Previous formulations directly minimized $\bigtriangledown\phi$ without any objective for ϕ directly. This can cause learned functions to depend on the random initialization of network weights or bias in sampling of batches. Many useful orbital invariants define fundamental parameters of a trajectory and as such can be used to update the state of that trajectory. Thus we used the formulation shown in figure 1 to drive the latent feature space of the shared network to infer parameters necessary for prediction. This formulation ensures that learned embeddings for x_t encode sufficient information to predict the following state of the orbital trajectory.

Early results yield strong prediction accuracy although learned functions ϕ are similar to previous formulations directly predicting $\phi(x_t)$ as seen in figure 2. Overall the new network was prone to over fitting and many hyper parameters resulted in a divergence of the network or a trivial $\phi(x)$ during training as the prediction error dominated the loss function, in the future, pre-training on the prediction network and then introducing a loss function over ϕ may produce more consistent results with less need for hyper-parameter tuning. We are continuing to experiment by reducing the size of the latent encoding of orbital parameters to the number of degrees in a 2D orbital trajectory motivated by the concept of the embedding becoming an auto-encoder for the trajectory.

Additionally, we began evaluating the learned invariant of trajectories in terms of know invariants including the Hamiltonian and the angular momentum of the orbital bodies. Currently the fit of these functions is poor, notably the central orbits cause large spikes and a linear model of known invariants shows poor accuracy. We are now looking into adding non-linearities into the known invariants and conducting a hyper parameter sweep to determine if any such learned deep invariants can be well represented as a function of known invariants.

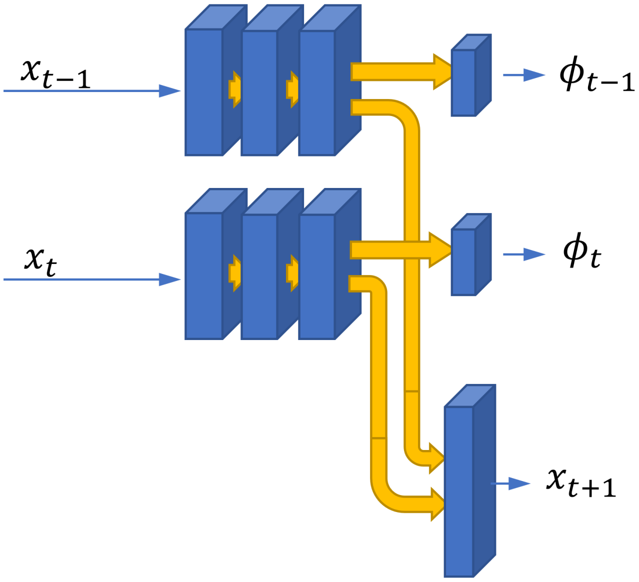
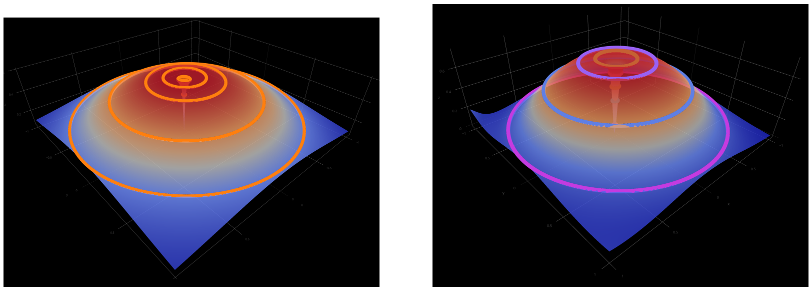


Figure 1: Using prediction to ground features of discovered invariants



(a) No prediction loss

(b) With prediction loss

Figure 2: Early training results - there seems to be more edge effects when incorporating prediction loss, this may be mitigated by lowering the dimension of the final layer of the shared network