Machine Learning-Inspired Advancements in Density Functional Theory

Jiaxi Zhao (NUS)
joint work with Tianbo Li, Zekun Shi, and Min Lin @ SAIL
Giovanni Vignale and Qianxiao Li @ NUS

August 7, 2025
Job talk @ Bytedance

Research

- Applied mathematics Ph.D. @ National University of Singapore developing data-driven hybrid simulation methods for PDE problems
  - Mitigating distribution shift in machine learning-augmented hybrid simulation (SISC 2025)
  - Generative subgrid-scale modeling (MLMP ICLR 2025)
  - Numerical analysis on neural network projected schemes for approximating one dimensional Wasserstein gradient flows (JCP 2025)
  - Variational conditional normalizing flows for computing second-order mean field control problems
  - Numerical analysis for data-driven SGS modeling
- Research intern @ Sea AI Lab improving the JAX-based solid-state DFT solver *Jrystal* (core developer) using machine-learning toolbox
  - Amortized Eigendecomposition for Neural Networks (NeurIPS 2024)
  - Normalizing Flow Distorted Planewaves
  - Adaptive Gaussian basis for quantum chemistry calculation
  - Benchmarking occupation parametrizations in direct optimization approach for solid-state density functional theory

Three fundamental challenges of DFT

$$
\begin{aligned}
E(\{\psi\}) = & \sum_{i=1}^{N} \int \mathrm{d}\mathbf{r}_i \psi_i^*(\mathbf{r}_i)(-\frac{1}{2}\Delta - \sum_{\text{atom } a} \frac{Z_a}{|\mathbf{r}_i - \mathbf{A}_a|})\psi_i(\mathbf{r}_i) \\
& + E_{\mathrm{H}}[\rho] + E_{\mathrm{xc}}[\rho] + E_{\text{ion-ion}} \\
\rho(\mathbf{r}) = & \sum_{i=1}^{N} |\psi_i(\mathbf{r})|^2 \\
\delta_{ij} = & \int \psi_i^*(\mathbf{r})\psi_j(\mathbf{r}) d\mathbf{r}
\end{aligned}
$$

- Direct optimization method for DFT
- Molecular systems: complexity of the basis set $\Longrightarrow$ Adaptive Gaussian basis set
- Crystal system: high plane-waves cutoff for crystal system $\Longrightarrow$ Normalizing Flow Distorted Planewaves
- XC functional for materials $\Longrightarrow$ MLXC under PAW formulation

Adaptive Gaussian Basis for Quantum Chemistry Calculation

Jiaxi Zhao, Giovanni Vignale, and Min Lin

Basis sets are complex

- Slater basis functions (STO):

$$\varphi^{\mathrm{S}} \equiv \mathrm{poly}(x, y, z) e^{-\alpha \|\mathbf{r}-\mathbf{A}\|_2}$$

- Gaussian basis functions (GTO):

$$\varphi^{\mathrm{G}} \equiv \mathrm{poly}(x, y, z) e^{-\alpha \|\mathbf{r}-\mathbf{A}\|_2^2}$$

- Numerical atomic orbitals:

$$\varphi^{\mathrm{NAO}} \equiv \sum_{L}^{L_A} Y_L(\widehat{\mathbf{r}-\mathbf{A}}) f_L(\|\mathbf{r}-\mathbf{A}\|_2^2)$$

Basis set integrals are computationally expensive

$$\iint e^{-\alpha\|\mathbf{r}_1-\mathbf{A}\|_2^2-\beta\|\mathbf{r}_2-\mathbf{B}\|_2^2-\gamma\|\mathbf{r}_1-\mathbf{C}\|_2^2-\delta\|\mathbf{r}_2-\mathbf{D}\|_2^2}\frac{\text{poly}(\mathbf{r}_1,\mathbf{r}_2)}{r_{12}}d\mathbf{r}_1 d\mathbf{r}_2$$

- Boys' rule
- Classical recursive-based method: MD, HPG, Rys
- Higher angular-momentum integrals become very expensive to calculate
- **Basis set optimization** is also a hot topic for research

Adaptive Gaussian basis

Adaptive Gaussian basis (Kerbl et al. 2023) with optimizable mean and covariance:

$$\phi_i = \sum_{j \in I} c_{ij} \mathcal{N}(\mathbf{r}; \mu_j, \Sigma_j)$$

In general, all the optimizable parameters are $\mu \in \mathbb{R}^{|I| \times 3}, \Sigma \in \left(\mathbb{S}^3_+\right)^{|I|}, C \in \mathbb{C}^{|I| \times N}$. $|I|$ is the size of the basis set, N is the number of electrons

- Polynomial $\Longrightarrow$ more flexible covariance structure
- Analytic Gaussian integral

Orthonormality and overlap matrix

- The orthonormality of the orbitals in SCF is guaranteed by the eigensolver
- In total energy minimization, we have to handle this explicitly.
- Coefficient matrix C should satisfy:

$$\mathsf{C}^\dagger \, \mathsf{S} \, \mathsf{C} = \mathsf{I}, \qquad \mathsf{S}_{ij} = \langle \mathcal{N}(\mathbf{r}; \mu_\mathbf{i}, \Sigma_i) | \, \mathcal{N}(\mathbf{r}; \mu_\mathbf{j}, \Sigma_j) \rangle$$

The orthonormality is enforced by QR decomposition + Cholesky factorization of S

Coulomb and external energy

- No analytic formula for electron repulsion integral (ERI)

$$\langle \mathcal{N}(\mathbf{r}_1|\mu_1, \Sigma_1)| \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} |\mathcal{N}(\mathbf{r}_2|\mu_2, \Sigma_2)\rangle$$

- Approximate the Coulomb kernel by a series of Gaussian modes:

$$\frac{1}{r} = \sum_i c_i e^{-\alpha_i r^2}$$

- Minimize the following

$$\min_{c_i, \alpha_i} \int_{B_M} \left(\frac{1}{|\mathbf{r}|} - \sum_i c_i e^{-\alpha_i |\mathbf{r}|^2} \right)^2 d\mathbf{r} = 4\pi \int_0^M \left(1 - \sum_i c_i r e^{-\alpha_i r^2} \right)^2 dr$$

More on double electron integrals

$$\langle \mathcal{N}(\mathbf{r}_1|\mu_1,\Sigma_1)| \frac{1}{|\mathbf{r}_1-\mathbf{r}_2|} |\mathcal{N}(\mathbf{r}_2|\mu_2,\Sigma_2)\rangle$$

$$= \int\int d\mathbf{r}_1 d\mathbf{r}_2 \frac{1}{\sqrt{(2\pi)^6 \det(\Sigma_1)\det(\Sigma_2)}} \sum_i c_i \exp\left[-\frac{1}{2}\left((\mathbf{r}_1-\mu_1)^T\Sigma_1^{-1}(\mathbf{r}_1-\mu_1)\right.\right.$$

$$\left.\left.+(\mathbf{r}_2-\mu_2)^T\Sigma_2^{-1}(\mathbf{r}_2-\mu_2)\right)-(\mathbf{r}_1-\mathbf{r}_2)^T\alpha_i\mathsf{I}(\mathbf{r}_1-\mathbf{r}_2)\right]$$

$$= \int\int d\mathbf{r} \frac{\sum_i c_i s(i) \exp\left[-\frac{1}{2}(\mathbf{r}-\mu(i))^T \begin{pmatrix} \Sigma_1^{-1}+2\alpha_i\mathsf{I} & -2\alpha_i\mathsf{I} \\ -2\alpha_i\mathsf{I} & \Sigma_2^{-1}+2\alpha_i\mathsf{I} \end{pmatrix} (\mathbf{r}-\mu(i))\right]}{\sqrt{(2\pi)^6 \det(\Sigma_1)\det(\Sigma_2)}}$$

- Solving a linear system of size 3×3, MVP of size 3×3, calculating the determinant of a 3×3 matrix

Parameterization of Gaussian basis

- Naive implementation:

$$\Sigma = MM^T$$

- Eigen parametrization:

$$\Sigma = \mathrm{U}\,\mathrm{D}\,\mathrm{U}^T$$

U_i: QR decomposition, D_i: softplus function

The core contribution of this work

- A self-contained integral pipeline for the Hartree-Fock calculation using the Gaussian basis functions with more flexible covariance
- A complete unit test to verify the correctness of the integral
- Efficient Fock build algorithm: single ERI FLOPs count around $60 \times 10 = 600$, **regardless of the orbital type!**
- Classical Fock build algorithm: (ss|ss), (ps|ps), (pp|pp) requires 33, 58, 1326 FLOPs for a single evaluation (Barca et al 2020, 2021)

Numerical experiments

	e_kin + e_ext	e_kin	e_ext	e_nuc	e_coul + e_exc	e_coul	e_exc	e_tot
OUR (H_2), 10	-2.5062	1.1254	-3.6316	0.7138	0.6591	1.3182	-0.6591	-1.1334
HF (STO-3G, 2)	-2.5049	*	*	0.7138	0.6745	*	*	-1.1167
HF (6-31G, 4)	-2.4902	*	*	0.7138	0.6497	*	*	-1.1267
HF (6-311G, 6)	-2.4924	*	*	0.7138	0.6507	*	*	-1.1280
OUR (CH_4), 22	-79.6713	40.0987	-119.7700	13.4477	26.1010	32.6936	-6.5926	-40.1225
HF (STO-3G, 9)	-79.3617	*	*	13.4477	26.1872	*	*	-39.7267
HF (6-31G, 17)	-79.6901	*	*	13.4477	26.0619	*	*	-40.1804
HF (6-311G, 25)	-79.6872	*	*	13.4477	26.0515	*	*	-40.1880
OUR (H_2O), 22	-122.9802	75.9462	-199.0371	9.1895	37.8786	46.8579	-8.9601	-76.0036
HF (STO-3G, 7)	-122.3614	*	*	9.1895	38.2089	*	*	-74.9630
HF (6-31G, 13)	-122.9701	*	*	9.1895	37.7966	*	*	-75.9839
HF (6-311G, 19)	-123.0146	*	*	9.1895	37.8157	*	*	-76.0094

Normalizing Flow Distorted Planewaves

Zekun Shi, Jiaxi Zhao, and Giovanni Vignale

A primer on planewave based solid-state DFT

- Bloch's theorem: $\phi_j(\mathbf{r}) = e^{i\mathbf{k_j}\cdot\mathbf{r}} u_j(\mathbf{r})$, $u_j(\mathbf{r})$ periodic in the unit cell
- Basis functions: plane-waves $e^{i\mathbf{k}\cdot\mathbf{r}}$, $\mathbf{k}$ lattice vector
- Double electron integral $\Longrightarrow$ Poisson equation (solved by FFT):

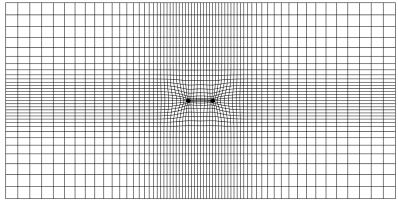
$$E_{\text{Coulomb}} = \frac{1}{2}\int_{\Omega} d\mathbf{r} \int_{\mathbb{R}^3} d\mathbf{r}' \frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|} d\mathbf{r} d\mathbf{r}',$$

$$E_{\text{Coulomb}} = \frac{1}{2}\int_{\Omega} d\mathbf{r}\rho(\mathbf{r}) V_{\text{H}}(\mathbf{r}) d\mathbf{r}, \quad \nabla^2 V_{\text{H}} = -4\pi\rho(\mathbf{r})$$

- Ewald summation for nucleus repulsion energy
- Systematical approach to obtain more accurate DFT calculation – increase the plane-wave energy cutoff

Literature

- Orbital peaks around atoms $\Longrightarrow$ High planewaves cutoff
- Use fixed analytic transformation mapping depending on the position of the atom: Gygi (1993), Zumbach et al. (1996)

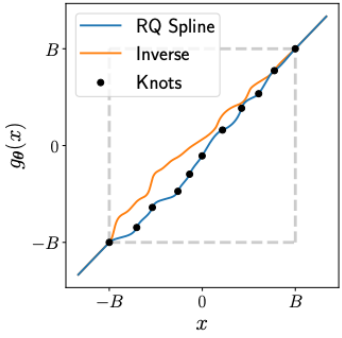


- Solve some OT-type problems to predetermine the grid before the SCF loop: Lindsey and Sharma (2024)

Thought: the design of the distorted grid has great freedom and has to be done by expert for each crystal system – can we have a more adaptive and automatic design of the distorted grid?

Normalizing flow on periodic domain

- Invertible mapping on S^1 parametrized by spline function, Durkan et al. (2019)



- Coupling layer to handle 3D domain:

$$\begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix} \xrightarrow[x_2^{(1)} = x_2^{(0)}]{x_1^{(1)} = f(x_1^{(0)}, \mathrm{NN}_2^{(1)}(x_2^{(0)}))} \begin{pmatrix} x_1^{(1)} \\ x_2^{(1)} \end{pmatrix} \xrightarrow[x_2^{(2)} = f(x_2^{(1)}, \mathrm{NN}_2^{(1)}(x_1^{(1)}))]{x_1^{(2)} = x_1^{(1)}} \begin{pmatrix} x_1^{(2)} \\ x_2^{(2)} \end{pmatrix}$$

- Autoregressive layer to improve expressivity

Distorted plane-wave basis

- Given invertible mapping $f\colon \Omega \to \Omega$, define distorted planewaves (DPW) $\phi_{\mathbf{G}}$ as planewaves in the parameter space:

$$\langle \mathbf{r}|\mathbf{G}\rangle := \phi_{\mathbf{G}}(\mathbf{r}) = \frac{1}{\sqrt{|\Omega|}} |J_{f^{-1}}(\mathbf{r})|^{\frac{1}{2}} \exp\left[\mathbf{G}^{\top} f^{-1}(\mathbf{r})\right]$$

- Basis are orthonormal by change of variable formula
- Bloch factor only affects the kinetic energy

Kinetic matrix element: FFT

- Matrix element involves Jacobian of the mapping:

$$\begin{aligned}&\langle \psi_{i,\mathbf{k}} | -\frac{1}{2}\nabla^2 | \psi_{j,\mathbf{k}} \rangle \\ =&\frac{1}{2}\sum_{\mathbf{G}',\mathbf{G}} c^*_{i,\mathbf{k},\mathbf{G}'} c_{j,\mathbf{k},\mathbf{G}} \{ B_{\mathbf{G}',\mathbf{G}} - G_q [U_{\mathbf{G}',\mathbf{G}}]_q + G'_q [U_{\mathbf{G}',\mathbf{G}}]_q + (G_p + k_p)(G'_q + k_q)[P_{\mathbf{G}',\mathbf{G}}]_{pq} \}\end{aligned}$$

- B, U, P are all block-Toeplitz
- FFT implementation of the Toeplitz matrix-vector product

Hartree energy: solving the Poisson equation

$$\nabla^2 V = -4\pi\rho$$

- Solved by FFT in plane-wave basis
- Solved by physics-informed neural network (PINN) — introducing an inner loop of optimization
- The loop of PINN needs not to be converged for each step, amortizing with the outer loop

Similar philosophy as the previous adaptive Gaussian basis.

Overall algorithm

- **Input:** Crystal system parameters, init params of NF, PINN, coefficient
- **While** E_{total} not converged:
  - Calculate the Jacobian of the NF at each grid point
  - Calculate the kinetic and XC energy
  - **For** $i = 1, 2, \ldots, N_{\text{PINN}}$:
    - Sample the collocation points $\{\mathbf{r}_i\}$ for PINN evaluation
    - Calculate the PINN loss: $\frac{1}{N}\sum_{i=1}^{N}(\nabla^2 V_\theta(\mathbf{r}_i) + 4\pi\rho(\mathbf{r}_i))^2$
    - Backpropogate to update the PINN parameters
  - Calculate the Hartree energy
  - Backpropogate through the total energy
- **Output:** Total energy E_{tot}, params of NF, PINN, coefficient

Numerical results

Band structure calculation for diamond with LDA. All energies are in eV unit.

Method	FFT grid size	band gap	L	X	Γ
PW	128	3.05869	7.51526	3.05869	4.79634
PW	96	3.14795	7.61833	3.14795	4.77946
PW	64	3.68519	7.72595	3.68519	4.88759
PW + ANC	96	3.08192	7.7782	3.08192	4.75455
PW + ANC	64	3.10424	7.77621	3.10424	4.88761
PW + ANC	48	1.98104	7.39363	1.98104	4.84532
FDPW + PINN	48	3.01882	7.26194	3.01882	4.41584

ANC: Gygi (2023), a light-weight pseudopotential

Summary of contributions

- DPW via normalizing flow with orthonormal property
- Calculation of the density related energy can be estimated efficiently by the Monte-Carlo method:

$$E_{\text{xc}} = \int_{\Omega} \epsilon_{\text{xc}}^{\text{LDA}}(\rho(\mathbf{r}))\rho(\mathbf{r})d\mathbf{r} = \frac{1}{N}\sum_{i=1}^{N} \epsilon_{\text{xc}}^{\text{LDA}}(\rho_i)$$

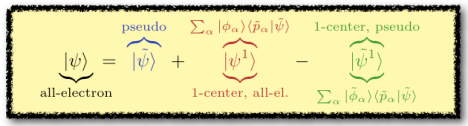
- Comparing to results like Gygi (1993) and Lindsey and Sharma (2024), our method generates the grid on the fly and adatively during the optimization loop
- Reduce the total grid size by 8 times on small crystal systems

Machine-Learning Exchange-Correlation Functional under Projector Augmented-Wave Method

- Implement PAW differentiably in *Jrystal* package
- Implement MLXC under PAW formulation

PAW is not a pseudopotential method! — Blöchl

- Atomic region $\Longrightarrow$ precomputed radial grid atomic DFT
- Interatomic region $\Longrightarrow$ plane-wave basis
- A clever way to glue together above two parts of the wavefunctions (PAW)



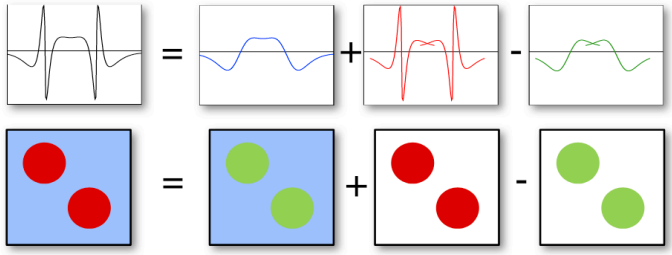


Figure: From Blöchl's slide

How PAW glue together atomic & interatomic region

Wavefunctions: $\psi = \widetilde{\psi} + \sum_a (\psi^a - \widetilde{\psi}^a)$

Density: $\rho = \widetilde{\rho} + \sum_a (\rho^a - \widetilde{\rho}^a)$

Energy: $E_{\text{kin}} = \widetilde{E}_{\text{kin}} + \frac{1}{2} \sum_{a,i,j} D^a_{ij} \int_{\mathbb{S}^a} \left(\nabla \phi^{a*}_i(\mathbf{r}) \nabla \phi^a_j(\mathbf{r}) - \nabla \widetilde{\phi^a_i}^*(\mathbf{r}) \nabla \widetilde{\phi}^a_j(\mathbf{r}) \right) d\mathbf{r}$

- PAW is an all-electron theory with rigorous mathematical formulation and controllable errors.

Alignment platform with GPAW

- Understand the mathematical derivation of the PAW method
- Handle the difference of the pseudopotential file systems used by ours and GPAW
- Design of the unit test in the code development of the PAW
- Fully differentiable PAW implementation, suitable for future MLXC applications

MLXC for molecules v.s. materials

- Golden standard computational method for molecules, CCDS(T) v.s. QMC for materials?
- Various datasets for molecules: QM9, etc v.s. MP for materials
- Pseudopotential is always included in the planewave calculations for material
- MLXC on molecules systems: DeepKS (Chen et al. 2020), DM21 (Kirkpatrick et al. 2021), Skala (Luise et al. 2025)

The drawback of MLXC under NCPP & USPP

- Standard way to calculate XC energy:

$$E_{\text{xc}} = E_{\text{xc}}[\widetilde{\rho}]$$

- With nonlinear core correction:

$$E_{\text{xc}} = E_{\text{xc}}[\widetilde{\rho} + \widetilde{\rho}^{\text{core}}] - E_{\text{xc}}[\widetilde{\rho}]$$

- The XC potential is included in the local part of the pseudopotential and is unchanged even using different XC for calculation

Explicitly handle the XC energy

$$E_{\mathrm{xc}} = E_{\mathrm{xc}}[\widetilde{\rho}] + \sum_a (E^a_{\mathrm{xc}}[\rho] - E^a_{\mathrm{xc}}[\widetilde{\rho}])$$

- The pseudopotential generating process and PAW calculation process are "decoupled"
- $E_{\mathrm{xc}}[\widetilde{\rho}]$: the XC energy for the pseudo-density, integrated over uniform grid
- $E_{\mathrm{xc}}[\rho^a] - E_{\mathrm{xc}}[\widetilde{\rho}^a]$: the XC energy difference of the on-site part

How to train the MLXC functional?

- High accuracy label for the crystal data is extremely sparse and expensive
- Training over the labels generated by the hybrid or double hybrid functional, e.g. HSE06

$$\min \mathbb{E}(E^{\mathrm{HSE06}} - E^{\mathrm{MLXC}})^2, \ \min \mathbb{E}(E_{\mathrm{xc}}^{\mathrm{HSE06}} - E_{\mathrm{xc}}^{\mathrm{MLXC}})^2$$

- Training over the set of stable materials:

$$\min \mathrm{Tr}(\nabla_{\mathbf{r}} E_{\mathrm{MLXC}}(\mathbf{r}))$$

An abstraction of SciML workflow

Simulating the dynamics:

$$\mathcal{L}(\mathbf{u}, \partial_t \mathbf{u}, \mathbf{y}, t) = \mathbf{0}, \quad \mathbf{y} = \phi(\mathbf{u}, t).$$

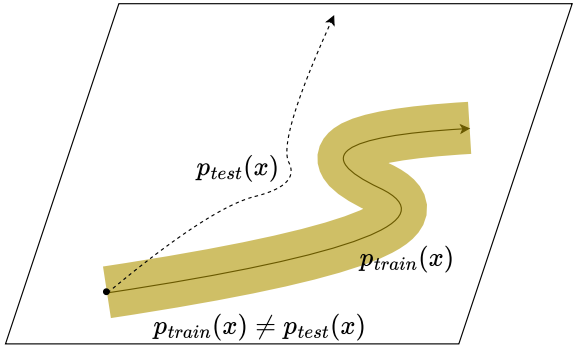
- $\mathcal{L}$ is known, possibly non-linear, while ϕ is un-known
- Datasets: $\{(\mathbf{u}_1, \mathbf{y}_1, t_1), (\mathbf{u}_2, \mathbf{y}_2, t_2), \cdots, (\mathbf{u}_N, \mathbf{y}_N, t_N)\}$
- Benchmark algorithm solves the ordinary least square:

$$\arg\min_{\theta} \mathbb{E} \left\| \mathbf{y} - \phi_\theta(\mathbf{u}, t) \right\|^2$$

Typical examples: subgrid-scale modeling, reynolds stress modeling, exchange-correlation functional

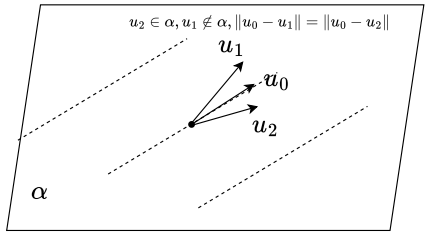
A-priori and a-posteriori discrepancy

- Models performing well a-priorily may not perform well in the simulation
- Training algorithm does not take the solver dynamics into account



Alg 1: Manifold regularization[1]

- Regularization encodes the information of the data manifold

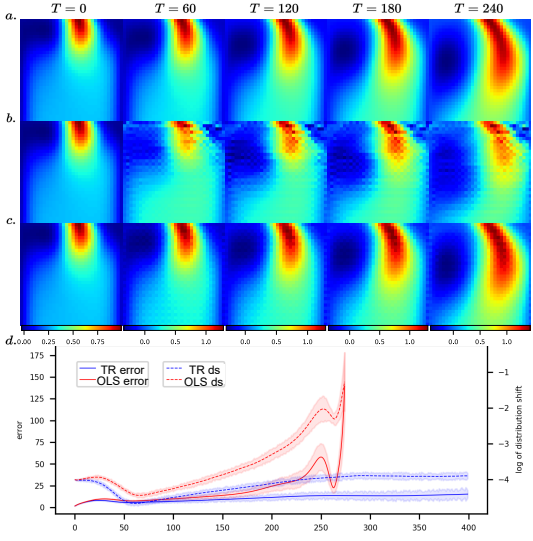


- For nonlinear case, learn a function $F(\mathbf{u})$ which has data manifold as a level set

$$I_{\mathrm{TR}}(\theta) := \mathbb{E}_{(\mathbf{u},\mathbf{y})}\left[\|\mathbf{y}_k - \phi_\theta(\mathbf{u})\|_2^2 + \lambda\left(\left(\nabla F(\mathbf{u})\right)^T L(\mathbf{u}, \phi_\theta(\mathbf{u}), t)\right)^2\right]$$

[1] Zhao, Jiaxi, and Qianxiao Li. "Mitigating Distribution Shift in Machine Learning–Augmented Hybrid Simulation." SIAM Journal on Scientific Computing 47.2 (2025): C475-C500.

Performance



Alg 2: Probabilistic ansatz[2]

Regression to generative modeling:

$$\tau = \phi_\theta(\mathbf{u}) \quad \rightarrow \quad \tau \sim p_\theta(\cdot|\mathbf{u})$$

Change of the loss functions:

$$\min_\theta \sum_n \left\| \phi_\theta(\widetilde{\mathbf{u}}^{(n)}) - \tau^{(n)} \right\|^2, \quad \max_\theta \sum_{i=n}^{N} \log p_\theta(\tau^{(n)}|\mathbf{u}^{(n)})$$

[2] Zhao, Jiaxi, Sohei Arisaka, and Qianxiao Li. "Generative subgrid-scale modeling." ICLR 2025 Workshop on Machine Learning Multiscale Processes

Alg 3: Correction-based SGS models

Considering a PDE with both linear and nonlinear terms

$$\partial_t \overline{u} + \mathcal{L}(\overline{u}) + \mathcal{N}(\overline{u}) + \tau(\overline{u}) = 0$$

- Filter-based approach:

$$\mathcal{F}(u^f) = u^c \Longrightarrow \tau^{(1)} = \mathcal{F}(\mathcal{N}(u^f)) - \mathcal{N}(u^c)$$

- Correction-based approach:

$$\mathcal{F}(u^f) = u^c \Longrightarrow \tau^{(2)} = \mathcal{F}(\text{solver}^f(u^f)) - \text{solver}^c(u^c) \tag{1}$$

Thank you for your attention! Q & A

- Email: jiaxi.zhao@u.nus.edu
- WeChat: