin{cases} \partial_t u(x,t) = -1 \cdot \nabla u(x,t), & \forall x \in (0,1)^{10}, t \in [0,1] \\ u(x,t) = 0, & \forall x \in \partial(0,1)^{10} \\ u(x,0) = g(x), & \forall x \in [0,1]^{10}, \end{cases}$$

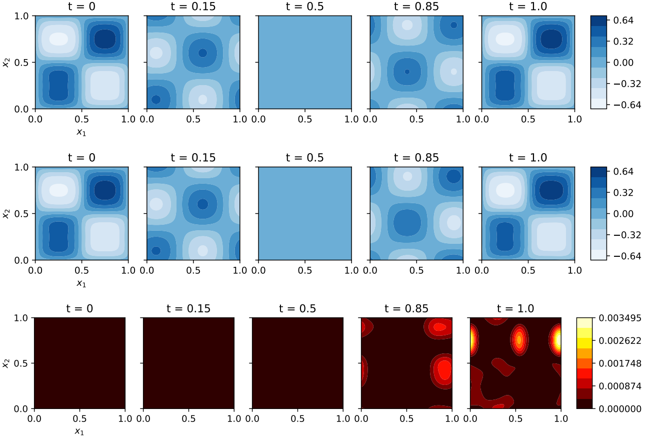
To have analytic examples to compare against, we tested our method against the following functions:

$$g_1(x) = \sin(2\pi x_1)\sin(2\pi x_2)\prod_{i=3}^{10}\sin(\pi x_i),$$

$$g_2(x) = \sin(2\pi x_1)\cos(2\pi x_2)\prod_{i=3}^{10}\sin(\pi x_i),$$

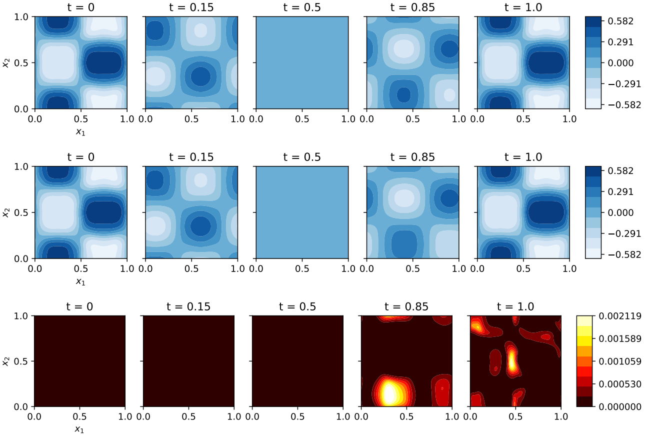
$$g_3(x) = \sin(4\pi x_1)\sin(2\pi x_2)\prod_{i=3}^{10}\sin(\pi x_i).$$

Numerical Results: 10D Transport Equation



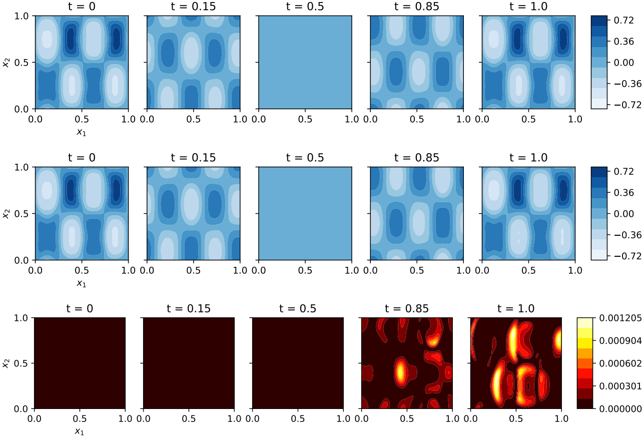
$$g_1(x) = \sin(2\pi x_1)\sin(2\pi x_2)\prod_{i=3}^{10}\sin(\pi x_i)$$

Numerical Results: 10D Transport Equation



$$g_2(x) = \sin(2\pi x_1)\cos(2\pi x_2)\prod_{i=3}^{10}\sin(\pi x_i)$$

Numerical Results: 10D Transport Equation



$$g_3(x) = \sin(4\pi x_1)\sin(2\pi x_2)\prod_{i=3}^{10}\sin(\pi x_i)$$

Numerical Results: 10D Transport Equation

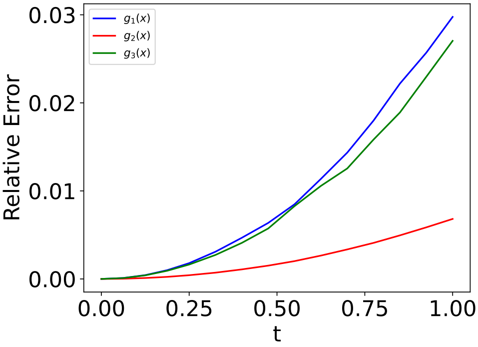


Figure: Comparison of the relative error $\|u^g(\cdot,t) - u_{\theta(t)}(\cdot)\|_{L^2(\Omega)}/\|u^g(\cdot,t)\|_{L^2(\Omega)}$ over time t for IVPs with initial values g_1, g_2 and g_3.

Numerical Results: 10D Heat Equation

Consider the following 10D heat equation:

$$\begin{cases} \partial_t u(x,t) = \Delta u(x,t), & \forall\, x \in (0,1)^{10}, t \in [0, 0.015] \\ u(x,t) = 0, & \forall\, x \in \partial(0,1)^{10} \\ u(x,0) = g(x), & \forall\, x \in [0,1]^{10}, \end{cases}$$

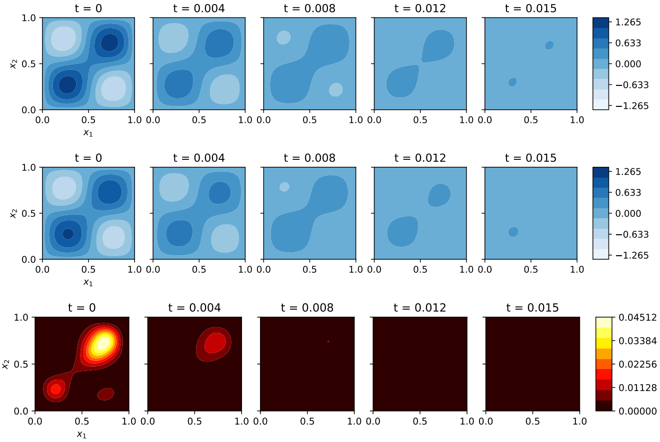
To have analytic examples to compare against, we tested our method against the following functions:

$$g_1(x) = \sin(2\pi x_1)\sin(2\pi x_2)\Pi_{i=3}^{10}\sin(\pi x_i) + 0.5\Pi_{i=1}^{10}\sin(\pi x_i),$$

$$g_2(x) = \sin(2\pi x_1)\sin(2\pi x_2)\Pi_{i=3}^{10}\sin(\pi x_i),$$

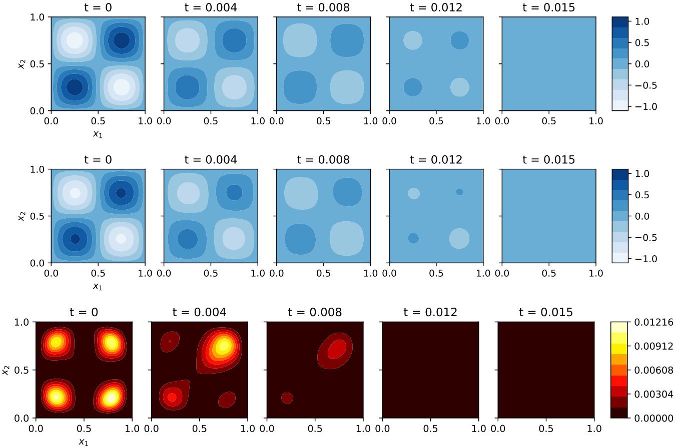
$$g_3(x) = \sin(2\pi x_1)\Pi_{i=2}^{10}\sin(\pi x_i).$$

Numerical Results: 10D Heat Equation



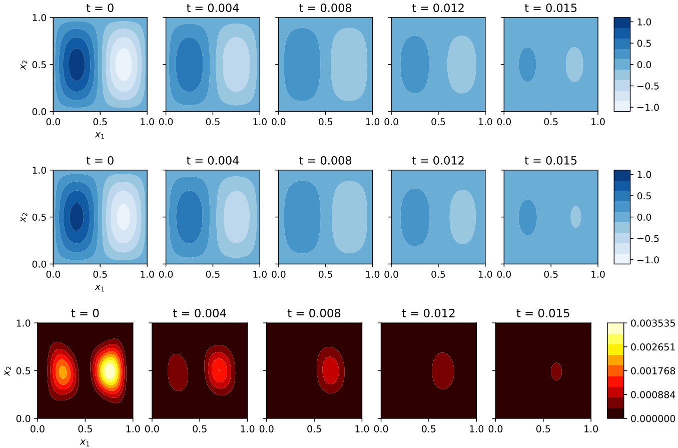
$$g_1(x) = \sin(2\pi x_1)\sin(2\pi x_2)\Pi_{i=3}^{10}\sin(\pi x_i) + 0.5\Pi_{i=1}^{10}\sin(\pi x_i)$$

Numerical Results: 10D Heat Equation



$$g_2(x) = \sin(2\pi x_1)\sin(2\pi x_2)\Pi_{i=3}^{10}\sin(\pi x_i)$$

Numerical Results: 10D Heat Equation



$$g_3(x) = \sin(2\pi x_1)\Pi_{i=2}^{10}\sin(\pi x_i).$$

Numerical Results: 10D Heat Equation

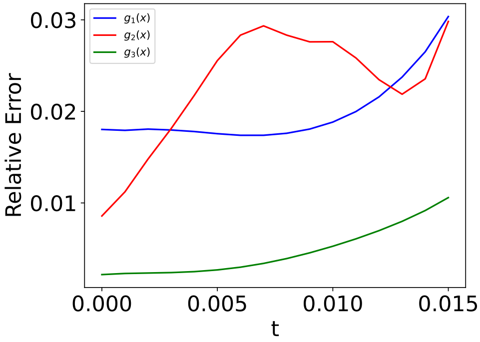


Figure: Comparison of the relative error $\|u^g(\cdot, t) - u_{\theta(t)}(\cdot)\|_{L^2(\Omega)}/\|u^g(\cdot, t)\|_{L^2(\Omega)}$ over time t for IVPs with initial values g_1, g_2 and g_3.

Numerical Results: 2D Allen-Cahn Equation

Consider the following 2D Allen-Cahn equation (we use the 2d Allen-Cahn as we lack analytic solution to compare to in high-dimensions):

$$\begin{cases} \partial_t u(x,t) = 10^{-3}\Delta u(x,t) + \frac{3}{2}\left(u(x,t) - u(x,t)^3\right), & \forall\, x \in (0,1)^2, t \in (0,0.6] \\ u(x,t) = 0 & x \in \partial(0,1)^2, t \in [0,0.6] \\ u(x,0) = g(x), & \forall\, x \in [0,1]^2 \end{cases}$$

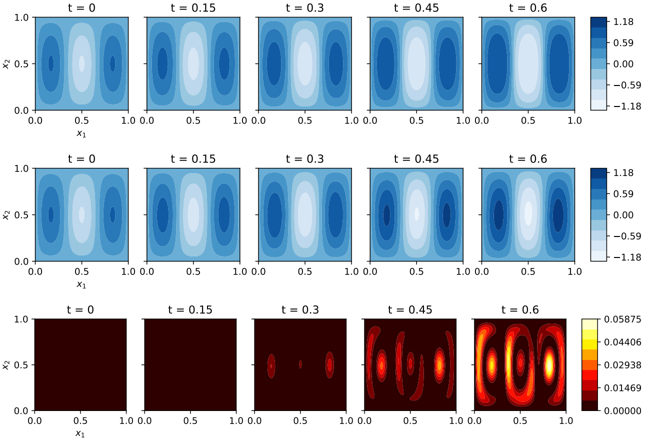
We compare to the following initials which are not contained in our training set as evidence for the model's success:

$$g_1(x) = 0.75\sin(3\pi x_1)\sin(\pi x_2)$$

$$g_2(x) = (x_1 - x_1^2)\cos(2\pi x_1)\sin(\pi x_2)$$

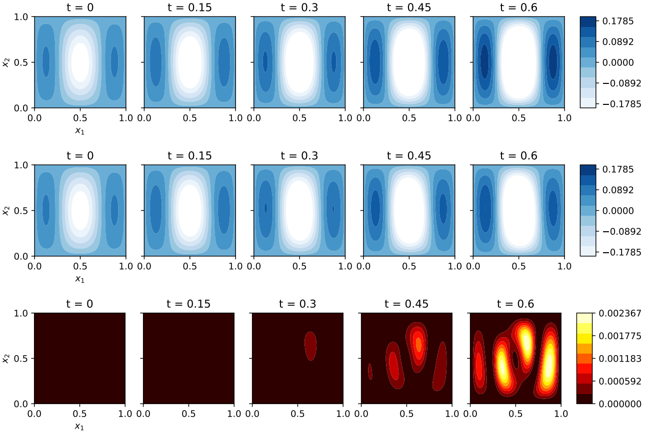
$$g_3(x) = (x_1 - x_1^2)\cos(2\pi x_1)\sin(2\pi x_2).$$

Numerical Results: 2D Allen-Cahn Equation



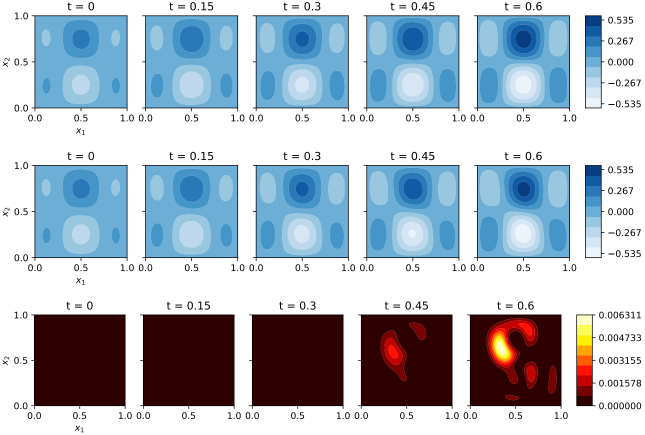
$$g_1(x) = 0.75 \sin(3\pi x_1) \sin(\pi x_2)$$

Numerical Results: 2D Allen-Cahn Equation



$$g_2(x) = (x_1 - x_1^2)\cos(2\pi x_1)\sin(\pi x_2)$$

Numerical Results: 2D Allen-Cahn Equation



$$g_3(x) = (x_1 - x_1^2)\cos(2\pi x_1)\sin(2\pi x_2).$$

Numerical Results: 2D Allen-Cahn Equation

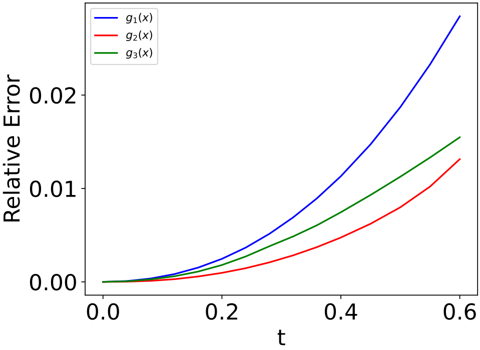


Figure: Comparison of the relative error.

Figure: Comparison of the relative error $\|u^g(\cdot,t) - u_{\theta(t)}(\cdot)\|_{L^2(\Omega)} / \|u^g(\cdot,t)\|_{L^2(\Omega)}$ over time t for IVPs with initial values g_1, g_2 and g_3.

Error Analysis

Assumption (Regularity)

The reduced-order model $u_\theta(\cdot) \in C^3(\Omega) \cap C(\bar{\Omega})$ *for every* $\theta \in \bar{\Theta}$ *and* $u(x;\cdot) \in C^1(\Theta) \cap C(\bar{\Theta})$. *Moreover, there exists* $L > 0$ *such that for all* $\theta \in \bar{\Theta}$

$$F[u_\theta] \in \mathcal{F}^L := \{f \in C^1(\Omega) \cap C(\bar{\Omega}) : \|f\|_\infty \leq L,\ \|\nabla f\|_\infty \leq L\}.$$

Assumption (Bounded projection error)

For any $\bar{\varepsilon} > 0$, *there exist a reduced-order model* u_θ *and a bounded open set* $\Theta \subset \mathbb{R}^m$, *such that for every* $\theta \in \bar{\Theta}$ *there exists a vector* $\alpha \in \mathbb{R}^m$ *satisfying*

$$\|\alpha \cdot \partial_\theta u_\theta - F[u_\theta]\|_{L^2(\Omega)} \leq \bar{\varepsilon}.$$

Error Analysis

Lemma (Uniform boundedness of approximate control (GYZ23))

Suppose the assumptions above are satisfied. For all $\varepsilon > \bar{\varepsilon}$ *there exists* $v : \bar{\Theta} \to \mathbb{R}^m$ *such that* v *is bounded over* $\bar{\Theta}$ *and*

$$\|v_\theta \cdot \partial_\theta u_\theta - F[u_\theta]\|_2 \leq \varepsilon.$$

Error Analysis

Proposition (Existence of neural control (GYZ23))

Suppose the assumptions above hold. Then for any $\varepsilon > 0$*, there exists a differentiable vector field parameterized as a neural network* $V_\xi : \bar{\Theta} \to \mathbb{R}^m$ *with parameter* ξ*, such that*

$$\|V_\xi(\theta) \cdot \partial_\theta u_\theta - F[u_\theta]\|_2 \leq \varepsilon,$$

for all $\theta \in \bar{\Theta}$*.*

Theorem (Error of controlled parametric solution (GYZ23))

- *Let* $F[u] = \nabla \cdot (A\nabla u) + b \cdot \nabla u + f(u)$ *where*
  - $z^\top A(x)z \geq \lambda |z|^2, \quad \forall z \in \mathbb{R}^d, \ x \in \Omega$,
  - $\|\nabla \cdot b\|_\infty \leq B, \ \lambda \geq 0$,
  - $f : \mathbb{R} \to \mathbb{R}$ *is* L_f*-Lipschitz.*
- *Suppose Assumptions 1 and 2 hold. Then:*
- *For all* $\varepsilon_0, \varepsilon > 0$ *and* $\|u_{\theta_0} - g\|_{L^2(\Omega)} \leq \varepsilon_0$, *there exists control field* V_ξ *such that*

$$\|u_{\theta_t}(\cdot) - u^g(\cdot, t)\|_{L^2(\Omega)} \leq e^{(L_f + B/2 - \lambda/C_p)t}(\varepsilon_0 + \varepsilon t),$$

  where
  - u^g *is a solution to (PDE) with initial* g,
  - $\theta(t)$ *is solved from (ODE) with* V_ξ *and initial* θ_0,
  - C_p *is a constant depending only on* Ω.

Summary and Future Work

Summary[1]

- We propose a new approach to numerically approximate solution operators of evolution PDEs;
- Our approach is particularly promising to tackle problems with high-dimension;
- We provide rigorous approximation error estimates of our method.

Future Work

- More effective ways to leverage useful θ in network training;
- Relaxation of assumptions on differential operators in proofs of error estimations;
- Applications to other interesting real-world problems.

[1] Preprint available at `https://arxiv.org/abs/2302.00045`.

Thank You