
From Black Hole to Galaxy: Neural Operator Framework for Accretion and Feedback Dynamics

Nihaal Bhojwani^{1,2*}, Chuwei Wang^{2*}, Hai-Yang Wang^{3*}, Chang Sun⁴,
Elias R. Most³, Anima Anandkumar²

¹Department of Computer, Mathematical, and Natural Sciences, University of Maryland,

²Department of Computing and Mathematical Sciences, California Institute of Technology,

³TAPIR & Walter Burke Institute for Theoretical Physics, California Institute of Technology

⁴Department of Physics, California Institute of Technology

`nbhojwan@umd.edu, {chuweiw, haiyangw, chsun, emost, anima}@caltech.edu`

Abstract

Modeling how supermassive black holes co-evolve with their host galaxies is notoriously hard because the relevant physics spans nine orders of magnitude in scale—from milliparsecs to megaparsecs—making end-to-end first-principles simulation infeasible. To characterize the feedback from the small scales, existing methods employ a static subgrid scheme or one based on theoretical guesses, which usually struggle to capture the time variability and derive physically faithful results. Neural operators are a class of machine learning models that achieve significant speed-up in simulating complex dynamics. We introduce a neural-operator-based “*subgrid black hole*” that learns the small-scale local dynamics and embeds it within the direct multi-level simulations. Trained on small-domain (general relativistic) magnetohydrodynamic data, the model predicts the unresolved dynamics needed to supply boundary conditions and fluxes at coarser levels across timesteps, enabling stable long-horizon rollouts without hand-crafted closures. Thanks to the great speedup in fine-scale evolution, our approach for the first time captures intrinsic variability in accretion-driven feedback, allowing dynamic coupling between the central black hole and galaxy-scale gas. This work reframes subgrid modeling in computational astrophysics with scale separation and provides a scalable path toward data-driven closures for a broad class of systems with central accretors.

1 Introduction

Background. Supermassive black holes (SMBHs) co-evolve with their host galaxies through a two-way feeding-feedback loop that operates across nine orders of magnitude in scale [1]. Accretion flows near the event horizon (milliparsec scales) are governed by general relativistic magnetohydrodynamics (GRMHD) and launch relativistic jets and winds. These outflows propagate from AU to megaparsec scales, well beyond the host galaxy, depositing energy and momentum into the interstellar and intergalactic media [2, 3]. The resulting feedback regulates star formation and galaxy growth, while galaxy-scale gas dynamics in turn modulate the inflow feeding the central black hole.

Challenge. This multiscale coupling makes end-to-end simulation computationally intractable: accurately resolving accretion flows demands timesteps set by the gravitational radius ($r_g \sim \text{mpc}$ for M87), while capturing galaxy-scale feedback requires following dynamics over $\sim 10^9$ times larger spatial and temporal scales. Bridging these domains requires passing information across many orders of magnitude in space and time without washing out variability or violating conservation.

*Equal contribution. In alphabetical order.

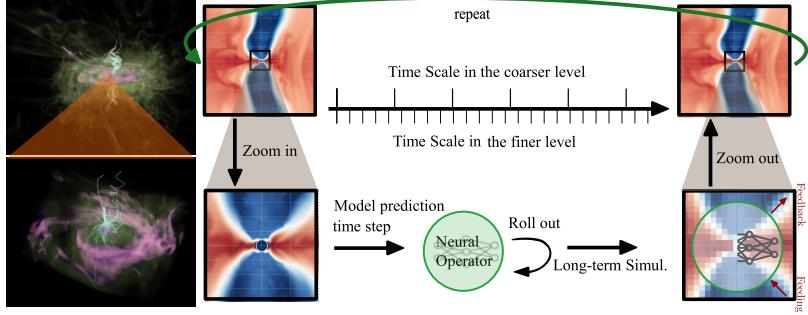


Figure 1: Illustration of our method in a two-level setting. The neural operator efficiently replaces the evolution on small scales (zoom in), enabling proper feedback on large scale (zoom out).

A considerable amount of efforts adopting various techniques has been made to bridge different scales, including direct simulations under reduced scale separation [4–7], nested mesh “zoom-in” approaches in Eulerian codes [8, 9], “Lagrangian hyper-refinement” in Lagrangian codes [10, 11], remapping between domains [12], and iterative “multi-zone” [13–15] or “cyclic-zoom” [16] schemes that repeatedly refine/de-refine around the SMBH. Among these, multi-zone and cyclic-zoom methods propagate small-scale feedback to larger domains and evolve to consistency, but can still struggle with temporal variability and with specifying a physically faithful inner boundary (here we name this by *subgrid black hole*) across levels. The inner boundary adopted this way could feed erroneous information (e.g., in both methods, the relativistic jet cannot be turned on/off when crossing the boundary) to the active simulation domain, consequently alter the final results.

Neural Operator. Neural operators (NOs) represent a recent advance in machine learning for partial differential equations, learning mappings between infinite-dimensional function spaces rather than approximating individual functions [17, 18]. This approach has demonstrated substantial computational acceleration and shown promise as surrogate models across various applications [19–24]. Recent studies have begun exploring NOs for coarse graining or closure modeling problems [23]. Astrophysical systems present unique challenges. Realistic galaxy formation involves large-scale, strongly coupled dynamics with central singularities at supermassive black holes, requiring stable long-term integration over cosmological timescales. The extreme dynamic range, combined with the inherent scarcity of training data in astrophysical simulations, creates a particularly demanding test case for NO or other machine learning methods. To the best of our knowledge, despite some preliminary attempts in simplified systems or transient-state predictions [25–28], no previous work has demonstrated NO performance in such complex, multi-scale systems with limited training data.

Contribution. We introduce a neural-operator-based “*subgrid black hole*” coupled to a direct multi-level solver, replacing hand-crafted closure rules with a data-driven model that captures realistic variability in accretion and feedback (see fig. 1 for a schematic illustration of our method):

(1) *Operator-learning subgrid*. We train a neural operator to approximate the small-scale (GR)MHD time-evolution semigroup. The learned model supplies boundary conditions and fluxes to the next coarser level, enabling long-horizon rollouts without assuming steady or time-averaged injection. (2) *Two-way multiscale coupling*. Embedded within a multi-level framework, the learned subgrid responds to evolving large-scale conditions and, in turn, drives feedback that propagates outward—preserving variability critical to galaxy–SMBH co-evolution. (3) *General applicability*. The demonstrated approach is broadly applicable as a subgrid model for systems with a central accretor (e.g., SMBHs and neutron stars), independent of problem setups and generalizable to different codes.

This work reframes subgrid modeling in computational astrophysics as *operator learning*: rather than prescribing fixed closures, we learn the small-scale dynamics to provide dynamically-updated boundary conditions for larger domains. This advance in subgrid modeling fills up the missing piece for the black hole accretion and feedback problem. More importantly, this could revolutionize the modeling of black hole feedback in cosmological simulations such as FIRE [29] and IllustrisTNG [30].

Related Works. Analytical subgrid prescriptions remain the standard tool in large-scale such as cosmological simulations (e.g., supernova feedback, cold-gas processes, and black-hole growth [31, 32]). Machine-learning-based subgrid/surrogate models are only beginning to appear, with early results for supernova feedback [33, 34] and for 2D black-hole accretion surrogates [35] trained on

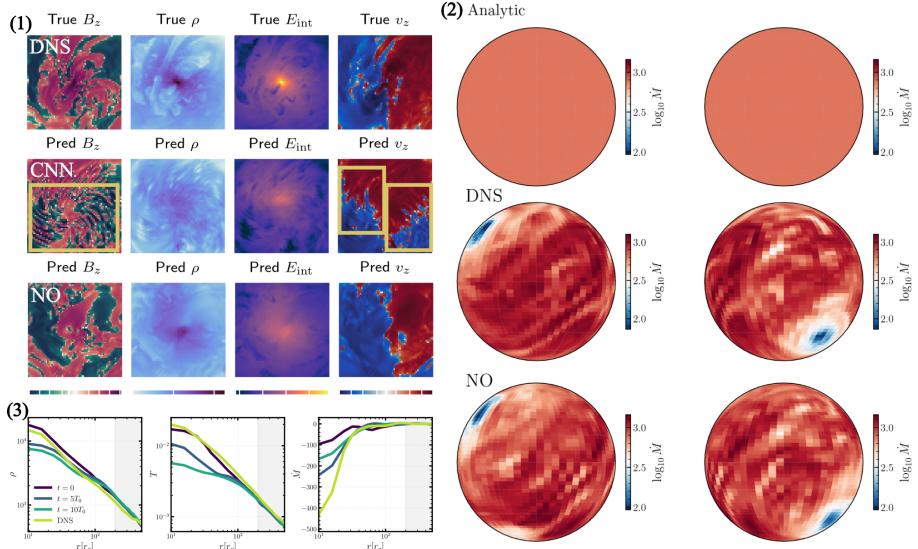


Figure 2: **MHD outputs.** Simulated (‘DNS’), CNN-predicted, and NO-predicted magnetic field component B_z , density ρ , internal energy E_{int} , and velocity component v_z from the MHD simulation on a horizontal slice through the domain center (after 50 steps). Unphysical artifacts are marked. (2) Comparison of the mass density distributions on a 2D sphere, between analytically prescribed, directed simulated, and NO-rollout (after 10 steps). (3) The radial profiles of predicted quantities: mass density ρ , temperature T , and total mass flux \dot{M} (*Left to Right*). Profile from simulation after 50 steps is also shown. The grey-shaded region shows the part that will be coupled to the direct simulation (see Appendix E).

Newtonian hydrodynamical simulation of steadily accreting torus. Our work advances this line by learning an operator-level “subgrid black hole” compatible with multi-level MHD/GRMHD.

2 Method

Method Framework. We consider the (general relativistic) magnetohydrodynamics (GRMHD) equations on a bounded spatial domain $\Omega \subset \mathbb{R}^3$ over a finite time interval. See Appx. A for governing equations and initial conditions. We introduce a two-level neural-operator–direct-numerical-simulation framework for the accretion–feedback problem with a “*subgrid black hole*” model. Our conventions are: (i) **domain decomposition**: a coarse level (with domain size $(n_L L)^3$) and a fine level (with domain size $(L)^3$) are defined, n_L is a large number which we choose to be 6; (ii) **domain treatment**: within affordable simulations, the coarse level is evolved via DNS, while the fine level is unresolved and represented by a subgrid model; (iii) **domain coupling**: resolving the fine level is ultimately necessary to address the multiscale problem discussed in Sec. 1. The framework is summarized in Fig. 1.

Directly simulating the fine and coarse levels together would shorten the timestep, thus increasing the cost, by a factor of at least n_L^2 . Instead, we first simulate the fine level over $t \in [0, T]$, producing $N_{\text{data}} = 300$ snapshots (saved every $\Delta T := T/N_{\text{data}}$) as the training set. We then train a local neural operator [36] using the fine level simulation data. This neural operator is coupled to the coarse level, which is subsequently simulated over $t \in [0, NT]$, with $N \sim 10^2$ in our current setup.

The direct simulation for 50 timestep (T/N_{data}) in the current time unit will cost 400 GPU hours. With neural operator inference which takes only a few GPU sec, we achieve a speed up of $\sim 10^5$ x.

Model Training. With supervised manner, we train a neural operator (LocalNO [36]) \mathcal{F}_θ to learn the mapping $u_t \rightarrow u_{t+\Delta T}$, where u represents all the physical quantities in the system (appx. A). After training, we employ the neural operator and roll out autoregressively in the fine level, mimicking the quasi–steady state attained on small scales as $t \rightarrow \infty$.

The large-scale coupled dynamics studied in this work pose significant challenges for training: (1) extreme dynamic range in function values; (2) scarcity of training data; (3) singularities and highly

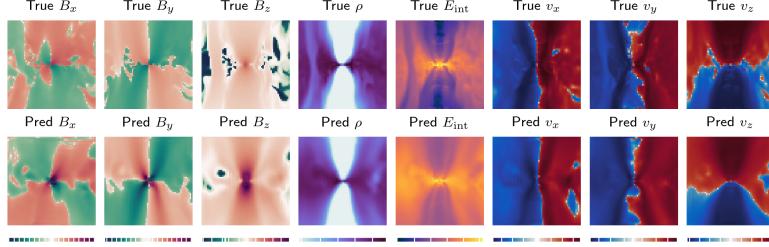


Figure 3: **GRMHD outputs.** Simulated and predicted magnetic field components (B_x, B_y, B_z), density ρ , internal energy E_{int} , and velocity components (v_x, v_y, v_z) from the GRMHD simulations. Jet structure near the polar region and disk structure near the midplane is preserved.

non-uniform spatial distributions near the black hole center; (4) long-horizon rollouts in chaotic dynamics; (5) coupling across multiple physical fields and PDEs.

We introduce a suite of training techniques specifically designed to overcome these challenges. The general principle is to incorporate physics prior to enhance the data efficiency and long-term stability. (1) **Magnitude normalization:** we apply (signed-) logarithmic transform for quantities with large dynamic range, i.e., density, energy, and magnetic field. After that we apply robust z-scoring and soft clipping to normalize the data. (2) **Enforcing radial scaling:** It is known that energy and density display a radial scaling law: $\log u(\vec{x}) \approx -k|\vec{x}| + b$, where k can be estimated from the training data. In our experiments, the models are trained to predict the residual relative to this law: $\log(\text{Final Output})(\vec{x}) = -k|\vec{x}| + \mathcal{N}^{-1}(\text{Model Output})(\vec{x})$, where \mathcal{N} is the denormalization. This reduces the dynamic range the model needs to learn, improving accuracy and ensuring physical consistency. (3) **Shell embedding:** Since the majority of the fluctuations of the functions occur near the black hole center, we employ a shell embedding to ensure the model to focus on the central region. We divide the domain into 8 regions based on the logarithmic distance to the center, and incorporate one-hot embedding of binary shell indicators $c \in \{0, 1\}^8$. We find this to perform better than standard position embedding. (4) **Combined loss functions:** In addition to standard L^2 and H^1 losses for all eight quantities, we add regularizations on dissipation [37] and the maximum of density and energy model prediction to prevent severe deviation from the radial scaling. A region-of-interest (ROI) mask further increases the loss weight near the black hole center [38].

Training takes 10 GPU hours, far less than the cost of direct numerical simulations. More technical details in this section and implementations are presented in Appendices A and C.

3 Results

For both MHD and GRMHD, we train the model with the last 250 snapshots of the dataset and use an 80/20 train/validation split. We also train a CNN as baseline method, and conduct ablation study to validate the effect and necessity of training techniques we mentioned above. See appx. D for details.

Evaluation. Recall that the ultimate goal in this study is to develop a *subgrid blackhole* providing realtime feedback for the large scale (coarse level), which requires long-term simulation in the fine level. Standard metric like L^2 relative error quantifying transient-state prediction does not make much sense here, as error will inevitably accumulate in these chaotic dynamics. In contrast, the most important merit is the ability to capture the fine-level statistical property well. As is the case, all the baseline learning methods in our ablation study and CNN achieve similar accuracy for per-step prediction, but none of these methods except the one we proposed generate unphysical artifacts after long rollouts (Table 2, Appx. D). Moreover, the absence of ground-truth (due to the intractable computation cost) for both long-term simulation in the fine level and simulation with real-time feedback in the coarse level suggests we can only make qualitative comparison and check certain observables to validate our method.

MHD Qualitative outputs are shown in Fig. 2 (1). As boxed yellow in the figure, predictions made by CNN contain unphysical ripples in the magnetic field, and the velocity field does not match the torus structure. Our method derive physically faithful results. In Fig. 2 (2), analytical prescribed subgrid scheme fails to capture the jet, demonstrating the importance of having a data-driven dynamical closure as in our approach. In Fig. 2 (3), the observables from our method match the truth well. Note that in our sub-grid blackhole framework, the fine level supplies the boundary for coarse level, thus

the model prediction is applicable as long as it provides accurate results near the outer boundary in the fine-level simulation.

GRMHD Qualitative outputs are shown in Fig. 3. Jet structure and central torus are preserved.

Subgrid Black Hole in Simulation The implementation detail and result for zooming back to the coarse-level are described in appx. E and fig. 6.

Our approach can serve as a first step (two-level version) towards a multi-level cyclic-zoom/multi-zone neural operator framework, enabling large-scale simulation with dynamic feeding-feedback that is previously intractable.

Acknowledgement The authors are grateful for discussions with Philip Hopkins. HYW thanks Minghao Guo for discussions on a closely related project. ERM and HYW acknowledge support from the National Science Foundation through award NSF-AST2508940. Part of the simulations were performed on DOE OLCF Summit under allocation AST198. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This work used Delta at the National Center for Supercomputing Applications (NCSA) through allocation CIS250856 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program [39], which is supported by U.S. National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

- [1] John Kormendy and Luis C. Ho. Coevolution (Or Not) of Supermassive Black Holes and Host Galaxies. *Ann. Rev. Astron. Astrophys.*, 51:511–653, 2013.
- [2] Roger Blandford, David Meier, and Anthony Readhead. Relativistic Jets from Active Galactic Nuclei. *ARA&A*, 57:467–509, August 2019.
- [3] B. R. McNamara and P. E. J. Nulsen. Heating Hot Atmospheres with Active Galactic Nuclei. *Ann. Rev. Astron. Astrophys.*, 45:117–175, 2007.
- [4] Aretaios Lalakos, Alexander Tchekhovskoy, Omer Bromberg, Ore Gottlieb, Jonatan Jacquemin-Ide, Matthew Liska, and Haocheng Zhang. Jets with a Twist: The Emergence of FR0 Jets in a 3D GRMHD Simulation of Zero-angular-momentum Black Hole Accretion. *Astrophys. J.*, 964(1):79, 2024.
- [5] Aretaios Lalakos, Ore Gottlieb, Nick Kaaz, Koushik Chatterjee, Matthew Liska, Ian M. Christie, Alexander Tchekhovskoy, Irina Zhuravleva, and Elena Nokhrina. Bridging the Bondi and Event Horizon Scales: 3D GRMHD Simulations Reveal X-shaped Radio Galaxy Morphology. *Astrophys. J. Lett.*, 936(1):L5, 2022.
- [6] Hector R. Olivares S., Monika A. Moscibrodzka, and Oliver Porth. General relativistic hydrodynamic simulations of perturbed transonic accretion. *Astron. Astrophys.*, 678:A141, 2023.
- [7] Aretaios Lalakos, Alexander Tchekhovskoy, Elias R. Most, Bart Ripperda, Koushik Chatterjee, and Matthew Liska. Universal Radial Scaling of Large-Scale Black Hole Accretion for Magnetically Arrested And Rocking Accretion Disks. *arXiv e-prints*, page arXiv:2505.23888, May 2025.
- [8] Minghao Guo, James M. Stone, Chang-Goo Kim, and Eliot Quataert. Toward Horizon-scale Accretion onto Supermassive Black Holes in Elliptical Galaxies. *Astrophys. J.*, 946(1):26, March 2023.
- [9] Minghao Guo, James M. Stone, Eliot Quataert, and Chang-Goo Kim. Magnetized Accretion onto and Feedback from Supermassive Black Holes in Elliptical Galaxies. *arXiv e-prints*, page arXiv:2405.11711, May 2024.
- [10] Philip F. Hopkins, Michael Y. Grudic, Kung-Yi Su, Sarah Wellons, Daniel Angles-Alcazar, Ulrich P. Steinwandel, David Guszejnov, Norman Murray, Claude-Andre Faucher-Giguere, Eliot Quataert, and Dusan Keres. FORGE’d in FIRE: Resolving the End of Star Formation and Structure of AGN Accretion Disks from Cosmological Initial Conditions. *The Open Journal of Astrophysics*, 7:18, March 2024.

- [11] Philip F. Hopkins, Jonathan Squire, Kung-Yi Su, Ulrich P. Steinwandel, Kyle Kremer, Yanlong Shi, Michael Y. Grudic, Sarah Wellons, Claude-Andre Faucher-Giguere, Daniel Angles-Alcazar, Norman Murray, and Eliot Quataert. FORGE'd in FIRE II: The Formation of Magnetically-Dominated Quasar Accretion Disks from Cosmological Initial Conditions. *The Open Journal of Astrophysics*, 7:19, March 2024.
- [12] Nicholas Kaaz, Matthew Liska, Alexander Tchekhovskoy, Philip F. Hopkins, and Jonatan Jacquemin-Ide. H-AMR FORGE'd in FIRE. I. Magnetic State Transitions, Jet Launching, and Radiative Emission in Super-Eddington, Highly Magnetized Quasar Disks Formed from Cosmological Initial Conditions. *Astrophys. J.*, 979(2):248, 2025.
- [13] Hyerin Cho, Ben S. Prather, Ramesh Narayan, Priyamvada Natarajan, Kung-Yi Su, Angelo Ricarte, and Koushik Chatterjee. Bridging Scales in Black Hole Accretion and Feedback: Magnetized Bondi Accretion in 3D GRMHD. *Astrophys. J. Lett.*, 959(2):L22, 2023.
- [14] Hyerin Cho, Ben S. Prather, Kung-Yi Su, Ramesh Narayan, and Priyamvada Natarajan. Multi-zone Modeling of Black Hole Accretion and Feedback in 3D GRMHD: Bridging Vast Spatial and Temporal Scales. *Astrophys. J.*, 977(2):200, 2024.
- [15] Hyerin Cho, Ben S. Prather, Ramesh Narayan, Kung-Yi Su, and Priyamvada Natarajan. Bridging Scales in Black Hole Accretion and Feedback: Relativistic Jet linking the Horizon to the Host Galaxy. *arxiv*, 7 2025.
- [16] Minghao Guo, James M. Stone, Eliot Quataert, and Volker Springel. Cyclic Zoom: Multi-scale GRMHD Modeling of Black Hole Accretion and Feedback. *arxiv*, 4 2025.
- [17] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.
- [18] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.
- [19] Vignesh Gopakumar, Stanislas Pamela, Lorenzo Zanisi, Zongyi Li, Ander Gray, Daniel Brennand, Nitesh Bhatia, Gregory Stathopoulos, Matt Kusner, Marc Peter Deisenroth, Anima Anandkumar, The Jorek Team, and Mast Team. Plasma surrogate modelling using Fourier neural operators. *Nuclear Fusion*, 64(5):056025, May 2024.
- [20] Jaideep Pathak, Shashank Subramanian, Peter Harrington, Sanjeev Raja, Ashesh Chattopadhyay, Morteza Mardani, Thorsten Kurth, David Hall, Zongyi Li, Kamyar Azizzadenesheli, et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.
- [21] Gege Wen, Zongyi Li, Qirui Long, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M Benson. Real-time high-resolution co 2 geological storage prediction using nested fourier neural operators. *Energy & Environmental Science*, 16(4):1732–1741, 2023.
- [22] Bahareh Tolooshams, Aditi Chandrashekhar, Rayhan Zirvi, Abbas Mammadov, Jiachen Yao, Chuwei Wang, and Anima Anandkumar. Equireg: Equivariance regularized diffusion for inverse problems. *arXiv preprint arXiv:2505.22973*, 2025.
- [23] Chuwei Wang, Julius Berner, Zongyi Li, Di Zhou, Jiayun Wang, Jane Bae, and Anima Anandkumar. Coarse Graining with Neural Operators for Simulating Chaotic Systems. *arXiv e-prints*, page arXiv:2408.05177, August 2024.
- [24] Jiayun Wang, Yousuf Aborahama, Arya Khokhar, Yang Zhang, Chuwei Wang, Karteekeya Sastry, Julius Berner, Yilin Luo, Boris Bonev, Zongyi Li, et al. Accelerating 3d photoacoustic computed tomography with end-to-end physics-aware neural operators. *arXiv preprint arXiv:2509.09894*, 2025.

- [25] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhardt, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.
- [26] Roberta Duarte, Rodrigo Nemmen, and João Paulo Navarro. Black hole weather forecasting with deep learning: a pilot study. *Monthly Notices of the Royal Astronomical Society*, 512(4):5848–5861, 2022.
- [27] Keith Poletti, Stella SR Offner, and Rachel A Ward. Modeling turbulent and self-gravitating fluids with fourier neural operators. *APL Machine Learning*, 3(2), 2025.
- [28] Semih Kacmaz, EA Huerta, and Roland Haas. Resolving turbulent magnetohydrodynamics: A hybrid operator-diffusion framework. *arXiv preprint arXiv:2507.02106*, 2025.
- [29] Philip F. Hopkins, Andrew Wetzel, Coral Wheeler, Robyn Sanderson, Michael Y. Grudić, Omid Sameie, Michael Boylan-Kolchin, Matthew Orr, Xiangcheng Ma, Claude-André Faucher-Giguère, Dušan Kereš, Eliot Quataert, Kung-Yi Su, Jorge Moreno, Robert Feldmann, James S. Bullock, Sarah R. Loebman, Daniel Anglés-Alcázar, Jonathan Stern, Lina Necib, Caleb R. Choban, and Christopher C. Hayward. FIRE-3: updated stellar evolution models, yields, and microphysics and fitting functions for applications in galaxy simulations. *MNRAS*, 519(2):3154–3181, February 2023.
- [30] Dylan Nelson et al. The IllustrisTNG simulations: public data release. *Comput. Astrophys. Cosmol.*, 6(1):2, 2019.
- [31] Philip F. Hopkins, Andrew Wetzel, Dusan Keres, Claude-Andre Faucher-Giguere, Eliot Quataert, Michael Boylan-Kolchin, Norman Murray, Christopher C. Hayward, and Kareem El-Badry. How To Model Supernovae in Simulations of Star and Galaxy Formation. *Mon. Not. Roy. Astron. Soc.*, 477(2):1578–1603, 2018.
- [32] Iryna S. Butsky, Cameron B. Hummels, Philip F. Hopkins, Thomas R. Quinn, and Jessica K. Werk. Cold Gas Subgrid Model (CGSM): a two-fluid framework for modelling unresolved cold gas in galaxy simulations. *MNRAS*, 535(2):1672–1683, December 2024.
- [33] Keiya Hirashima, Kana Moriwaki, Michiko S. Fujii, Yutaka Hirai, Takayuki R. Saitoh, and Junichiro Makino. 3D-Spatiotemporal forecasting the expansion of supernova shells using deep learning towards high-resolution galaxy simulations. *MNRAS*, 526(3):4054–4066, December 2023.
- [34] Keiya Hirashima, Kana Moriwaki, Michiko S. Fujii, Yutaka Hirai, Takayuki R. Saitoh, Junichiro Makino, Ulrich P. Steinwandel, and Shirley Ho. ASURA-FDPS-ML: Star-by-star Galaxy Simulations Accelerated by Surrogate Modeling for Supernova Feedback. *ApJ*, 987(1):86, July 2025.
- [35] Roberta Duarte, Rodrigo Nemmen, and João Paulo Navarro. Black hole weather forecasting with deep learning: a pilot study. *MNRAS*, 512(4):5848–5861, June 2022.
- [36] Miguel Liu-Schiaffini, Julius Berner, Boris Bonev, Thorsten Kurth, Kamyar Azizzadenesheli, and Anima Anandkumar. Neural operators with localized integral and differential kernels. *arXiv preprint arXiv:2402.16845*, 2024.
- [37] Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning chaotic dynamics in dissipative systems. *Advances in Neural Information Processing Systems*, 35:16768–16781, 2022.
- [38] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.
- [39] Timothy J. Boerner, Stephen Deems, Thomas R. Furlani, Shelley L. Knuth, and John Towns. Access: Advancing innovation: Nsf’s advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing 2023: Computing for the Common Good*, PEARC ’23, page 173–176, New York, NY, USA, 2023. Association for Computing Machinery.

- [40] James M. Stone, Kengo Tomida, Christopher J. White, and Kyle G. Felker. The Athena++ Adaptive Mesh Refinement Framework: Design and Magnetohydrodynamic Solvers. *Astrophys. J.s*, 249(1):4, July 2020.
- [41] Christian Trott, Luc Berger-Vergiat, David Poliakoff, Sivasankaran Rajamanickam, Damien Lebrun-Grandie, Jonathan Madsen, Nader Al Awar, Milos Gligoric, Galen Shipman, and Geoff Womeldorff. The Kokkos EcoSystem: Comprehensive Performance Portability for High Performance Computing. *Computing in Science and Engineering*, 23(5):10–18, September 2021.
- [42] James M. Stone, Patrick D. Mullen, Drummond Fielding, Philipp Grete, Minghao Guo, Philipp Kempski, Elias R. Most, Christopher J. White, and George N. Wong. AthenaK: A Performance-Portable Version of the Athena++ AMR Framework. *arXiv e-prints*, page arXiv:2409.16053, September 2024.
- [43] P. Colella and Paul R. Woodward. The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations. *Journal of Computational Physics*, 54:174–201, September 1984.
- [44] Takahiro Miyoshi and Kanya Kusano. A multi-state HLL approximate Riemann solver for ideal magnetohydrodynamics. *Journal of Computational Physics*, 208(1):315–344, September 2005.
- [45] Thomas A. Gardiner and James M. Stone. An unsplit Godunov method for ideal MHD via constrained transport in three dimensions. *Journal of Computational Physics*, 227(8):4123–4141, April 2008.
- [46] M. Nicole Lemaster and James M. Stone. Dissipation and Heating in Supersonic Hydrodynamic and MHD Turbulence. *Astrophys. J.*, 691(2):1092–1108, February 2009.
- [47] L. G. Fishbone and V. Moncrief. Relativistic fluid disks in orbit around Kerr black holes. *Astrophys. J.*, 207:962–976, August 1976.
- [48] Thomas A. Gardiner and James M. Stone. An unsplit Godunov method for ideal MHD via constrained transport. *Journal of Computational Physics*, 205(2):509–539, May 2005.
- [49] James M. Stone, Thomas A. Gardiner, Peter Teuben, John F. Hawley, and Jacob B. Simon. Athena: A New Code for Astrophysical MHD. *Astrophys. J. Suppl.*, 178:137, 2008.

Appendix

The structure of the appendix is as follows.

- Appendix A describes dynamical systems of MHD and GRMHD, as well as their numerical simulations.
- Appendix B summarizes the observables we considered in this work as a validation metric.
- Appendix C presents the detailed methods we adopt to train a neural operator under this large-scale astrophysics problem scenario, as well as the implementation details for model training.
- Appendix D contains the experimental results for ablation studies and comparisons with baseline methods.
- Appendix E provides the detailed method for zoom-out part of our feeding-back paradigm and the results.

A Method-Simulation

We use a performance portable version of Athena++ [40] based on Kokkos library [41]-AthenaK [42]. We use piecewise parabolic reconstruction [43], an HLLD Riemann solver [44], a constraint transport algorithm [45] for the divergence-free magnetic field evolution, and first-order flux correction [46].

A.1 Magneto-hydrodynamics Simulations: Magnetized Bondi Accretion

The setup we used in this work is magnetized Bondi accretion onto a central supermassive black hole [9]. Except for the case that the current run is without cooling and heating terms, the setup is the same as the fiducial run in [9]. In this problem, we solve the Newtonian magnetohydrodynamics equations in conservative form:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = s_\rho, \quad (1)$$

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} + P \mathbf{I} - \mathbf{B} \mathbf{B}) = \mathbf{s}_p - \rho \nabla \Phi, \quad (2)$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + P) \mathbf{v} - (\mathbf{B} \cdot \mathbf{v}) \mathbf{B}] = s_E - \rho \mathbf{v} \cdot \nabla \Phi, \quad (3)$$

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla \times [\mathbf{v} \times \mathbf{B}] = 0, \quad (4)$$

where $\rho : \mathbb{R}^3 \rightarrow \mathbb{R}$ is the gas density, $\mathbf{v} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is the velocity, $P : \mathbb{R}^3 \rightarrow \mathbb{R}$ is the pressure, $E = P/(\gamma - 1) + \rho|v|^2/2$ is the total energy density, and $-\nabla \Phi$ is the gravitational acceleration due to the central accretor. The flow evolved following ideal gas law with adiabatic index $\gamma = 5/3$. The source terms s_ρ , \mathbf{s}_p , and s_E represent removal of mass, momentum, and energy by the central black hole, respectively (e.g., accretion or feedback). These source terms are concentrated at the very center of the domain, covering 1 grid for the training set resolution.

In code unit, $GM = r_0 = \rho_0 = 1$. The output quantities include $(\rho, P, \mathbf{v} = (v_x, v_y, v_z), \mathbf{B} = (B_x, B_y, B_z))$.

A.2 GRMHD Simulations

The conservative Valencia formulations for solving GRMHD equations with induction equation

$$\partial_t \mathbf{U} + \partial_i \mathbf{F} = \mathbf{S} \quad (5)$$

$$\partial_t (\sqrt{-g} B^i) + \partial_j (\sqrt{-g} (b^i u^j - b^j u^i)) = 0 \quad (6)$$

where the conserved variables, fluxes, and source terms associated with the connection are

$$\mathbf{U} = \sqrt{-g} \begin{bmatrix} \rho u^t \\ T_i^t \\ T_t^t + \rho u^t \end{bmatrix}, \mathbf{F} = \sqrt{-g} \begin{bmatrix} \rho u^j \\ T_i^j \\ T_t^j + \rho u^j \end{bmatrix}, \mathbf{S} = \sqrt{-g} \begin{bmatrix} 0 \\ \frac{1}{2} (\partial_i g_{\alpha\beta}) T^{\alpha\beta} \\ 0 \end{bmatrix} \quad (7)$$

respectively, where the metric is \mathbf{g} ($g = \det \mathbf{g}$). The rest-mass density is ρ , the coordinate frame 4-velocity is u^μ and the stress-energy tensor is

$$T^\mu_\nu = w u^\mu u_\nu - b^\mu b_\nu + (p_g + p_m) \delta^\mu_\nu \quad (8)$$

where the magnetic field is b^μ , the magnetic pressure is $p_m = b_\mu b^\mu / 2$, and the total enthalpy is w . The non-monopole constraint is preserved when evolving the 3-components B^i of the magnetic field:

$$\partial_j (\sqrt{-g} B^j) = 0 \quad (9)$$

where

$$b^t = u_i B^i, b^i = \frac{1}{u^t} (B^i + b^t u^i). \quad (10)$$

We solve the GRMHD equations in a Cartesian Kerr-Schild coordinate. We adopt piecewise parabolic spatial reconstruction, HLL-E Riemann solver, RK2 time integrator, and first-order flux correction [46].

The setup we used in this work is the accretion of a Fishbone-Moncrief torus [47] onto a spinning black hole (with spin $a = 0.9$). We choose the default magnetically arrested state setup in AthenaK public repo.

In all runs, we adopt $G = M = \rho_0 = 1$. The output quantities include $(\rho, P, \mathbf{v} = (v_x, v_y, v_z), \mathbf{B} = (B_x, B_y, B_z))$.

B Observables

Here we use the spherically averaged mass density ρ , temperature $T = P/\rho$, and mass accretion rate \dot{M} as observables (shown in panel (2) of Fig. 2). The mass accretion rate is defined by

$$\dot{M} \equiv - \int_S \rho u^r \sqrt{-g} d\Omega, \quad (11)$$

In the Newtonian limit, $\sqrt{-g} = 1$.

C Implementation Details

C.1 System and Environment

Training is executed on Linux (x86_64) with CPython 3.11 and CUDA. A single NVIDIA GeForce RTX 4090 GPU is used; data loading employs pinned memory and worker parallelism. Mixed precision is disabled by default; gradient accumulation is enabled.

C.2 Data Representation

Each snapshot contains eight 3D fields on a 64^3 grid, stacked as a tensor of shape (C, D, H, W) with $C=8$: magnetic-field components B_x, B_y, B_z (variables `bcc1`, `bcc2`, `bcc3`), mass density ρ (`dens`), internal energy e (`eint`), and velocity components v_x, v_y, v_z (`velx`, `vely`, `velz`).

C.3 Normalization: Channel-wise Transforms, Robust Scaling, and Soft Clipping

Transforms. Before standardization, each channel x_c is mapped by a type-specific T_c :

$$\hat{x}_c = \begin{cases} \log_{10}(x_c + \varepsilon_c), & \text{pos (dens, eint)}, \\ \operatorname{sgn}(x_c) \log_{10}\left(1 + \frac{|x_c|}{\varepsilon_c}\right), & \text{signed (bcc1:3)}, \\ x_c, & \text{linear (velx:velz)}. \end{cases}$$

Per-channel scales ε_c are estimated from a representative training subset:

$$\varepsilon_c = \begin{cases} 10^{\lfloor \log_{10}(\min x_c) \rfloor - 2}, & \text{pos}, \\ 10^{\lfloor \log_{10}(\max |x_c|) \rfloor - 2}, & \text{signed}, \\ 0, & \text{linear}. \end{cases}$$

Robust z-scoring and soft clipping. Let $m_c = \text{median}(\hat{x}_c)$ and $s_c = 1.4826 \text{ median}(|\hat{x}_c - m_c|)$ (clamped to 10^{-6}). Define

$$z_c = \frac{\hat{x}_c - m_c}{s_c}, \quad \tilde{z}_c = \gamma \tanh\left(\frac{z_c}{\gamma}\right),$$

with soft-clip range $\gamma=6$ (default). Decoding inverts the pipeline via

$$z_c = \gamma \operatorname{atanh}\left(\frac{\tilde{z}_c}{\gamma}\right), \quad \hat{x}_c = s_c z_c + m_c, \quad x_c = T_c^{-1}(\hat{x}_c),$$

where

$$T_c^{-1}(u) = \begin{cases} 10^u - \varepsilon_c, & \text{pos,} \\ \operatorname{sgn}(u) \varepsilon_c (10^{|u|} - 1), & \text{signed,} \\ u, & \text{linear.} \end{cases}$$

Because of soft clipping, the inverse is approximate near $|\tilde{z}_c| \approx \gamma$; we clamp inputs to atanh to 0.99γ .

C.4 Radial Shell Positional Channels

To provide absolute radial context, we augment inputs with m concentric *shell* indicators on the 64^3 grid. Let the grid center be c and define $r(i, j, k) = \|(i, j, k) - c\|_2$. Choose logarithmically spaced boundaries $0 = b_0 < b_1 < \dots < b_m = r_{\max}$ on $[10^{-1}, r_{\max}]$ (defaults: $m=8, r_{\max}=10$). Each voxel receives an index $s(i, j, k)$ with $b_s \leq r < b_{s+1}$ (voxels with $r \geq r_{\max}$ use $s=m-1$). We one-hot encode into $E \in \{0, 1\}^{m \times 64 \times 64 \times 64}$ and *concatenate* E to the eight physical channels. Shell channels are binary and are never normalized; normalization applies only to the physical channels.

C.5 Single-Stage Radial Scaling Baseline (Residualization)

We add a fixed radial reference to the `dens` and `eint` outputs and train the model to predict the residual. With a precomputed radius grid r on 64^3 , the reference is

$$B(r) = k r,$$

where the slope k is estimated offline by least-squares regression of $\log_{10}(U)$ versus r using $U = \text{dens} + \text{eint}$. During inference, the wrapped predictor returns $\hat{y} = f_\theta(x) + B(r)$ on the `dens/eint` channels.

Residual envelope around the radial baseline. With the reference $B(r)$ applied to `dens/eint`, the network predicts residuals $\eta_c = \hat{y}_c - B(r)$ for $c \in \{\rho, e\}$. Since $B(r)$ captures the global radial decay, η_c should be near-zero and encode only local, non-radial structure. To prevent the model from re-learning the radial trend or drifting by large offsets, we impose a symmetric hinge “envelope” of half-width Δ_c (in normalized units) around $B(r)$:

$$\mathcal{L}_{\text{env}} = w_\rho \mathbb{E}[(|\hat{y}_\rho - B(r)| - \Delta_\rho)_+^2] + w_e \mathbb{E}[(|\hat{y}_e - B(r)| - \Delta_e)_+^2],$$

where $(u)_+ = \max(u, 0)$. The envelope is radius-aware (centered at $B(r)$), gives zero penalty to residuals within $\pm\Delta_c$ so fine structure can be learned, and grows quadratically outside to discourage large bias. We choose Δ_c in normalized coordinates (fixed hyperparameters; e.g., $\Delta_\rho = \Delta_e = 1.5$ unless noted), roughly matching the robust spread of residuals on the training set.

C.6 Data-Driven Physical Bounds and Mapping to Normalized Space

We compute robust bounds for density and internal energy on the *training partition* using quantiles $q_{\text{low}}, q_{\text{high}}$ (defaults: 0.001, 0.999). With a symmetric multiplicative margin δ (default: 5%) in physical units,

$$\ell_\rho = (1-\delta) Q_{q_{\text{low}}}(\rho), \quad u_\rho = (1+\delta) Q_{q_{\text{high}}}(\rho), \quad \ell_e = (1-\delta) Q_{q_{\text{low}}}(e), \quad u_e = (1+\delta) Q_{q_{\text{high}}}(e).$$

Bounds are mapped to the normalized domain via each channel’s transform and robust scaling:

$$b^{\text{norm}} = \frac{T_c(b) - m_c}{s_c}.$$

During training we penalize violations wrt. these normalized thresholds; at evaluation we clamp `dens/eint` in normalized space to $[b^{\text{norm}}, u^{\text{norm}}]$ prior to decoding.

C.7 Objective Function

Let \hat{y} and y denote predictions and targets in the normalized domain, with channels grouped as magnetic field $\mathbf{B}:=\text{bcc1}:3$, velocity $\mathbf{v}:=\text{velx}:\text{velz}$, density ρ , and internal energy e .

Component-weighted L^2 .

$$\mathcal{L}_{\mathbf{B}} = \lambda_B \|\hat{y}_{\mathbf{B}} - y_{\mathbf{B}}\|_2^2, \quad \mathcal{L}_{\mathbf{v}} = \lambda_{\text{vel}} \|\hat{y}_{\mathbf{v}} - y_{\mathbf{v}}\|_2^2, \quad \mathcal{L}_{\rho} = \|\hat{y}_{\rho} - y_{\rho}\|_2^2, \quad \mathcal{L}_e = \|\hat{y}_e - y_e\|_2^2.$$

H^1 match.

$$\mathcal{L}_{H^1} = \lambda_{H^1} \sum_c \|\nabla \hat{y}_c - \nabla y_c\|_2^2.$$

Velocity ROI emphasis. Let $v = \text{decode}(y_{\mathbf{v}})$ and $s = \|v\|_2$ (voxelwise speed). For each batch, define a threshold τ as the $(100 - \rho)\%$ quantile of s (default $\rho=20$). With mask $M = \mathbf{1}[s \geq \tau]$, scale $\kappa=8$, and linear epoch ramp $r(e) = \min(1, e/375)$,

$$\mathcal{L}_{\text{vel,ROI}} = r(e) \kappa \mathbb{E} \left[\frac{\|(\hat{y}_{\mathbf{v}} - y_{\mathbf{v}}) \odot M\|_2}{\|y_{\mathbf{v}} \odot M\|_2 + \epsilon} \right].$$

Dissipative regularizer. To discourage spurious amplification, we penalize *only* positive growth of the global L^2 norm in normalized space. Let $\|\cdot\|$ denote the L^2 norm over all channels/voxels. From the training set compute $R_{\max} = \max \|x\|$ and set $R_{\text{in}} = 1.05 R_{\max}$ and $R_{\text{out}} = 1.5 R_{\text{in}}$. Define a contracted target $y_{\text{tgt}} = (R_{\text{in}}/R_{\text{out}}) x$. For an input x with raw prediction y_{pred} , use a smooth gate $\rho(x) = \sigma(\beta [R_{\text{in}} - \|x\|])$ (with $\beta=10$) and blend $\tilde{y} = \rho(x) y_{\text{pred}} + (1 - \rho(x)) y_{\text{tgt}}$. The loss is

$$\mathcal{L}_{\text{diss}} = \alpha \mathbb{E} \left[(\|\tilde{y}\| - \|x\|)_+ \right], \quad \alpha = 5 \times 10^{-4}, \quad (u)_+ = \max(u, 0).$$

Intuitively, states with $\|x\| \ll R_{\text{in}}$ are unaffected ($\rho \approx 1$), while large-amplitude states are softly nudged towards a uniformly contracted copy.

Constraint penalties (normalized domain).

$$\mathcal{L}_{\text{constr}} = \lambda_e^{\text{lo}} \mathbb{E}[\max\{0, \ell_e^{\text{norm}} - \hat{y}_e\}^2] + \lambda_{\rho}^{\text{lo}} \mathbb{E}[\max\{0, \ell_{\rho}^{\text{norm}} - \hat{y}_{\rho}\}^2] \quad (12)$$

$$+ \lambda_{\rho}^{\text{hi}} \mathbb{E}[\max\{0, \hat{y}_{\rho} - u_{\rho}^{\text{norm}}\}^2] + \lambda_e^{\text{hi}} \mathbb{E}[\max\{0, \hat{y}_e - u_e^{\text{norm}}\}^2]. \quad (13)$$

Total loss.

$$\mathcal{L} = \underbrace{\mathcal{L}_{\mathbf{B}} + \mathcal{L}_{\mathbf{v}} + \mathcal{L}_{\rho} + \mathcal{L}_e}_{\text{component-weighted } L^2} + \lambda_{H^1} \mathcal{L}_{H^1} + \mathcal{L}_{\text{vel,ROI}} + \mathcal{L}_{\text{dissip}} + \mathcal{L}_{\text{env}} + \mathcal{L}_{\text{constr}}.$$

C.8 Backbone and Wrapper

Backbone. We employ a 3D *Local Neural Operator* with equidistant discrete–continuous convolutions (DISCO) [36] specialized to volumetric inputs. When shell labels are enabled, inputs have $8+m$ channels, where m comes from shell embedding; outputs remain the eight physical channels.

Wrapper. A *Scaling Baseline Wrapper* adds $B(r) = k r$ to the `dens/eint` channels on top of the base model’s residual prediction.

C.9 Optimization and Schedule

Defaults.

- **Epochs:** 1200; **Batch size:** 4; **Gradient accumulation:** 4 (effective batch 16).
- **Optimizer:** Adam, learning rate 10^{-3} , weight decay 10^{-4} .
- **LR schedule:** Linear warmup for 75 epochs from $0.1 \times \text{lr}$, then cosine annealing to 10^{-6} .
- **Gradient clip:** global norm 1.0; **Early stopping:** patience 100 epochs.

Table 1: Default training configuration (summary).

Parameter	Value
Epochs	1200
Batch size / Grad. accumulation	4 / 4 (effective 16)
Adam learning rate / Weight decay	$1.0 \times 10^{-3} / 1.0 \times 10^{-4}$
LR schedule	Warmup 75 → Cosine to 10^{-6}
Gradient clip (global norm)	1.0
Early stopping patience	100
Normalization clip range γ	6.0
Shell labels	enabled, $m=8, r_{\max}=10$
Scaling baseline	single-stage, $k = -1.78 \times 10^{-2}$
Quantile bounds ($q_{\text{low}}, q_{\text{high}}$)	(0.001, 0.999), margin 5%
Velocity ROI	top 20%, scale 8, ramp 375
L^2 weights	$\lambda_B=1.2, \lambda_{\text{vel}}=1.0$
H^1	$\lambda_{H^1}=0.05$
Dissipative regularizer	$\alpha_{\text{dissip}}=5 \times 10^{-4}$
Residual envelope	$\Delta_\rho=\Delta_e=1.5, w_\rho=w_e=0.05$
Constraint penalties	$\lambda_e^{\text{lo}}=\lambda_\rho^{\text{lo}}=0.05$; upper off

- **Normalization clip range:** $\gamma=6.0$.
- **Shell labels:** enabled; $m=8, r_{\max}=10$.
- **Scaling baseline:** enabled; slope $k = -1.78 \times 10^{-2}$ (single-stage).
- **Quantile bounds:** $q_{\text{low}}=0.001, q_{\text{high}}=0.999$, margin $\delta=5\%$.
- **Velocity ROI:** top 20%, scale 8.0, ramp 375 epochs.
- **Loss weights:** $\lambda_B=1.2, \lambda_{\text{vel}}=1.0, \lambda_{H^1}=0.05, \alpha_{\text{dissip}}=5 \times 10^{-4}$, residual envelope $w_\rho=w_e=0.05$ with $\Delta_\rho=\Delta_e=1.5$, constraint floors $\lambda_e^{\text{lo}}=\lambda_\rho^{\text{lo}}=0.05$, upper-bound weights off by default.

C.10 Metrics, Logging, and Evaluation

Validation metric. Unless noted otherwise, we report volumetric relative L_2 error (%) on the validation set in the normalized domain (after transforms and soft clipping), either per-channel or average across channels.

Diagnostics and logging. Per-batch diagnostics include tensor shapes and dtype, global min/max/mean/std, NaN/Inf counts, gradient norms at regular intervals, and rates of constraint violations. Loss components and scheduler state are logged throughout training.

Evaluation-time clamping and decoding. For `dens/eint`, predictions are clamped in normalized space to $[\ell^{\text{norm}}, u^{\text{norm}}]$ before decoding to physical units. We then compute spectra and multi-step rollout diagnostics as applicable.

C.11 Practical Notes

- **Soft-clip inversion is approximate.** Values near the clip boundary do not round-trip exactly; inputs to `atanh` are clamped to 0.99 γ .
- **Bounds depend on the training partition.** Quantile bounds are computed on training data to prevent leakage; changing the split or δ will move thresholds.
- **Normalized-space semantics.** Envelope widths and penalties are specified in z-scored units; their physical meaning depends on m_c and s_c .
- **Channel dimensionality with shells.** Enabling shell labels increases input channels from 8 to $8+m$; outputs remain the eight physical channels.

D Ablation Study

Protocol (held constant). Unless stated otherwise, all settings match Sec. C: same data split, canonical 64^3 grid, channel-wise transforms with robust z-scoring and soft clipping, optimization/schedule (epochs, batch, LR, warmup+cosine, gradient clipping, early stopping), and reporting in the normalized domain. Decoding and evaluation-time clamping for `dens/eint` follow the same procedure.

D.1 Ablation configurations

We ablate one component at a time, keeping everything else as in the default setup (“*Ours*”). Let $\mathbf{B}:=\text{bcc1:3}$, $\mathbf{v}:=\text{velx:velz}$, $\rho:=\text{dens}$, $e:=\text{eint}$, and let $B(r)=k r$ denote the single-stage radial baseline applied on the ρ/e channels.

- **Plain L2.** Remove all auxiliary loss terms and weights; minimize an *unweighted* per-voxel MSE across the eight channels in the normalized domain:

$$\mathcal{L}_{\text{plain}} = \sum_{c \in \{\mathbf{B}, \mathbf{v}, \rho, e\}} \|\hat{y}_c - y_c\|_2^2.$$

Concretely, set $\lambda_B=\lambda_{\text{vel}}=1$, disable \mathcal{L}_{H^1} , ROI velocity, dissipative regularizer.

- **No radial/constraint.** Keep component-weighted L^2 , H^1 , ROI velocity, and the dissipative regularizer as in Sec. C, but *disable any radius-specific priors/constraints*: (i) remove the scaling baseline $B(r)$ (train/predict in the raw normalized space), (ii) remove residual envelope penalties on ρ/e , (iii) remove constraint penalties (and do not clamp at evaluation). Positional channels remain as in the default.
- **No PE/Radial Shell.** Remove all positional encodings: do not concatenate shell one-hots and do not use any alternative positional features. Also set the inner-region loss nudge to zero so there is no implicit radial reweighting; keep all other terms (component-weighted L^2 , H^1 , ROI velocity, dissipative regularizer, constraints, and the single-stage $B(r)$).
- **PE (Fourier positional encoding).** Replace shell one-hots with Fourier features of *normalized Cartesian coordinates* $(\xi, \eta, \zeta) \in [0, 1]^3$: for $\ell = 0, \dots, L-1$ with $L=4$,

$$\phi_\ell(u) = [\sin(2\pi 2^\ell u), \cos(2\pi 2^\ell u)], \quad u \in \{\xi, \eta, \zeta\}.$$

Concatenate the $3 \times 2L$ channels to the physical inputs; do not use shell one-hots. All losses, constraints, and $B(r)$ remain enabled.

- **CNN backbone (3D CNN instead of neural operator).** We replace the Local Neural Operator with a lightweight 3D U-Net-style encoder-decoder matched for parameter count (base width 32; four resolution levels with channel widths [32, 64, 128, 256]). Each stage uses residual $3 \times 3 \times 3$ conv blocks with GroupNorm and SiLU activations; downsampling is $\times 2$ via max-pooling and upsampling via transposed convolutions. Inputs are the eight physics channels + n_{shell} radial shell one-hots; outputs are the physics channels. All training losses, constraints, and the radial baseline $B(r)$ are kept identical to the operator models.

D.2 What each ablation isolates

- *Plain L2* probes the net contribution of all auxiliary inductive biases beyond plain fidelity.
- *No radial/constraint* isolates the effect of the scaling prior and data-driven bounds from the rest of the loss design.
- *No PE/Radial Shell* measures the value of absolute positional information.
- *PE* compares Fourier PE against our shell-based positional encoding under otherwise identical training.
- *CNN backbone* tests whether the gains are architectural (operator layer) or purely due to losses/normalization.

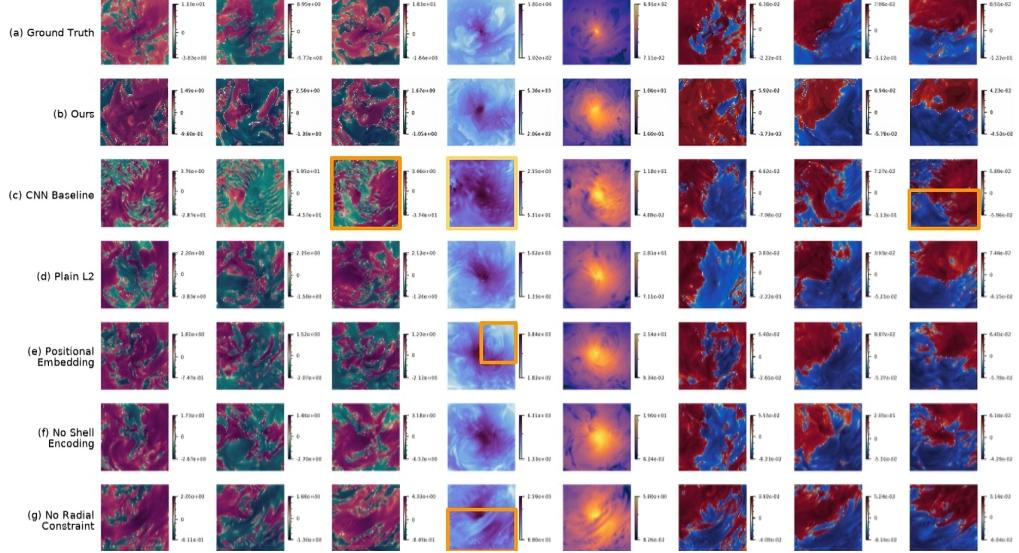


Figure 4: **Ablation study on a central slice at step $t=50$.** Columns (left→right): B_x , B_y , B_z , mass density ρ , internal energy e , and velocity components (v_x, v_y, v_z). Values are in the model’s normalized domain (see Methods). Rows: (a) *Ground truth*; (b) *Ours* (LocalNO + shell positional encoding + radial baseline/constraints + ROI and H^1 terms); (c) *CNN baseline* (pure convolutional surrogate); (d) *Plain L^2* (no component weights/ROI/ H^1); (e) *Positional Embedding only* (Fourier features, no shells); (f) *No Shell Encoding*; (g) *No Radial Constraint/Baseline*. Colored boxes mark typical failure modes observed in ablations: **yellow**: over-smoothing/texture blur; **orange**: spurious mass/energy (unphysical deficits or excess). The full model (b) most closely matches the ground truth, preserving small-scale magnetic structure and inner-torus morphology while avoiding excessive smoothing and mass artifacts.

Table 2: This table reports the avg. relative L_2 error (%) across all channels, along with the per-channel relative L_2 error (%) based on the validation set.

Configuration	Avg. Error (%)	B_x (%)	B_y (%)	B_z (%)	ρ (%)	e (%)	v_x (%)	v_y (%)	v_z (%)
Ours	14.02	16.71	17.01	14.73	10.98	11.25	13.47	14.04	13.94
PE (Fourier)	13.87	16.77	17.06	14.87	9.85	10.42	13.73	14.08	14.15
No PE/Radial Shell	13.93	16.84	17.15	14.89	10.03	10.42	13.87	14.18	14.06
No radial/constraint	14.17	16.84	17.26	14.87	10.87	11.33	13.82	14.29	14.07
Plain L2	13.69	16.76	16.98	14.67	9.55	9.89	13.45	14.06	14.16
CNN backbone	19.09	23.85	23.91	21.44	15.21	14.87	17.46	17.80	18.17

D.3 Rollout for 50 ΔT in MHD (fig. 4)

D.4 Rollout for 100 ΔT in MHD (fig. 5)

D.5 L^2 Loss while preserving dynamics near the singularity

When omitting certain components in our ablation study, we notice a lower L2 validation error for density and internal energy, yet observe poor adherence to the ground truth for all fields at or directly around the black hole. The gravitational effect of the singularity creates a localized region of increased complexity, marginal in size when compared to the entire domain, causing it to be neglected by the Neural Operator in the objective to minimize overall loss. In such a case, we risk losing dynamics near the inner torus region and physics critical to the entire system, emphasizing the need for radial awareness and weightage.

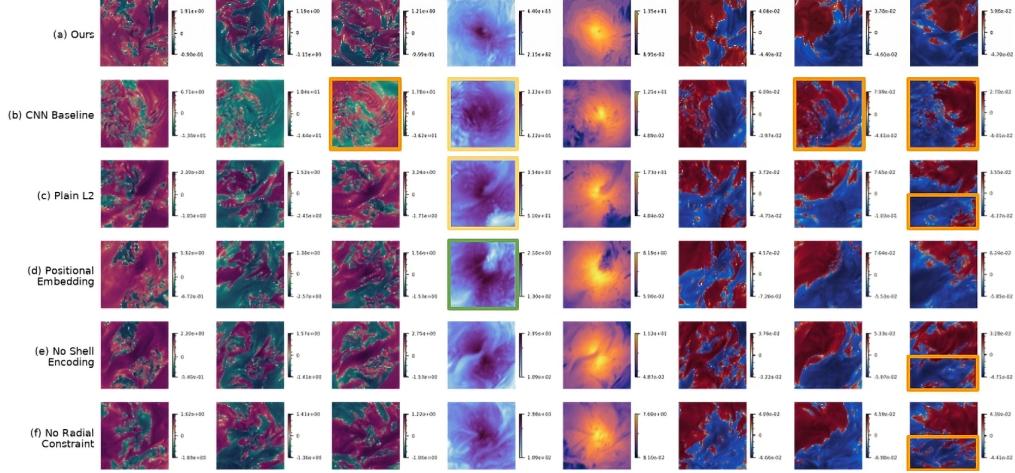


Figure 5: Ablation study on a central slice at step $t=100$. Columns (left→right): B_x , B_y , B_z , mass density ρ , internal energy e , and velocity components (v_x, v_y, v_z). Values are in the model’s normalized domain (see Methods). Rows: (a) *Ours* (LocalNO + shell positional encoding + radial baseline/constraints + ROI and H^1 terms); (b) *CNN baseline* (pure convolutional surrogate); (c) *Plain L^2* (no component weights/ROI/ H^1); (d) *Positional Embedding only* (Fourier features, no shells); (e) *No Shell Encoding*; (f) *No Radial Constraint/Baseline*. Colored boxes mark typical failure modes observed in ablations: **yellow**: over-smoothing/texture blur; **green**: anisotropic diffusion of the torus; **orange**: spurious mass/energy (unphysical deficits or excess). The full model (a) most closely maintains proper structure, preserving small-scale magnetic structure and inner-torus morphology while avoiding excessive smoothing and mass artifacts.

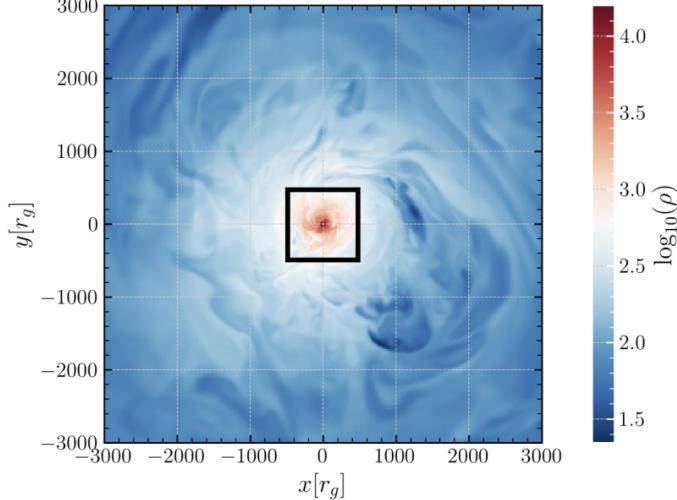


Figure 6: Two-dimensional slice of mass density after coupling the neural operator back to the code. The subgrid region (roll-out of the neural operator) is denoted by the black square.

D.6 Reporting details

For each row in Table 2, train with the same optimizer and schedule, and evaluate on the same validation split. Unless explicitly disabled by the ablation, keep: component-weighted L^2 , H^1 loss, ROI velocity emphasis, dissipative regularizer, data-driven bounds with evaluation-time clamping, robust normalization, soft clipping, and the single-stage baseline $B(r)=kr$.

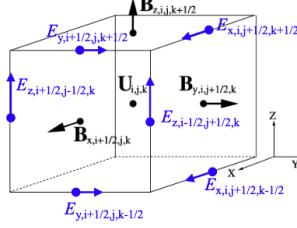


Figure 7: Positions of hydrodynamic variables, magnetic fields, and electric fields. The hydrodynamic variables are at the body-center of each cell, while the electric fields and magnetic fields are at the face-centers and edge-centers.

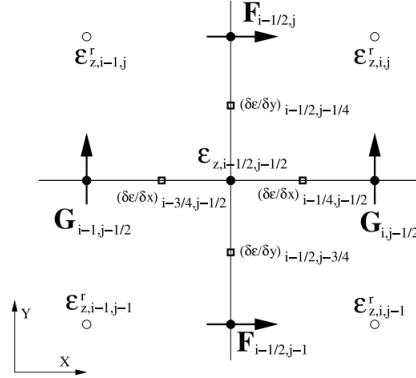


Figure 8: The schematic diagram of the x-y slice, in which shows the cell-centered reference state of EMF E_z^r , cell-cornered EMF E_z used for recovering the area-averaged magnetic field, and the fluxes of conserved variables in the x and y directions.

E Coupling NO to Simulation

We briefly describe how we couple the neural operator to the direct simulation at coarser level in this section. First, we directly output the rollout from neural operator as hdf5 files and each time read in two files to interpolate between two timesteps linearly. The hydrodynamic variables (ρ, P, v_x, v_y, v_z) are used to overwrite the corresponding values at the inner boundary directly. The treatment of coupling the magnetic field to the inner boundary is discussed in detail below.

E.1 Review: Divergence-Free Magnetic Field and Constraint Transport

The magnetic fields are evolved via Stoke's law:

$$\frac{\partial}{\partial t} \int_S \mathbf{B} \cdot d\mathbf{S} = - \int_L \mathbf{E} \cdot d\mathbf{l}, \quad (14)$$

in the differential form :

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \times \mathbf{E} = 0, \quad (15)$$

where the electric field (electromotive force [EMF]) $\mathbf{E} = -\mathbf{v} \times \mathbf{B}$ in ideal MHD.

Constraint transport (CT) algorithm update the area-averaged magnetic fields using the line-averaged EMF at cell corners. First, we can separately define the face-centered, area-averaged magnetic field (B_x) $_{i+1/2,j,k}$ and the edge-centered, line-averaged EMF (E_x) $_{i,j+1/2,k-1/2}$ as follow (taking x-component as an example):

$$(B_x)_{i+1/2,j,k} = \frac{1}{\Delta y \Delta z} \int_S B_x(y, z) dy dz, \quad (16)$$

$$(E_x)_{i,j+1/2,k-1/2} = \frac{1}{\Delta x \Delta t} \int E_x(x) dx dt, \quad (17)$$

The magnetic field at the $n + 1$ timestep is updated using the electro-magnetic forces accordingly

$$B_{x,i+1/2,j,k}^{n+1} = B_{x,i+1/2,j,k}^n - \frac{\Delta t}{\Delta y} \left(E_{z,i-1/2,j+1/2,k}^{n+1/2} - E_{z,i-1/2,j-1/2,k}^{n+1/2} \right) \quad (18)$$

$$+ \frac{\Delta t}{\Delta z} \left(E_{y,i-1/2,j,k+1/2}^{n+1/2} - E_{z,i-1/2,j,k-1/2}^{n+1/2} \right) \quad (19)$$

Then the divergence of magnetic field $\nabla \cdot \mathbf{B}$ is preserved to machine accuracy

$$\nabla \cdot \mathbf{B} = \frac{B_{x,i+1/2,j,k} - B_{x,i-1/2,j,k}}{\Delta x} + \frac{B_{y,i,j+1/2,k} - B_{x,i,j-1/2,k}}{\Delta y} + \frac{B_{z,i,j,k+1/2} - B_{z,i,j,k-1/2}}{\Delta z} \quad (20)$$

E.2 Coupling NO with Constraint Transport

In Athena++/K, it is the Riemann solver that returns area-averaged electric fields at cell faces, while the CT requires the line-averaged EMF at cell corners to update the area-averaged magnetic fields, here this is achieved by [48, 45, 49]:

$$E_{z,i-1/2,j-1/2} = \frac{1}{4} (E_{z,i-1/2,j} + E_{z,i-1/2,j+1} + E_{z,i,j-1/2} + E_{z,i+1,j-1/2}) \quad (21)$$

$$+ \frac{\delta y}{8} \left[\left(\frac{\partial E_z}{\partial y} \right)_{i-1/2,j-1/4} - \left(\frac{\partial E_z}{\partial y} \right)_{i-1/2,j-3/4} \right] \quad (22)$$

$$+ \frac{\delta x}{8} \left[\left(\frac{\partial E_z}{\partial x} \right)_{i-1/4,j-1/2} - \left(\frac{\partial E_z}{\partial x} \right)_{i-3/4,j-1/2} \right], \quad (23)$$

where the derivative of EMF on each cell face is computed from the upwinded contact mode from the Riemann solver, i.e., suppressing the j subscript,

$$\left(\frac{\partial E_z}{\partial y} \right)_{i-1/2} = \begin{cases} (\partial E_z / \partial y)_{i-1}, & \text{for } v_{x,i-1/2} > 0 \\ (\partial E_z / \partial y)_i, & \text{for } v_{x,i-1/2} < 0 \\ \frac{1}{2} \left[\left(\frac{\partial E_z}{\partial y} \right)_{i-1} + \left(\frac{\partial E_z}{\partial y} \right)_i \right], & \text{otherwise,} \end{cases} \quad (24)$$

The cell-centered derivatives of EMF are calculated from the face-centered EMF (Godunov flux) and cell-centered reference EMF E_z^r (see Fig. 8):

$$\left(\frac{\partial E_z}{\partial y} \right)_{i,j-1/4} = 2 \left(\frac{E_{z,i,j}^r - E_{z,i,j-1/2}}{\delta y} \right), \quad (25)$$

For more details on how to compute the reference EMF in a certain timestep, see [49]. Since the NO is trained to predict the cell-centered volume-averaged magnetic field, from Eq. 23 and 25, we replace the face-centered EMF from Riemann solver with that from NO prediction.