Appendix

A First-principles simulations of laser-plasma interactions

A plasma is a collection of unbound, moving ions and electrons, interacting through the electromagnetic (EM-) fields. It can be produced through for example the irradiation of a high-intensity ($I > 10^{18}$ W/cm^2) laser pulse onto a solid material such as a metal foil. Through the nonlinear interplay between the electrons, ions, and the EM-fields, these laser-plasma interactions can generate high-energy ($>$ MeV) ions for applications in material science (Patel et al., 2003), imaging (Rygg et al., 2008), and medical therapy (Kroll et al., 2022). To understand these nonlinear processes, it is necessary to use kinetic treatment of the plasma.

A.1 Particle and continuum descriptions of plasmas

One example of kinetic treatment of plasma is the particle-in-cell (PIC) method (Dawson, 1983; Birdsall & Langdon, 1991). In this method we solve the Klimontovich equation (Klimontovich 1967) for finite size particles, coupled to Maxwell's equations for the EM-fields. The numerical procedure consists of solving Maxwell's equations on a spatial grid using the current and charge densities that are obtained by weighting the discrete plasma particles onto the grid. The particles are then advanced via the Lorentz force associated with the EM-fields. This particle-based simulation technique captures the kinetic microphysics of plasmas, and to the extent that quantum mechanical effects can be neglected, provides a first-principles description of plasma dynamics.

A plasma can also be described exactly as a continuum, by fluid equations – the evolution of the velocity moments of the distribution function f of the plasma particles ($M_N(\mathbf{x}) := \int d^N v f(\mathbf{x}, \mathbf{v}) \mathbf{v}^N$ is the N-th order velocity moment). The fluid equations can be derived from the kinetic equations and form an infinite hierarchy of exact coupled conservation equations for each fluid moment (with the n-th moment depending explicitly on the $(n+1)$-th). In practice, this infinite hierarchy needs to be truncated after the first few moments, through the so-called closure relation – a relation that expresses the evolution of the highest-order moment considered in terms of the lower-order moments. For a plasma near local thermodynamic equilibrium, such as the *thermal* population of the electrons and ions studied in our work, these closures allow us to describe the plasma self-consistently at larger spatial and temporal scales (relative to the PIC scales), using a finite number of moments. Indeed, we use the first three moments – the mass, momentum, and energy densities of the particles – to describe the dynamics of the *thermal* fluids, and learn their coupling with the kinetic (*non-thermal*) particles and the EM-fields.

A.2 Laser-plasma interactions

The generation of high-energy ion beams from intense laser-solid interactions has been an active area of research due to the potential of producing high-energy, high-charge, high-current ion beams in much more compact systems than solid-state based accelerators (*e.g.* linear accelerators, cyclotrons). The enormous accelerating gradients (TV/m; $\sim$5 orders of magnitude larger than solid-state based accelerators) can be sustained in a plasma (Wilks et al., 2001), accelerating ions to MeV energies within millimeters.

For these laser intensities it is useful to define the normalized vector potential $a_0 \simeq 0.85\sqrt{I[\text{W/cm}^2](\lambda_0[\mu\text{m}])^2/10^{18}}$, where λ_0 is the laser wavelength. The laser electric field can accelerate electrons to relativistic speeds in one cycle if $a_0 > 1$. For ion acceleration, a solid-density target is typically used, for which the laser E-field accelerates the electrons in the small, skin depth, layer near the front surface, producing very energetic electrons. In addition to the acceleration near the front surface, these energetic electrons cross the dense target and escape into the vacuum on the rear side, setting up a strong charge separation E-field that accelerates the ions from the back surface. As a result, the ions will be laminar and possess small divergence angle, and exhibit an energy spectrum typically characterized by an exponentially decreasing distribution (Snavely et al., 2000; Wilks et al., 2001).

A.3 SIMULATION PARAMETERS

In our PIC simulations we consider an intense laser interacting with a planar, solid-density target. In our 1D simulations a laser with frequency ω_0 is launched along the x direction from the left boundary and irradiates an electron-proton plasma (*i.e.* $m_i = m_p = 1836\, m_e$). The laser pulse duration is $\simeq 15$ fs, with intensity $2 \leqslant a_0 \leqslant 20$. The plasma density follows a step-like profile with thickness $\simeq 5\,\mu$m and electron number density 10^{23} cm^{-3} (considering a laser wavelength of 1μm), corresponding to a solid-density target such as a metal foil.

The target is simulated with 1000 particles per cell per species, and the total simulation domain of $\simeq 80\,\mu$m is resolved with a spatial resolution (cell size) of $0.03\, c/\omega_0$. The time step is chosen according to the Courant–Friedrichs–Lewy condition, and the system is evolved over 2000 time steps (or 8500 for $a_0 = 20$). Periodic boundary conditions for both particles and fields are used.

A.4 COMPUTATIONAL CHALLENGES AND OPPORTUNITIES

Studying these laser-plasma interactions using the particle-in-cell (PIC) method requires resolving the fastest and smallest oscillations of the electrons in the plasmas. As a consequence, modeling these interactions with first-principles simulations is very computational demanding. For example, a 3D one-to-one simulation of laser-plasma interactions typically involves evolving the dynamics of billions of (numerical) particles on a billion-cell grid, requiring millions of CPU hours to compute. In this work, we consider only one spatial dimension x, while retaining the three dimensions in momentum. This is identical to simulating a particle distribution uniform in the y and z directions. See Appendix A for details of laser-plasma interactions and the PIC method.

It is important to note that in more realistic 2D and 3D geometries a significant enhancement of speed-up can be achieved, due to the much smaller proportion of the non-thermal particles. This is a result of the finite spot size of the laser, which will only interact with a finite volume of the plasma, and accelerate a small fraction of the particles to non-thermal. In a 3D simulation of these laser-plasma interactions, the non-thermal particles compose typically $< 0.1\%$ of the total number of particles while encompassing $> 50\%$ of the system energy. Compared with the 1D simulations presented in this work ($\simeq 20\%$), one expects a speed-up of > 1000x in 3D.

B DETAILS FOR SEPARATION METHOD

Here we describe the details of the separation method for preparing the labeled (thermal and non-thermal) data. The algorithm separates the particles (from the original simulations) into thermal and non-thermal populations. This is done by computing locally the moments of the particle velocity distribution $f(v)$ and considering non-thermal particles those with velocities v that exceed a threshold αv_{th} – a given multiple (α) of the thermal velocity v_{th} (= the velocity spread σ_v for non-relativistic velocities; Cohen et al. 2010; Fiuza et al. 2011). At each time step, we study the velocity distribution function of the particles in the local neighborhood in space $\mathcal{N}_i$ of each particle $\mathcal{P}_i$. Without knowing which particles belong to the thermal population *a priori*, we begin by identifying a population that is likely thermal at iteration 1. The velocity spread σ_{v_1} computed from this population provides a first estimate of the threshold $\alpha\sigma_v$, allowing us remove particles with $v > \alpha\sigma_v$ from the distribution. The procedure is repeated (on the updated distribution) until a convergence of the value of the threshold is reached at iteration N. The remaining particles now constitute the thermal population, characterized by the fluid velocity $v_{\text{fl}} = \langle v_j \rangle \,\forall\, j \in \mathcal{N}$ and thermal velocity $v_{\text{th}} = \sigma_{v_N}$ ($\langle \cdots \rangle$ denotes the average value). These are the mean and sigma of the Gaussian. The corresponding threshold αv_{th} is then used to determine the population $\mathcal{P}_i$ belongs to. Note that only $\mathcal{P}_i$ has been given a label, the other particles in $\mathcal{N}_i$ were only used to calculate v_{fl} and v_{th}.

Figure 7 illustrates an iteration in detail. From the example velocity distribution function shown in Fig. 7(a), we recognize an absolute peak located at v_{abs} with height h_{abs}. Among the peaks $(v_\mu, h_\mu) \,\forall\, h_\mu \leqslant X h_{\text{abs}}$ for a constant X, pick the one with v_μ closest to zero as the first guess of thermal population and denote the peak location as (v_0, h_0) (Fig. 7b). Search down and outwards from (v_0, h_0) (*i.e.* in both directions of v with $v_+ > v_0 > v_-$) until the heights $h_+ \equiv h(v = v_+) = Y h_0$ and $h_- \equiv h(v = v_-) = Y h_0$ (Fig. 7c). The population enclosed by $\{v | v_+ > v > v_-\}$ is regarded as the population to estimate the threshold. Transform all the particles in $\mathcal{N}$ into the local

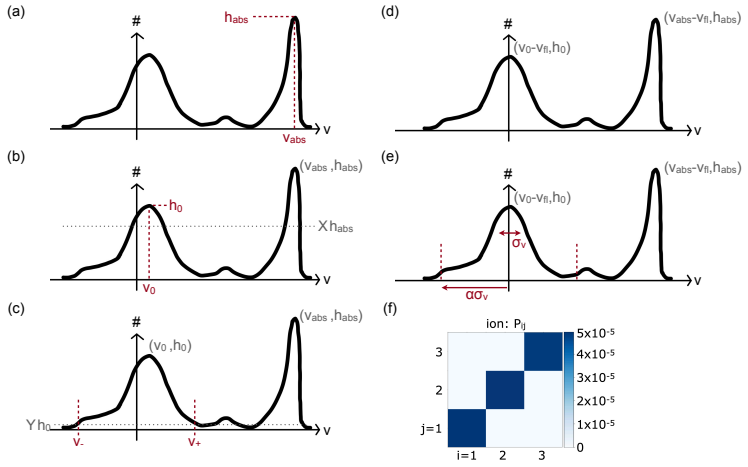

Figure 7: The separation method identifies the thermal and non-thermal populations from the local velocity distribution of particles.

fluid frame according to v_{fl} (Fig. 7d; for non-relativistic velocities this amounts to $v_i \to v_i - v_{\text{fl}}$). Calculate σ_v to obtain the threshold (Fig. 7e).

The size of $\mathcal{N}_i$, values of X, Y, and α are tunable parameters determined by the physics and fine-tuned empirically. We have found that $\mathcal{N}_i \simeq 60$ cells, $X = 0.8$, $Y = 0.1$, and $\alpha = 3$ and 5 for the water and laser-plasma systems lead to the most physically accurate separation.

The accuracy of the algorithm is evaluated by the pressure tensor P_{ij} of the thermal population, where $P_{ij} = \langle v_i v_j \rangle - \langle v_i \rangle \langle v_j \rangle$ characterizes the second-order velocity moments and a is directly related to how well the distribution is described by a Gaussian. Figure 7(f) exemplifies that indeed, for the identified thermal population of the ions for the laser-plasma system, the diagonal terms are all comparable and the off-diagonal terms are negligibly small. Namely, the velocity distribution can be described by a Gaussian.

C Full Pipeline

C.1 Component details

C.1.1 Components (a) and (b)

For both M or M_{NT}, there's 20 moment directional components that needs to be predicted - 10 for ions, 10 for electrons - of the zero (1), first (3), and second (6) moment orders. Thus, for both components, we instantiate 20 models with the same hyperparameter configuration in f_M or $f_{M_{\text{NT}}}$. We also train Component (1) with the push-forward trick and multi-step loss, shown in Brandstetter et al. (2022) to greatly enhance capability of generalization.

C.1.2 Component (c)

In order to parameterize the distribution to sample, a three-dimensional Gaussian is taken to have mean $\mathbb{E}[u] \approx$ the first order moments in $\hat{M}_{\text{NT}}^{t+1}$, and covariance matrix $\mathbb{E}[(u - \mathbb{E}[u])(u - \mathbb{E}[u])^T] \in \mathbb{R}^{3\times3}$ informed by the second order moments in $\hat{M}_{\text{NT}}^{t+1}$. To be able to sample, we require a valid (positive definite) covariance matrix. In order to enforce this condition, we obtain the ground-truth L using the Cholesky decomposition $\mathbb{E}[(u - \mathbb{E}[u])(u - \mathbb{E}[u])^T] = LL^T$ and train $f_{M_{\text{NT}}}$ to predict $\hat{L}$. During rollout, we approximate $\mathbb{E}[(u - \mathbb{E}[u])(u - \mathbb{E}[u])^T] \approx \hat{L}\hat{L}^T$, and use the valid covariance matrix to sample a set of particles which is then "injected". The cumulative set of injected particles constitute the PIC loop.

C.1.3 COMPONENT (D)

After obtaining the sample of injected particles, we subtract their moments from $\hat{M}^{t+1}$ to obtain the thermal population's moments $\hat{M}'^{t+1}$ for the next time step.

C.1.4 COMPONENTS (E) (F) (G)

Each of these components by default use the respective components from the imitation solver. The imitation solver implements the equations used in the OSIRIS PIC simulation, and is tested to match the simulation to machine precision error on any single timestep. These components run efficiently because we now only advance the much smaller sub-population of non-thermal particles $(x, \mathbf{u})^{t+1}$. Figure 8 illustrates the full pipeline of our method, concretized into the laser-plasma simulation.

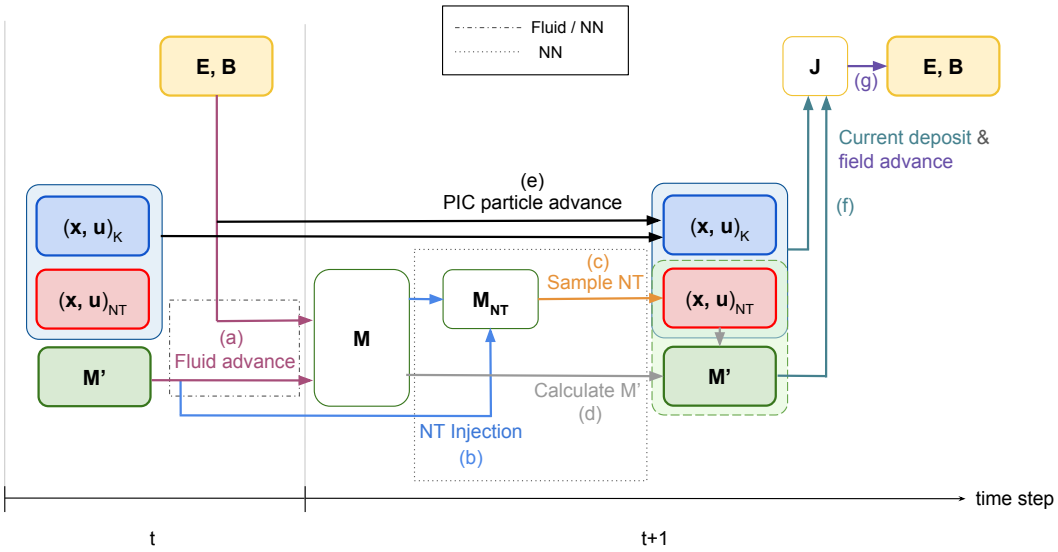

Figure 8: Schematic of LHPC for the multi-scale laser-plasma interaction, which is the application of the general pipeline (Fig. 1) to the current problem. The components correspond almost exactly to Eqs. 1a - 1g except current deposit and field advance are separated.

D ADDITIONAL RESULTS FOR MULTI-SCALE LASER-PLASMA INTERACTION

Here we provide additional results for our experiments, as shown in Tables 2 and 3. Table 2 uses the same dataset split as Table 1, but focus on rollout starting from $t = 1200$, where there is most injection and dynamical activity. From the table, we see that the conclusion is similar as Table 1 in main text: LHPC achieves significant error reduction for key quantities of interest: EM field and MPIC (energetic particles).

To ablate different aspect of our model, we perform an additional experiment with a different train: test split, where the test dataset is in the *future* of the training (Table 3). We take the 10^{th} trajectory with the highest laser intensity that consists of $T = 8480$ time steps, where the training time-range is [1000, 1900] and the testing (rollout) time-range is 1900 onwards. We explore an ablation study on various strategies to use the additional data [1800, 1900]. Namely, we compare finetuning with simply retraining the model on [1000, 1900]. We also use the multi-step loss with the push-forward trick, reported by Brandstetter et al. (2022) and other works to enhance the generalization of the model, and because it can accurately mimic the input distribution during rollout.

E ADDITIONAL DETAILS FOR ARCHITECTURE AND TRAINING

We use a convolutional network (CNN) as the base architecture for modeling fluid network $g_{\text{fluid},\theta}$ (Fig. 9) and injection network $g_{\text{inject},\varphi}$. For both networks, we use the same 6 layers, with the same

Method	Component	Error @ step 1	Error @ step 10	Error @ step 20	Error @ step 50	Speed (s/step)
GT Solver (full PIC)	–	–	–	–	–	9.21E-01
FNO: All-fluid	Field	4.02E-02	3.45E-01	6.45E-01	1.28E+00	8.45E-02
	M_{PIC}	–	–	–	–	
	M	8.91E-03	6.64E-02	9.58E-02	4.20E-01	
FNO: Bi-Gaussian	Field	3.17E-02	2.49E-01	4.00E-01	3.80E-01	1.69E-01
	M_{PIC}	3.16E-02	1.31E-01	2.17E-01	**4.40E-01**	
	M	9.12E-03	5.85E-02	9.88E-02	2.98E-01	
Baseline: All-fluid	Field	7.39E-03	4.27E-02	9.05E-02	2.71E-01	**4.88E-02**
	M_{PIC}	–	–	–	–	
	M	5.23E-03	**3.16E-02**	**6.62E-02**	**1.46E-01**	
Baseline: Bi-Gaussian	Field	6.62E-03	4.63E-02	1.09E-01	2.98E-01	1.27E-01
	M_{PIC}	1.53E-02	4.95E-02	8.66E-02	4.52E-01	
	M	**5.22E-03**	3.50E-02	7.18E-02	1.62E-01	
LHPC (no-coupling)	Field	**1.10E-03**	9.27E-03	1.39E-02	**7.32E-02**	1.05E-01
	M_{PIC}	**1.02E-02**	4.16E-02	7.58E-02	1.92E-01	
	M	6.22E-03	7.05E-02	1.48E-02	3.80E-01	
LHPC	Field	**1.10E-03**	**8.08E-03**	**1.07E-02**	7.33E-02	1.15E-01
	M_{PIC}	**1.02E-02**	**1.89E-02**	**3.70E-02**	6.62E-01	
	M	6.22E-03	4.31E-02	9.65E-02	2.86E-01	

Table 2: Results for laser-plasma interactions for time range $t = 1200$–1250. Trajectory 5 ($a_0 = 10$) is held out for testing, and the model is trained on the other 9 datasets. We report performance on Dataset 5 and rollout at $t = 1200$, where there is the most injection and dynamical activity. As shown, the model remains stable within this chaotic, unseen time range, showing generalization to an unseen trajectory.

Method	Trained on	Finetuned on	Finetuning multi-step loss	Rollout L2 step 1	Rollout L2 step 5	Rollout L2 step 10	Rollout L2 step 20	Rollout L2 step 30	Speed (s/step)
Ground-truth Solver (full PIC)	N/A	N/A	N/A	(N/A, N/A)	(N/A, N/A)	(N/A, N/A)	(N/A, N/A)	(N/A, N/A)	1.71E+00
Baseline: All-fluid	1000-1800	N/A	N/A	(9.19E-02, 6.22E-03)	(3.81E-01, 2.07E-02)	(8.49E-01, 4.48E-02)	(2.16E-01, 9.31E-02)	(3.94E-01, 1.46E-01)	2.31E-02
	1000-1900	N/A	N/A	(2.68E-02, 5.10E-03)	(6.53E-02, 1.87E-02)	(1.08E-01, 4.01E-02)	(2.27E-01, 7.83E-02)	(3.80E-01, 1.46E-01)	2.58E-02
	1000-1800	1800-1900	1	(1.73E-02, 5.13E-03)	(5.17E-02, 2.22E-02)	(8.99E-02, 4.28E-02)	(5.18E-01, 1.48E-01)	(5.28E-01, 7.45E-01)	2.20E-02
	1000-1800	1800-1900	1:1 2: 0.5 3:0.1	(1.68E-02, 4.88E-03)	(4.79E-02, 2.13E-02)	(7.66E-02, 3.99E-02)	(1.72E-01, 8.21E-02)	(2.53E-01, 1.25E-01)	**2.13E-02**
LHPC	1000-1800	N/A	N/A	(3.78E-03, 4.44E-03)	(**1.01E-02**, 2.74E-02)	(1.80E-02, 3.84E-02)	(5.98E-02, 9.48E-02)	(1.21E-01, 2.06E-01)	1.80E-01
	1000-1900	N/A	N/A	(**3.73E-03**, **4.31E-03**)	(1.09E-02, 1.99E-02)	(2.12E-02, 3.96E-02)	(4.41E-02, 9.81E-02)	(8.42E-02, 2.40E-01)	1.76E-01
	1000-1800	1800-1900	1	(3.78E-03, 5.34E-03)	(1.10E-02, **1.47E-02**)	(**1.88E-02**, **3.41E-02**)	(6.92E-02, 1.02E-01)	(2.08E-01, 2.30E-01)	1.78E-01
	1000-1800	1800-1900	1:1 2:0.5 3:0.1 4:0.1	(4.32E-03, 6.46E-03)	(2.57E-02, 2.07E-02)	(4.02E-02, 3.60E-02)	(**4.15E-02**, **7.11E-02**)	(**7.10E-02**, **1.19E-01**)	1.64E-01

Table 3: Ablation study for multi-physics laser-plasma interactions for the 10$^{\text{th}}$ dataset that has highest laser intensity ($a_0 = 20$). We compare amongst: (a) train only single-step model from [1000, 1800] vs. (b) from [1000, 1900] vs. (c) taking (a) and finetuning with single-step loss on [1800, 1900] vs (d) taking (a) and finetuning with multi-step push-forward trick loss on [1800, 1900]. We compare with the All-fluid baseline, and report result from the best hyperparameter configuration. Each cell reports the relative L2 error for the EM-field $[E^t, B^t]$, for M^t.

kernel sizes of 1,1,3,7,3,1, with feature sizes of 64, allowing feature extraction and local information exchange among neighboring cells. For each feature of the moments, we use a different feature head as shown in Fig. 9.

Training. We use Adam (Kingma & Ba, 2014) optimizer, with starting learning rate of 10^{-3}. The training consists of two stages. In the first stage, we train both $g_{M,\theta}$ and $g_{M_{NT},\varphi}$ with single-step loss:

$$L_1 = \mathbb{E}_t \left[\ell(\hat{M}^{t+1}, M^{t+1}) + \ell(\hat{M}^{t+1}_{\text{NT}}, M^{t+1}_{\text{NT}}) \right] \tag{2}$$

In the second stage, we fine-tune with predicting $N = 4$ steps into the future with push-forward trick (Brandstetter et al. 2022) that stops the gradient on the input. The loss is given by:

$$L_2 = \mathbb{E}_t \left[\sum_{i=1}^{N} \alpha_i \ell(\hat{M}^{t+i}, M^{t+i}) + \sum_{i=1}^{N} \alpha_i \ell(\hat{M}^{t+i}_{\text{NT}}, M^{t+i}_{\text{NT}}) \right] \tag{3}$$

For both losses, we have

$$\hat{M}^{t+i} = g_{M,\theta}\left(\text{sg}((\hat{E},\hat{B})^{t+i-1}), \text{sg}(\hat{M'}^{t+i-1})\right) \ , i = 1, 2, ...N \tag{4}$$

$$\hat{M}_{\text{NT}}^{t+i} = g_{M_{\text{NT}},\varphi}\left(\text{sg}(\hat{M}^{t+i-1}), \text{sg}(\hat{M'}^{t+i-1})\right) \ , i = 1, 2, ...N \tag{5}$$

Notice the hat notation $\hat{\cdot}$ in the input arguments, denoting that they are autoregressive prediction of the LHPC at the previous time step, but the gradient is stopped (the "sg" notation). Essentially, we rollout the pipeline of LHPC for multiple steps, and use it to provide a realistic input that contains the rollout error, requiring the model to not only predict well, but able to adapt to noise due to the rollout error. We find that this significantly improve the long-term prediction performance. For the coefficient of the multi-step loss in Eq. 4, we set $(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, 0.5, 0.1, 0.1)$ with decreasing weight for longer time steps. This put more emphasis on the single-step loss to make the training more stable, and also have weight on longer-term future to improve long-term prediction.

For the loss function $\ell(\cdot, \cdot)$ in Eq. 3 and 4, we use

$$\ell(\hat{y}, y) = |\hat{y} - y|^{1.5} \tag{6}$$

We find that this achieves a better performance than the alternative of MSE loss (with exponent of 2) and MAE loss (with exponent of 1). MSE will give very small gradient if the loss is small, and not able to encourage that the prediction to be exactly 0 in the vacuum. On the other hand, MAE is harder to train and do not penalize more for larger errors. Our choice of loss function $\ell(\cdot, \cdot)$ strikes a good balance between the two and enjoys the benefit of both loss functions.

For single-step and multi-step training, we both train 500 epochs, with and cosine learning rate scheduling (Loshchilov & Hutter, 2016).

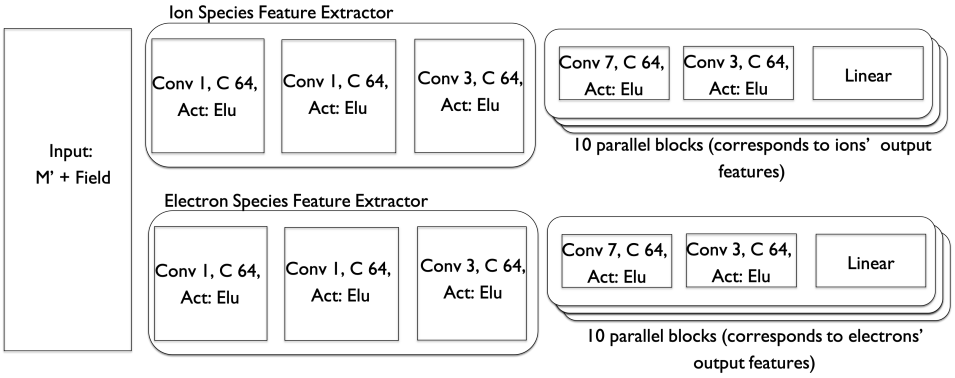

Figure 9: Schematic of the architecture of $g_{\text{fluid},\theta}$.