Emulating Radiative Transfer in Astrophysical Environments

Rune Rost[1] **Lorenzo Branca**[1] **Tobias Buck**[1]
[1]Interdisciplinary Center for Scientific Computing, Heidelberg University, Germany
rune.rost@stud.uni-heidelberg.de
lorenzo.branca@iwr.uni-heidelberg.de
tobias.buck@iwr.uni-heidelberg.de

Abstract

Radiative transfer is a fundamental process in astrophysics, essential for both interpreting observations and modeling thermal and dynamical feedback in simulations via ionizing radiation and photon pressure. However, numerically solving the underlying radiative transfer equation is computationally intensive due to the complex interaction of light with matter and the disparity between the speed of light and the typical gas velocities in astrophysical environments, making it particularly expensive to include the effects of on-the-fly radiation in hydrodynamic simulations. This motivates the development of surrogate models that can significantly accelerate radiative transfer calculations while preserving high accuracy. We present a surrogate model based on a Fourier Neural Operator architecture combined with U-Nets. Our model approximates three-dimensional, monochromatic radiative transfer in time-dependent regimes, in absorption-emission approximation, achieving speedups of more than 2 orders of magnitude while maintaining an average relative error below 3%, demonstrating our approach's potential to be integrated into state-of-the-art hydrodynamic simulations.

1 Introduction

Radiative transfer is a cornerstone of computational astrophysics, providing the essential link between physical models and observational diagnostics [e.g. 24, 3]. Simulating the propagation of radiation through astrophysical media, such as stellar atmospheres, interstellar clouds, or galaxy clusters, requires solving the radiative transfer equation (RTE), shown in Equation (1).

$$\frac{1}{c}\frac{\partial I_\nu}{\partial t} + \omega \cdot \nabla I_\nu + (k_{\nu,s} + k_{\nu,a})\rho I_\nu = j_\nu \rho + \frac{k_{\nu,s}\rho}{4\pi}\int_{\mathcal{S}} I_\nu \, dw' \tag{1}$$

The RTE is a Partial Differential Equation (PDE) that is often coupled to hydrodynamics, thermochemistry, and dust physics. The quantity of interest is the spectral radiative intensity I_ν, whose evolution is determined by the scattering and absorption coefficients $k_{\nu,s}$ and $k_{\nu,a}$, the density ρ, the emission j, and the integral term representing scattering. Due to its high dimensionality (dependence on time t, spatial position x, direction ω, and frequency ν), the RTE is highly complex and computationally expensive to solve numerically. This has motivated the development of a wide range of numerical techniques, including Monte Carlo methods [20], ray tracing [25], moment-based approximations [24], and RT approximations using the gravity tree [21, 13]. Unfortunately, numerical methods often suffer from high computational costs, dimensionality issues, or instability. In particular, computational cost becomes a major concern when incorporating on-the-fly radiative transport into hydrodynamic simulations to account for the effects of radiation on gas thermodynamics (via photoheating) and dynamics (via radiation pressure). In moment-based approaches, the large value of the speed of light necessitates much smaller time steps ($c >> v_{gas}$); in ray-tracing approaches, memory requirements are the primary bottleneck, scaling as $\mathcal{O}(N_{sources}N_{cells})$.

1st Workshop on Differentiable Systems and Scientific Machine Learning EurIPS 2025.

While traditional deep learning approaches and Physics-Informed Neural Networks have been explored for developing surrogate models for the simulation of radiative transfer [19, 18, 27], they often struggle with generalization across discretizations and parameter settings, and are further known to face challenges with stability and accuracy in such high-dimensional PDE problems [28, 7]. Furthermore, to the best of our knowledge, no prior work has presented an emulator for time-dependent radiative transfer. We therefore propose a data-driven approach based on Neural Operators, a recently introduced class of deep learning architectures inspired by the Universal Approximation Theorem for Operators [6], to develop a surrogate model for simulating radiative transfer. Operating on infinite-dimensional function spaces, Neural Operators extend traditional neural networks, enabling the efficient approximation of PDE solution operators with improved generalization across grid discretizations and parameter variations. We present a Neural Operator-based surrogate model prototype designed to replace conventional on-the-fly radiative transfer solvers within hydrodynamic simulations, achieving significant speedups while maintaining high accuracy.

2 Surrogate Model Architecture

Neural Operators are deep learning architectures that, unlike traditional neural networks, learn mappings between infinite-dimensional function spaces, capturing intrinsic properties rather than discretization-specific details, and thus enable more robust and broadly applicable models. This work employs a specific class of Neural Operators known as the Fourier Neural Operator (FNO) [16] and combines it with a U-Net architecture, following the approach chosen in [29]. The high-level architecture is shown in Figure 1. For simplicity, all visualizations in this figure use 2D slices, even though the model in this work is trained and evaluated using full 3D data.

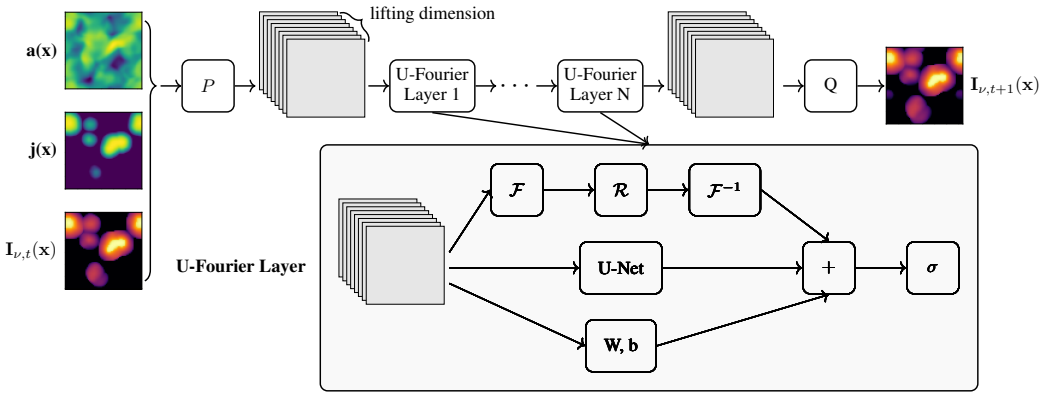

Figure 1: Schematic overview of U-FNO architecture. Input fields are lifted to a higher-dimensional latent space via a lifting layer $\mathcal{P}$, processed by U-Fourier Layers that combine a convolution integral (via Fast Fourier Transform (FFT) $\mathcal{F}$, multiplication with weights $\mathcal{R}$, and inverse FFT $\mathcal{F}^{-1}$), a U-Net, and an affine transformation (W, b), and finally projected to the output field via $\mathcal{Q}$.

In brief, a U-FNO comprises a lifting layer $\mathcal{P}$, which maps the input function to a higher-dimensional latent space, thereby enhancing the model's expressivity. This is followed up by a series of U-Fourier Layers, which combine a convolutional integral operator with an affine transformation and a U-Net. To compute the convolutional integral operator efficiently, inputs are transformed into Fourier space via FFT, multiplied with learnable weights $\mathcal{R}$, and afterward transformed back via inverse FFT, allowing the network to capture global dependencies. A U-Net is an encoder–decoder convolutional network with symmetric downsampling and upsampling paths linked by skip connections, enabling precise localization of fine-scale details [23], particularly if kernels are small. This is especially relevant given that the data used in this study is highly complex, including sharp gradients and discontinuities, and U-Nets have been shown to excel in such settings [e.g. 5]. Finally, a projection layer $\mathcal{Q}$ maps the latent representation to the desired output dimension.

3 Results

In this study, we present a U-FNO-based surrogate model designed as a drop-in replacement for on-the-fly radiative transfer solvers in hydrodynamic simulations. The model predicts the temporal evolution of radiative intensity through recurrent application over time steps, providing an efficient alternative to computationally expensive numerical solvers. To align with state-of-the-art hydrodynamic simulations, we consider radiative transfer in a scattering-free regime, as scattering is typically neglected in on-the-fly computations. Moreover, the angular dependence of the radiative intensity is currently omitted but planned for future inclusion. Accordingly, the model is trained to predict $I_{\nu,t+1}(\mathbf{x})$ from $I_{\nu,t}(\mathbf{x})$, $a(\mathbf{x}) = k_{\nu,a}\rho(\mathbf{x})$, and $j(\mathbf{x}) = j_\nu\rho(\mathbf{x})$, as illustrated in Figure 1. While the presented model focuses on a single frequency ν, modern hydrodynamic simulations typically partition the radiation spectrum into a small number of frequency (energy) bins ($\sim$10), centered on the photoionization and photodissociation energies of key gas-phase species [14, 24, 8]. Since these bins are independent in the absence of scattering, our approach naturally scales to such multi-frequency setups by training one model per frequency bin, ensuring consistent generalization regardless of the number of bins used in the simulation. Alternatively, a unified model can be trained to jointly predict the evolution across all frequencies (see Appendix A for details). The model is implemented in JAX [2], leveraging its efficient support for just-in-time compilation and GPU acceleration.

To train and evaluate the model, we generate a diverse dataset of absorption and emission fields based on turbulent periodic boxes produced with the hydrodynamic code `jf1uids` [26] in a 64^3 domain. To ensure sufficiently heterogeneous density– and thus opacity–fields, we vary the turbulence random seed, the amplitude of velocity fluctuations, and the slope of the turbulent kinetic-energy power spectrum across simulations, running each simulation until the turbulence spectrum reaches a stable equilibrium. This setup approximates conditions closely resembling those in giant molecular clouds (star-forming regions) that are among the most extensively studied numerically for their radiative effects [8]. The absorption field $a(\mathbf{x})$ is correlated with the density field, and radiation sources $j(\mathbf{x})$ are placed in the top 1.5% of the density field. For each pair ($a(\mathbf{x})$, $j(\mathbf{x})$), the corresponding radiative intensity evolution is computed over ten time steps, starting from $I_{\nu,0}(\mathbf{x})$=0, using ray tracing to complete the dataset (see Appendix C). The dataset is split into a training (70%), validation (10%), and test (20%) set. Training leverages pairs of consecutive intensities ($I_{\nu,t}(\mathbf{x}), I_{\nu,t+1}(\mathbf{x})$) alongside the corresponding absorption and emission fields to train the model for next-state intensity predictions, and key hyperparameters are optimized using Optuna [1]. To stabilize the training process, all fields are log-transformed and min–max normalized to the range $[0, 1]$. Additionally, we employ a relative loss combining pixelwise and spatial gradient differences to encourage sharp feature reconstruction (see Appendix E for training details). Training and evaluation are performed on an NVIDIA H100 GPU. During inference, full temporal evolution is obtained by recursively feeding predictions back as input. Figure 2 compares the preprocessed numerically computed (top row) and the predicted (middle row) temporal evolution of the radiative intensity at the midplane (z=32) for a random test set sample. Additional cross-sections as well as the absorption and emission fields are shown in Appendixes C and G.

The model accurately captures the temporal evolution of the radiative intensity. Predictions closely match the reference solutions, and residuals remain low, with deviations mainly near edges around evolving structures. The shown sample reflects the model's overall test performance, with an average relative error of 2.9% per pixel for next-state predictions, highlighting its ability to generalize across varying scenarios. This accuracy aligns with state-of-the-art works [27, 19]. Notably, the integration of the U-Net significantly improves performance, as a pure FNO model only reaches a relative error of 60%. Due to the recurrent prediction scheme, errors accumulate over time but remain small, as shown by the residuals (see Appendix F for a detailed assessment). Crucially, the surrogate model achieves a speedup of ~$600\times$, producing full spatiotemporal predictions in 0.1s compared to 59.2s for the numerical solver. Additionally, memory costs do not scale with the number of sources, addressing a key computational bottleneck of ray-tracing algorithms and further highlighting the surrogate's potential as a drop-in replacement for on-the-fly solvers in hydrodynamic simulations.

To contextualize the performance of our approach relative to existing emulators, we also apply it to monochromatic three-dimensional static radiative transfer, neglecting scattering and angular dependence. Our model achieves a speedup of more than three orders of magnitude while maintaining a relative error of 2.6%, matching or surpassing the results reported in other 3D static RT emulators [27]. Further details are provided in Appendix D.

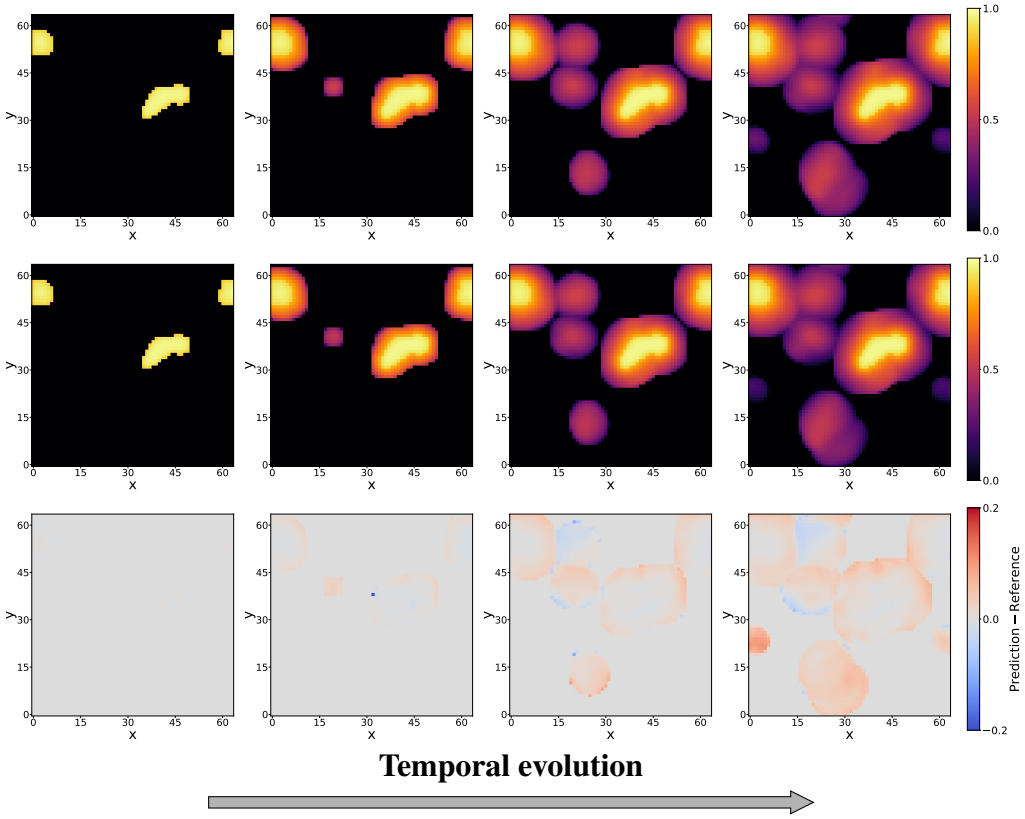

Figure 2: Selected timesteps of the temporal evolution of radiative intensity at z=32: preprocessed numerical reference (top row), model prediction (middle row), and residual (bottom row).

4 Discussion

Summary. Simulating radiative transfer numerically is computationally demanding, especially when included on-the-fly in hydrodynamic simulations. We present a U-FNO-based surrogate model prototype, designed as a drop-in replacement for conventional on-the-fly solvers. The model predicts the temporal evolution of radiative intensity via recurrent application, closely matching numerical references with an average relative error of 2.9% for next-state predictions and achieving a speedup of ~600× over state-of-the-art ray tracing, with memory costs independent of the number of sources. Demonstrated for a single frequency, this approach generalizes naturally to multi-frequency setups by training one model per frequency bin. All code required to reproduce our experiments is publicly available[1].

Limitations. Due to the integrated U-Nets, the current U-FNO model is not invariant to discretization or resolution, lacking the flexibility of simpler FNO-based architectures. In fixed-grid scenarios, however, this limitation is not critical, and additional techniques also offer resolution-independent formulations [29]. Moreover, as the U-FNO predicts the entire field, localized predictions cannot be made in isolation, leading to unnecessary overhead when only specific regions are of interest. Our current model demonstrates predictions for only a single frequency, however, separate models can be trained for each frequency bin for integration into multi-frequency simulations.

Outlook. Future research will extend our models to more general approximations of the RTE solution operator by also considering the angular dependency of the radiative intensity. Moreover, it will focus on training a unified model capable of predicting the temporal evolution of radiative intensity across multiple frequency bins simultaneously. Preliminary results are shown in Appendix A. Furthermore, we plan to integrate the developed surrogate model into full simulation pipelines to accelerate computations and reduce memory costs.

[1] `https://github.com/RuneRost/Astro-RT.git`

Acknowledgments and Disclosure of Funding

This work is funded by the Carl-Zeiss-Stiftung through the NEXUS program. This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster). We acknowledge the usage of the AI-clusters Tom and Jerry funded by the Field of Focus 2 of Heidelberg University.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A Next-generation Hyperparameter Optimization Framework. *arXiv e-prints*, page arXiv:1907.10902, July 2019.

[2] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[3] Tobias Buck, Andrea V. Macciò, Aura Obreja, Aaron A. Dutton, Rosa Domínguez-Tenreiro, and Gian Luigi Granato. NIHAO XIII: Clumpy discs or clumpy light in high-redshift galaxies? *Monthly Notices of the Royal Astronomical Society*, 468(3):3628–3649, July 2017.

[4] Tobias Buck, Christoph Pfrommer, Philipp Girichidis, and Bogdan Corobean. Escaping the maze: a statistical subgrid model for cloud-scale density structures in the interstellar medium. *Monthly Notices of the Royal Astronomical Society*, 513(1):1414–1428, June 2022.

[5] Tobias Buck and Steffen Wolf. Predicting resolved galaxy properties from photometric images using convolutional neural networks. *arXiv e-prints*, page arXiv:2111.01154, November 2021.

[6] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its applications to dynamic systems. *Neural Networks, IEEE Transactions on*, pages 911 – 917, 08 1995.

[7] Maarten V. de Hoop, Daniel Zhengyu Huang, Elizabeth Qian, and Andrew M. Stuart. The Cost-Accuracy Trade-Off In Operator Learning With Neural Networks. *arXiv e-prints*, page arXiv:2203.13181, March 2022.

[8] D. Decataldo, A. Lupi, A. Ferrara, A. Pallottini, and M. Fumagalli. Shaping the structure of a GMC with radiation and winds. *Monthly Notices of the Royal Astronomical Society*, 497(4):4718–4732, October 2020.

[9] Yuwei Fan, Cindy Orozco Bohorquez, and Lexing Ying. BCR-Net: A neural network based on the nonstandard wavelet form. *Journal of Computational Physics*, 384:1–15, May 2019.

[10] Christoph Federrath and Ralf S. Klessen. The Star Formation Rate of Turbulent Magnetized Clouds: Comparing Theory, Simulations, and Observations. *The Astrophysical Journal*, 761(2):156, December 2012.

[11] Philipp Girichidis, Lukas Konstandin, Anthony P. Whitworth, and Ralf S. Klessen. On the Evolution of the Density Probability Density Function in Strongly Self-gravitating Systems. *The Astrophysical Journal*, 781(2):91, February 2014.

[12] Chiyu "Max" Jiang, Karthik Kashinath, Prabhat, and Philip Marcus. Enforcing physical constraints in CNNs through differentiable PDE layer. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2019.

[13] R. Kannan, G. S. Stinson, A. V. Macciò, J. F. Hennawi, R. Woods, J. Wadsley, S. Shen, T. Robitaille, S. Cantalupo, T. R. Quinn, and C. Christensen. Galaxy formation with local photoionization feedback - I. Methods. *Monthly Notices of the Royal Astronomical Society*, 437(3):2882–2893, January 2014.

[14] Rahul Kannan, Mark Vogelsberger, Federico Marinacci, Ryan McKinnon, Rüdiger Pakmor, and Volker Springel. AREPO-RT: radiation hydrodynamics on a moving mesh. *Monthly Notices of the Royal Astronomical Society*, 485(1):117–149, May 2019.

[15] Nikola B. Kovachki, Samuel Lanthaler, and Andrew M. Stuart. Operator Learning: Algorithms and Analysis. *arXiv e-prints*, page arXiv:2402.15715, February 2024.

[16] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier Neural Operator for Parametric Partial Differential Equations. *arXiv e-prints*, page arXiv:2010.08895, October 2020.

[17] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. *arXiv e-prints*, page arXiv:1711.05101, November 2017.

[18] Xiaoyi Lu and Yi Wang. Surrogate modeling for radiative heat transfer using physics-informed deep neural operator networks. *Proceedings of the Combustion Institute*, 40(1):105282, 2024.

[19] Siddhartha Mishra and Roberto Molinaro. Physics informed neural networks for simulating radiative transfer. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 270:107705, August 2021.

[20] Ulrich M. Noebauer and Stuart A. Sim. Monte carlo radiative transfer. *Living Reviews in Computational Astrophysics*, 5(1):1, 2019.

[21] Aura Obreja, Andrea V. Macciò, Benjamin Moster, Silviu M. Udrescu, Tobias Buck, Rahul Kannan, Aaron A. Dutton, and Marvin Blank. Local photoionization feedback effects on galaxies. *Monthly Notices of the Royal Astronomical Society*, 490(2):1518–1538, December 2019.

[22] Paolo Padoan and Åke Nordlund. The Star Formation Rate of Supersonic Magnetohydrodynamic Turbulence. *The Astrophysical Journal*, 730(1):40, March 2011.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, page arXiv:1505.04597, May 2015.

[24] J. Rosdahl, J. Blaizot, D. Aubert, T. Stranex, and R. Teyssier. RAMSES-RT: radiation hydrodynamics in the cosmological context. *Monthly Notices of the Royal Astronomical Society*, 436(3):2188–2231, December 2013.

[25] J. Steinacker, A. Bacmann, and T. Henning. Ray Tracing for Complex Astrophysical High-opacity Structures. *The Astrophysical Journal*, 645(2):920–927, July 2006.

[26] Leonard Storcks and Tobias Buck. Differentiable Conservative Radially Symmetric Fluid Simulations and Stellar Winds – jf1uids. *arXiv e-prints*, page arXiv:2410.23093, October 2024.

[27] Shiqi Su, Frederik De Ceuster, Jaehoon Cha, Mark I. Wilkinson, Jeyan Thiyagalingam, Jeremy Yates, Yi-Hang Zhu, and Jan Bolte. Emulating CO Line Radiative Transfer with Deep Learning. *arXiv e-prints*, page arXiv:2507.11398, July 2025.

[28] Han Wan, Rui Zhang, and Hao Sun. Spectral-inspired Neural Operator for Data-efficient PDE Simulation in Physics-agnostic Regimes. *arXiv e-prints*, page arXiv:2505.21573, May 2025.

[29] Gege Wen, Zongyi Li, Kamyar Azizzadenesheli, Anima Anandkumar, and Sally M. Benson. U-FNO-An enhanced Fourier neural operator-based deep-learning model for multiphase flow. *Advances in Water Resources*, 163:104180, May 2022.

Appendix

The following appendix contains additional material complementing the main text, including architectural details, dataset descriptions, model and training configurations, and additional evaluation results. Moreover, we provide an outlook on future research by presenting preliminary results from a unified model currently being trained to predict the temporal evolution of radiative intensity across multiple frequency bins simultaneously.

A Outlook

Our current research focuses on training a unified model that jointly predicts the temporal evolution of radiative intensity across multiple frequency bins simultaneously. Similar to the monochromatic setup presented in the main part, we generate absorption and emission fields based on turbulent periodic boxes produced with the hydrodynamic code `jf1uids` [26]in a 64^3 domain. However, instead of a single frequency, we now consider six frequency bins and vary $j(\mathbf{x}) = j_\nu \rho$ and $a(\mathbf{x}) = k_{\nu,a}\rho$ across bins. For each frequency bin, the corresponding radiative intensity evolution is computed via ray tracing. Training is then performed using pairs of consecutive intensity fields ($\{I_{\nu_k,t}(\mathbf{x})\}_{k=1}^6$, $\{I_{\nu_k,t+1}(\mathbf{x})\}_{k=1}^6$) Figure 3 illustrates selected time steps of the models' predicted temporal evolution of radiative intensity for three representative frequency bins, where each row corresponds to a different frequency bin.

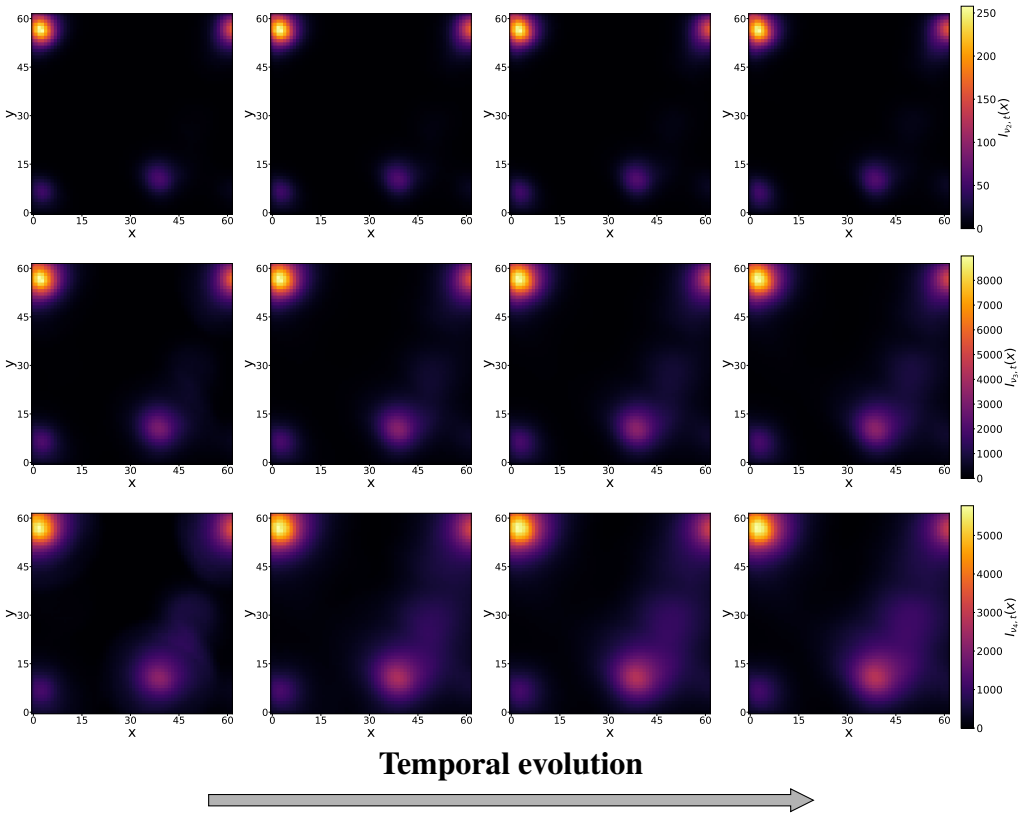

Figure 3: Selected timesteps of the temporal evolution of radiative intensity at z=32 for three representative frequency bins: each row corresponds to a different frequency bin.

As shown, the frequency dependence of the absorption results in different temporal evolutions of the radiative intensity across the different frequency bins. This is intended purely as an outlook on ongoing work, as the relative error is roughly two magnitudes higher than for the single-frequency model presented in the main part. Hence, further architecture– and hyperparameter–optimization is necessary to account for the increased data complexity. This is also the reason why we do not include a comparison to the numerical reference here. Due to the large size of the data ($6 \times 64 \times 64 \times 64 \approx 1.5$

million pixels for a single time step), training such a unified model is highly computationally demanding. Consequently, it still may often remain advantageous to generalize the approach from the main part by training separate models for individual frequency bins.

B Details of the U-FNO Architecture.

Due to the page limit of the main part, a comprehensive description of the basic model architecture used in this work is provided here. For our models, we combined a typical FNO architecture with additional U-Nets in each Fourier layer, following the approach in [29] to build models capable of capturing global dependencies as well as modeling fine-scale features.

The original FNO architecture was proposed in 2020 in [16] and relies on the use of the Fourier Transformation. Motivated by the previous extension of PDE-solving Fourier spectral methods to neural networks [9, 12], FNOs have since then proven to be very effective in solving high-dimensional PDEs. As a class of Neural Operators, they can be seen as a generalization of traditional neural networks that enable mapping between infinite-dimensional function spaces. Consequently, unlike in traditional neural networks, their layers do not define an affine transformation but instead rely on integral operators that capture intrinsic properties of the input function. A key feature of FNOs is that the integral operator can be represented as a convolution, which allows for computation via the Fourier Transformation. Specifically, the input is transformed to Fourier space, multiplied with learnable weights, and then transformed back using inverse Fourier Transformation. To implement this in practice, functions are typically discretized on a uniform, equidistant grid, resulting in a finite-dimensional tensor representation. This discretization enables efficient evaluation of the integral operator via the Fast Fourier Transform (FFT), as implemented in modern deep learning libraries [15]. To further improve computational efficiency, the resulting Fourier coefficients are truncated beyond a fixed number of modes K in each dimension, with the remaining entries set to zero. This reduces model complexity and mitigates overfitting by concentrating on the most relevant frequency components and thereby encourages smoother, more stable predictions. The convolutional integral operator is combined with an affine transformation. Together, this makes up a so-called Fourier layer.

As data gets more complex, FNOs often struggle to model fine-scale details, resulting in blurry predictions that lack sharpness and thus fail to accurately model steep gradients and discontinuities. Therefore, in our model the previously described Fourier layers are extended with U-Nets [23], resulting in the U-Fourier Layers. As mentioned in the main part, a U-Net is an encoder–decoder convolutional network with symmetric downsampling and upsampling paths linked by skip connections. By choosing small kernels with sizes of 2 or 3 in each spatial dimension, the additional U-Nets allow our model to precisely locate fine-scale details in the prediction of the intensity patterns.

In the final architecture, a concatenation of U-Fourier Layers is preceded by a lifting layer that maps the input into a higher-dimensional latent space and followed by a projection layer that maps it back to the desired output dimension.

C Dataset

In the following, we provide a detailed description of the dataset used to train the surrogate model we present in the main part. As described there, this dataset consists of samples that each contain an absorption field and an emission field, along with ten snapshots representing the respective temporal evolution of the radiative intensity starting from $I_{\nu,0}(\mathbf{x}) = 0$ and evolving in fixed-size time steps.

As described in the main part, the absorption and emission fields are generated based on turbulent periodic boxes produced with the hydrodynamics code `jf1luids` [26] in a 64^3 domain. To produce sufficiently heterogeneous density– and thus opacity–fields, we vary the turbulence random seed, the amplitude of velocity fluctuations, and the slope of the turbulent kinetic-energy power spectrum across simulations and run each simulation until the turbulence spectrum reaches a stable equilibrium. This setup allows us to mimic conditions typical of giant molecular clouds (star-forming regions) that are among the most extensively studied numerically for their radiative effects [8]. Subsequently, the absorption field $a(\mathbf{x})$ is correlated with the density field, and radiation sources $j(\mathbf{x})$ are placed in the top 1.5% of the density field. As a result, the network inputs span a wide variety of structures with statistical properties directly inherited from the underlying fluid dynamics.

To complete the dataset, we employ a ray tracing algorithm[2] to compute the corresponding radiative intensity evolution across ten time steps for each pair $(a(\mathbf{x}), j(\mathbf{x}))$. The code determines the radiative intensity by following rays emitted from the sources as they propagate through the domain and interact with matter. To align with state-of-the-art hydrodynamic simulations, we consider radiative transfer in a scattering-free regime, as scattering is typically neglected in on-the-fly computations. Moreover, the angular dependence of the radiative intensity is currently omitted by summing the total intensity in each pixel but is planned for future inclusion. The entire radiation code is implemented in JAX [2] and exploits parallelization to achieve highly efficient computations. Starting from $I_{\nu,0}(\mathbf{x})$=0, the ray tracing code computes the radiative intensity $I_{\nu_k,t}(\mathbf{x})$ based on $(a(\mathbf{x}), j(\mathbf{x}))$ for ten consecutive, equally spaced time steps. In total, 400 pairs of absorption and emission fields are simulated, and for each pair, a full ray-tracing simulation is performed to compute the corresponding temporal evolution of the radiative intensity.

For training the surrogate model on next-state intensity prediction, this dataset needs to be further processed. Each sample $(a(\mathbf{x}), j(\mathbf{x}), I_{\nu,0}(\mathbf{x}), \ldots, I_{\nu,9}(\mathbf{x}))$ is split up into 9 new samples of the form $(a(\mathbf{x}), j(\mathbf{x}), I_{\nu,t}(\mathbf{x}), I_{\nu,t+1}(\mathbf{x}))$, with each new sample containing two consecutive radiative intensity fields, as well as the corresponding absorption and emission fields. Thus, the network can be trained to predict the radiative intensity at a given time t (in the sample shown in Figure 4: t=9) based on an absorption and an emission field as well as the radiative intensity at time t-1 (in the sample shown in Figure 4: t-1=8). The final dataset therefore contains 3600 samples, which proves to be sufficient given Neural Operators' ability to leverage the high functional diversity in the training data.

Figure 4 shows the absorption field and the emission field, as well as the radiative intensity at the midplane (z=32) at time t=8 and t=9 from the test sample whose temporal intensity evolution is shown in Figure 2. During inference, full temporal evolution is obtained by recursively feeding predictions back as input. Unlike in the main part, we show the unpreprocessed fields, so the intensity map appears different than in Figure 2.

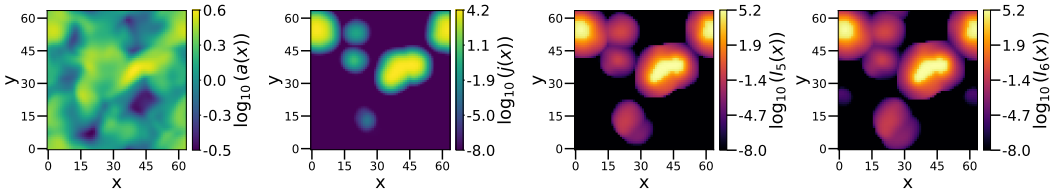

Figure 4: Absorption field (first image), emission field (second image), and the corresponding numerically computed radiative intensity at z=32 at time step t=8 (third image) and t=9 (fourth image) of the test sample for which the comparison between prediction and numerical reference is shown in Figure 2.

As this illustrates, both the inputs and outputs represent highly complex functions, posing a particularly challenging learning task that highlights the effectiveness of the proposed architecture.

D Steady-state predictions

As mentioned in the main part, to contextualize the performance of our approach relative to existing emulators, we also apply it to monochromatic three-dimensional static radiative transfer, neglecting scattering and angular dependence.

Therefore, we generate a dataset consisting of diverse samples, each comprising an absorption field $a(\mathbf{x}) = k_{\nu,a}\rho(\mathbf{x})$ and an emission field $j(\mathbf{x}) = j_\nu \rho(\mathbf{x})$ as input, and the resulting steady-state radiative intensity $I_\nu(\mathbf{x})$ as target. To construct these fields, the density in a 64^3 domain is modeled as a log-normal random field, mimicking the structure of a turbulent star-forming region [e.g. 22, 10, 11, 4]. High-density regions (top 1% of pixels) are masked as belonging to a star in order to assign them as centers for Gaussian emissivity profiles to mimic stellar emission for $j(\mathbf{x})$. Additionally, the absorption field is defined as $a = k_{\nu,a}\rho(\mathbf{x})$, where $k_{\nu,a}$, which determines the medium's opacity, is set to one for simplicity. Varying the random seed of the density field initialization ensures diverse, non-redundant samples.

[2]Code available at `https://github.com/lorenzobranca/Ray-trax.git`.

To determine the radiative intensity for each pair of absorption and emission fields, we employ a tracing algorithm[3]. As before, we restrict the calculation to monochromatic radiation and neglect scattering while also omitting angular dependence by summing the total intensity in each pixel. For each pair of absorption and emission fields, the radiative intensity is computed in the limit for $t \to \infty$, i.e. the equilibrium setting in after a long time. In total, 1000 samples are generated, a number sufficient given the ability of Neural Operators to leverage the high functional diversity in the training data. Modeling the density as a log-normal random field ensures that input fields span a wide variety of structurally different functions rather than mere parametrized variants of one profile. Consequently, learning a mapping from inputs to their corresponding outputs represents a highly complex task, thereby demonstrating the Neural Operators' capability to process and generalize across complex function spaces.

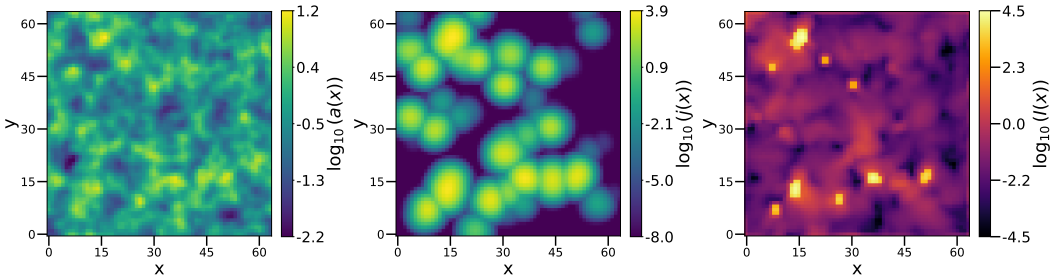

Figure 5: Absorption field (left), emission field (middle), and the corresponding numerically computed steady-state radiative intensity (right) of the test sample for which the comparison between prediction and numerical reference is shown in Figure 6.

Figure 5 shows an example input, namely an absorption field (left panel) and an emission field (middle panel), together with the corresponding target, i.e. the numerically computed steady-state radiative intensity field (right panel). This is the same sample for which the comparison between numerical reference and prediction is shown in Figure 6. Similar to this figure, the cross-section at the mid-plane at z=32 is displayed. Note that Figure 5 displays the unpreprocessed fields, so the intensity map appears different than in Figure 6. As Figure 5 shows, the inputs represent highly complex functions, producing intricate intensity patterns that the surrogate model must learn to reproduce.

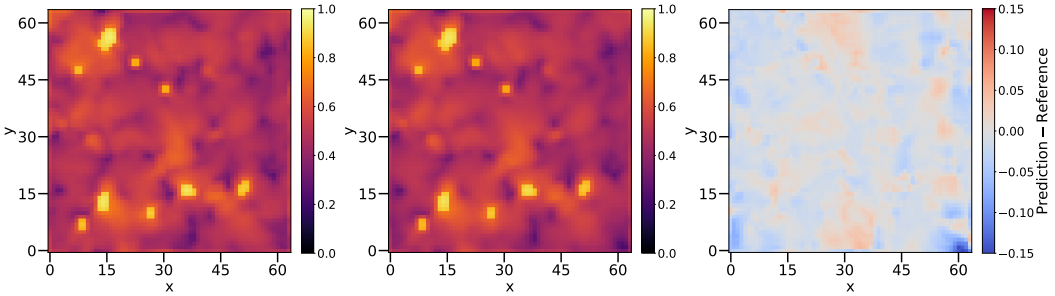

Figure 6: Comparison of steady-state radiative intensity at z=32: preprocessed numerical reference (left), model prediction (middle), and corresponding residual (right).

For the training, the dataset is split into a training (70%), validation (10%), and test (20%) set, and key hyperparameters are optimized using Optuna [1] (see Appendix E for details). To stabilize the training process, all fields are log-transformed and subsequently min–max normalized to the range $[0, 1]$. Additionally, a relative loss combining pixelwise and spatial gradient differences is used to encourage sharp feature reconstruction. Training and evaluation are performed on a single NVIDIA A100 GPU.

Figure 6 shows a comparison of the preprocessed (log-transformed and min-max normalized) numerically computed (left panel) and the predicted (middle panel) radiative intensity for a random sample from the test set at the mid-plane (z=32). Additional cross-sections are shown in Appendix G.

[3]Code available at `https://github.com/lorenzobranca/Ray-trax.git`.

The predicted intensity closely matches the numerical reference. The sample reflects the model's overall test performance, demonstrating its ability to accurately approximate the underlying PDE, with a mean relative error of 2.6% per pixel across the test set. Notably, the surrogate model offers a speedup of ~$6750\times$, achieving predictions in 0.003s compared to 20.3s for the numerical solver, and memory costs are independent of the number of sources. The results match or even surpass the results reported in other 3D static RT emulators [27].

E Hyperparameters of Models and Training Procedure

The following section provides additional implementation details on the hyperparameters of the surrogate model we present in the main part as well as the training setup and procedure. We additionally include this information for the previously mentioned model for steady-state radiative intensity predictions. Both models are trained using the Adam optimizer with decoupled weight decay (AdamW) [17]. The learning rate follows an exponential schedule. Since the pixel values in the inputs and outputs spanned over multiple orders of magnitude, the data is preprocessed prior to training. First, the logarithm is applied, followed by a min-max normalization to scale the data to the interval of $[0, 1]$. This is performed independently for absorption, emission, and intensity fields. Following the approach in [29], we employ a loss function that additionally considers the spatial gradients of the predictions to encourage sharp feature reconstruction, resulting in a relative loss that accounts for deviations both in pixel values and in their spatial gradients. For a single prediction, its mathematical formulation is given by:

$$\mathcal{L} = \frac{\sqrt{\sum_{i=1}^{N}(\hat{f}_i - f_i)^2}}{\sqrt{\sum_{i=1}^{N}(f_i)^2}} + \lambda \frac{\sqrt{\sum_{i=1}^{N}(\nabla \hat{f}_i - \nabla f_i)^2}}{\sqrt{\sum_{i=1}^{N}(\nabla f_i)^2}} \tag{2}$$

Here, i indexes the N pixels, which are flattened into a one-dimensional vector. f_i is the target value for pixel i, $\hat{f}_i$ the corresponding prediction, and ∇ represents the spatial derivative.

To achieve optimal performance, most hyperparameters for both the training procedure and the model architectures are optimized using Optuna. Optuna is an open source library for automatic hyperparameter optimization in machine learning, which enables efficient and systematic identification of the best hyperparameters for a given model [1]. The following tables list the values of the final key hyperparameters used for the model architectures and the model training. Table 1 shows the hyperparameters for the steady-state model, while Table 2 lists those for the recurrent model presented in the main part. Among these hyperparameters, only the number of layers, the U-Net kernel size, and the loss-function parameter λ are set manually rather than optimized with Optuna. The number of layers and the U-Net kernel size are fixed manually due to architectural and memory constraints, and λ is set based on previous studies indicating that the chosen value provides a good compromise between predictive accuracy and sharpness.

For both models, we observe that training is robust to small variations in these hyperparameters. The ranges of Fourier modes and network widths tested were deliberately limited, as increasing them substantially raises computational costs while yielding only marginal improvements. Consequently, the models use a relatively low number of modes and a modest width, which proves sufficient to accurately capture the underlying mappings and produce highly precise predictions. Additionally, manual tests of U-Net architectures with varying kernel sizes confirmed that smaller kernels better enable the models to resolve fine-scale details.

The steady-state model is trained on 700 samples (70% of the dataset for the scenario of steady-state radiative intensity) for 40 epochs, taking approximately 80 minutes. The recurrent model from the main part is trained for 20 epochs. Since each of the 280 full-simulation samples (70% of the dataset for the scenario of the temporal evolution of radiative intensity) is split into 9 training samples, the recurrent model is trained on substantially more training data. Combined with the model's increased width, this results in a considerably longer training time of approximately 320 minutes.

Table 1: Hyperparameters of Architecture and Training Procedure for the Steady-State Model.

Model		Training	
Number of Layers	6	Initial Learning Rate	0.0005
Layer Width	16	Decay Rate	0.9000
Number of Modes	4	Weight Decay	0.0050
U-Net Kernel Size	3	Dropout Probability	0.08
U-Net Width	16	λ in Loss	0.5

Table 2: Hyperparameters of Architecture and Training Procedure for the Recurrent Model.

Model		Training	
Number of Layers	6	Initial Learning Rate	0.0006
Layer Width	32	Decay Rate	0.9120
Number of Modes	4	Weight Decay	0.0052
U-Net Kernel Size	2	Dropout Probability	0.08
U-Net Width	32	λ in Loss	0.5

F Quantitative Analysis of Prediction Accuracy

To evaluate the overall predictive accuracy across the entire test set, we provide histograms of the pixelwise relative deviations. Similar to the previous section, we do this for both the model presented in the main part and the steady-state model. Figure 7 shows the relative deviations between the preprocessed numerically computed steady-state radiative intensity and the steady-state model prediction, computed over all test samples. The histogram indicates that the relative deviations approximately follow a Gaussian distribution. Overall, relative errors remain low across all pixels in the test set, with a slight bias towards underestimation. A plausible explanation is that the network produces smoother predictions, particularly when modeling sharp gradients and discontinuities, leading to reduced extrema. Since the intensity is generally high throughout the domain, these less extreme predictions result in predictions being too small on average and cause a negative bias. Nevertheless, the accuracy is very high, with only a small portion of pixels deviating by more than 8% and an average per-pixel deviation of only 2.6%.

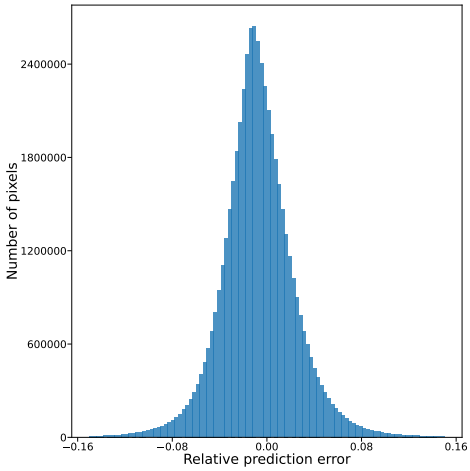

Figure 7: Pixelwise relative deviations between the preprocessed numerically computed steady-state radiative intensity and the model prediction computed over all test samples.

Figure 8 shows the pixelwise relative deviations at each time step of the recurrent radiative intensity predictions, computed over all test samples. The relative deviations remain generally low across

all timesteps. Even in the final time step, the majority of pixels exhibit relative deviations below 10%. However, a gradual decrease in accuracy over time is noticeable, caused by the accumulation of errors inherent to the recurrent prediction approach. As stated in the main part, the average per-pixel relative error for next-state predictions is 2.9%. During inference, however, the model uses its own previous outputs as inputs to predict the full temporal evolution. Consequently, predictions are based on states that already deviate from the numerical reference. The bias towards more positive residuals suggests that this causes a slightly faster evolution of structures in the predictions compared to the reference. Nevertheless, the overall prediction accuracy remains high.

Note that in all histograms presenting relative deviations, we only include pixels for which the preprocessed numerically computed radiative intensity is unequal to zero. Hence, in Figure 8, the histograms for later time steps contain a larger number of entries. Likewise, when computing the relative errors reported in the main part, pixels for which the preprocessed numerically computed radiative intensity is zero are excluded.

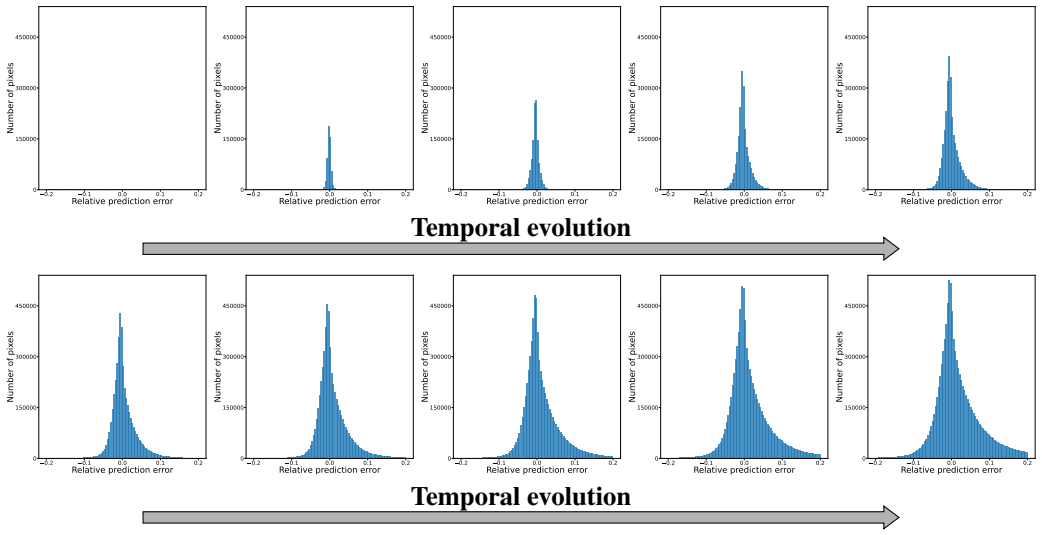

Figure 8: Temporal evolution of pixelwise residuals (upper block) and relative errors (lower block) computed over all test samples: Timesteps proceed from top left to bottom right.

G Extended Cross-Sectional Analysis of Model Predictions

In addition to the z=32 cross-section displayed in Figure 6, Figure 9 for the same test sample presents three cross-sections, at x=32, y=32, and z=32, of the preprocessed (log-transformed and min-max normalized) numerical reference and the corresponding predicted radiative intensity. These are shown together with the corresponding residuals to illustrate the model's performance across all spatial directions and indicate that predictions are consistently accurate. Residuals in the x=32 and y=32 cross-sections are of similar magnitude to those in the z=32 cross-section that was already presented. This confirms the steady-state model's ability to determine the radiative intensity distribution across all spatial dimensions.

Finally, in addition to Figure 2 shown in the main part, Figure 10 for the same test sample shows the complete temporal evolution of radiative intensity along all time steps for cross-sections at x=32, y=32, and z=32. This involves the preprocessed (log-transformed and min-max normalized) numerical reference, the predicted radiative intensity, as well as the corresponding residuals. Similar to the previous figure, this demonstrates the recurrent model's ability to precisely predict the temporal evolution of radiative intensity across all spatial dimensions.

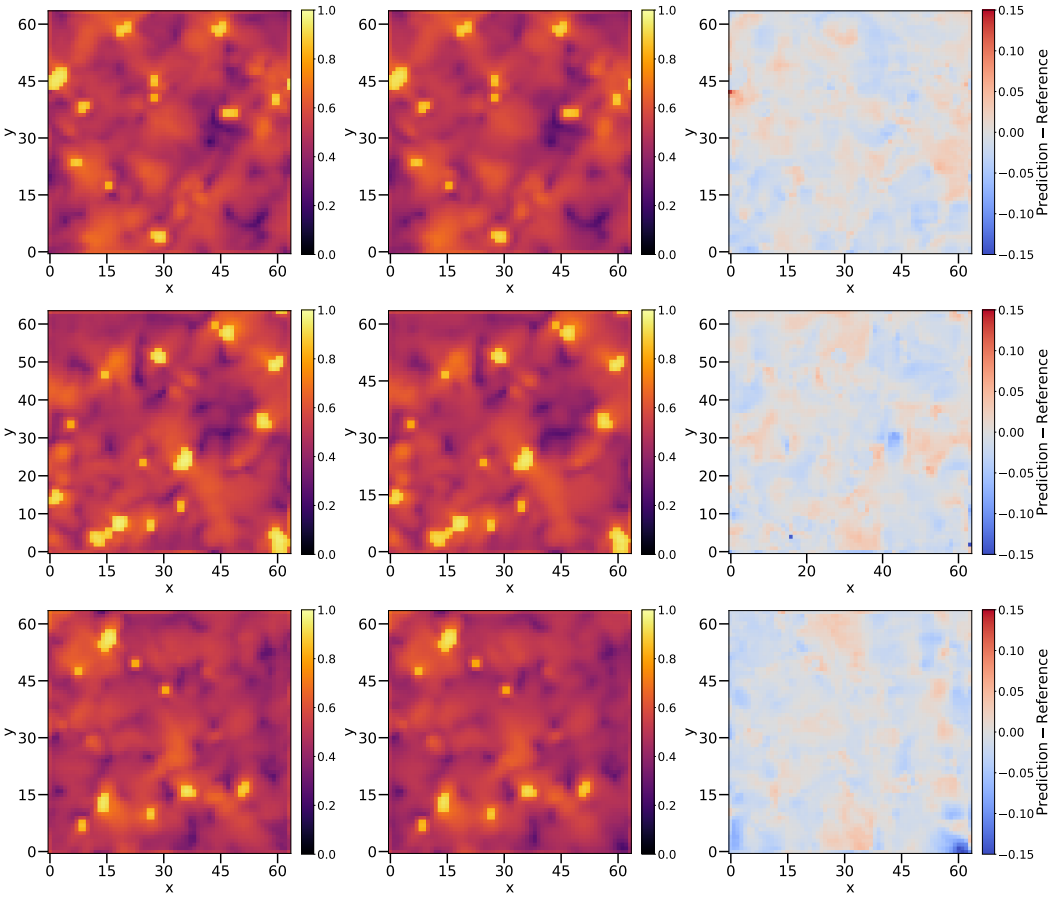

Figure 9: Comparison of the steady-state radiative intensity at cross-section x=32 (first row), y=32 (second row), and z=32 (third row): Respectively, in each row the left image shows the preprocessed numerical reference, the middle image the model prediction, and the right image the corresponding residual.

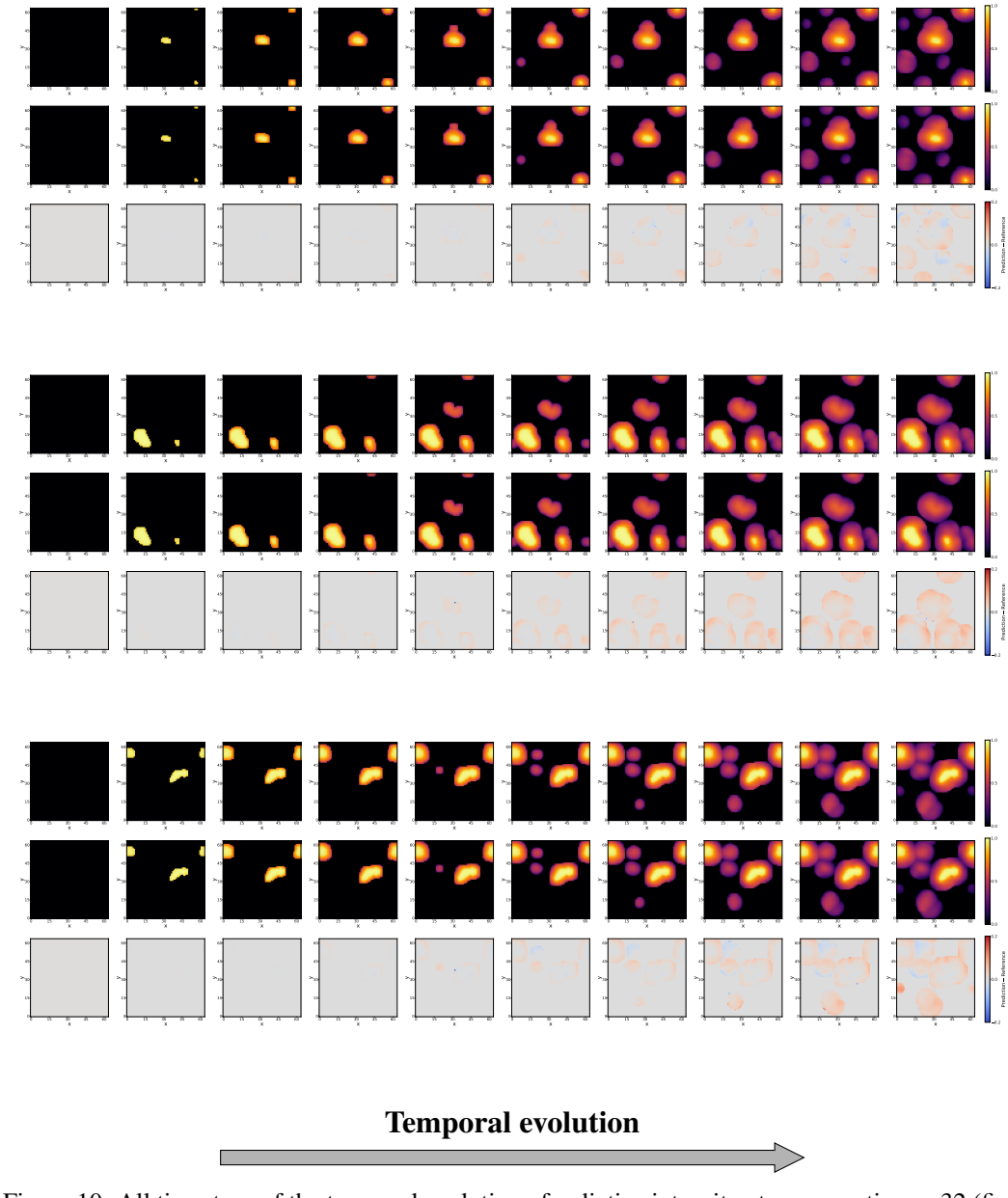

Figure 10: All timesteps of the temporal evolution of radiative intensity at cross-section x=32 (first block), y=32 (second block), and z=32 (third block): Respectively, in each block the top row shows the preprocessed numerical reference, the middle row the model prediction, and the bottom row the residual.