CONTROLLING PINN OPTIMIZATION DYNAMICS WITH REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training a physics-informed neural network (PINN) requires decisions about loss weights, optimizer settings, and the allocation of updates as the solution evolves. We study reinforcement learning (RL) as a controller for these decisions in a holographic inverse problem: reconstructing a scalar potential jointly with bulk fields constrained by gravitational equations and reference data. We formulate the control loop and analyze eight experience-collection schedules for a localized model-based controller, using 20 completed replicates and a non-localized reference. The three strongest individual final-controller evaluations have potential root mean square errors of 0.761, 0.770, and 1.14. Schedule means and medians reveal substantial variation, while saved cycle evaluations show that additional training can reverse earlier gains. Examining the recovered wells and their rising branches further distinguishes a low scalar loss from accurate potential reconstruction. These results identify promising individual trajectories and motivate controller evaluation that accounts for physical geometry, replicate stability, and the distribution of experience across training stages.

1 INTRODUCTION

Physics-informed neural networks couple neural approximation with differential equations and observational constraints (Raissi et al., 2019). Their training, however, involves more than minimizing a single homogeneous prediction error. Residuals from different equations can compete, boundary information must propagate through the domain, and the relative difficulty of these tasks changes as the approximation improves. Consequently, the sequence of optimization decisions can be as consequential as the initial choice of hyperparameters.

We investigate how a learning algorithm can control that sequence. An RL controller observes the PINN's residuals and training context, chooses optimization settings, and receives feedback after the resulting updates. This makes it possible to condition a decision on the current approximation and on the work remaining in a training stage. The central question is how to organize this interaction so that it improves the reconstructed physical quantity, rather than merely producing a favorable internal training signal.

The holographic inverse problem provides a demanding setting for this question. A single scalar potential must be compatible with many bulk solutions associated with different thermodynamic states. Features of the potential, including extrema and changes of curvature, encode physical information that a global residual can obscure. The gravitational-duals studies of Bea et al. (2024) and Jimenez et al. (2026) demonstrate why potential reconstruction should be assessed through its geometry and through the physical observables it generates. They also show that localization, sampling, and staged training can substantially affect the inverse solution. These observations motivate studying the training trajectory itself as an object of control.

Our contribution has three parts. First, we describe a sequential control formulation that connects residual observations to optimization actions and cycle-based experience collection. Second, we analyze eight schedules for localized FLEXplore, varying the distribution of memories, exploration, and the length of training stages. Third, we combine aggregate and replicate results with saved cycle potentials to examine when numerical progress corresponds to recovery of the target geometry. This analysis makes the relationship between experience allocation, training stability, and physical reconstruction explicit.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2 Related Work

PINN optimization and adaptive training. The original PINN framework supports both forward solution and inverse identification by combining data with differential-equation residuals (Raissi et al., 2019). Subsequent work has shown that this composite objective introduces distinctive optimization difficulties. Wang et al. (2020) identify unbalanced back-propagated gradients and use gradient statistics to adapt the balance between loss components. Krishnapriyan et al. (2021) show that failure can arise from a difficult optimization landscape even when the network has adequate expressive capacity; their curriculum and sequential formulations change how the physical constraints are learned. Thus, improving a PINN may require changing the path through the objective, not only enlarging the network.

Adaptive weighting and sampling address complementary aspects of this problem. Self-adaptive PINNs learn weights at individual training points, concentrating effort in regions that remain difficult (McClenny and Braga-Neto, 2023). Causal training introduces a physically ordered weighting of temporal residuals (Wang et al., 2022). These methods provide mechanisms for deciding which errors should dominate an update. Our controller instead coordinates several optimization settings over successive blocks of updates. Adaptive weights, sampling rules, and physically motivated curricula are therefore relevant baselines and possible components of the controller's action space; their success does not establish that RL is preferable.

Architectural locality is a complementary response. Chawla and Jain (n.d.) introduce LPINNs, which multiply the first hidden representation of an otherwise unchanged dense PINN by input-dependent localization functions, so that first-layer units acquire receptive fields without partitioning the domain or adding interface losses. Their screen also reports that the benefit depends on the chosen family of localization functions rather than following from locality alone. The gated localization used by the solver studied here originates from that work; Section 3.3 gives the form we use and how it is applied to the holographic inverse problem.

Learning optimization decisions. Learning an optimizer and controlling an optimizer are closely related approaches. Bihlo (2024) meta-learn update rules for PINNs and study transfer between differential equations. Dynamic algorithm configuration formalizes the online adjustment of algorithm parameters as a contextual decision problem (Biedenkapp et al., 2020). This supplies a direct motivation for RL: an action can influence both immediate progress and the states encountered by later decisions. Value-based control and parameterized action spaces provide useful algorithmic foundations (Mnih et al., 2015; Hausknecht and Stone, 2015). In scientific machine learning, HJBPPO uses a PINN constraint to improve an RL value function (Mukherjee and Liu, 2023); here the direction of control is reversed, with RL selecting how a separate physics-informed solver is trained. The distinctive focus of our study is the interaction between this control loop and the schedule by which the controller acquires experience.

Holographic inverse problems and potential geometry. Bea et al. (2024) reconstruct a gravitational scalar potential from an equation of state by jointly learning the potential and a bundle of Einstein–Klein–Gordon solutions. The thermodynamic data supply horizon boundary conditions, while the equations couple those conditions to the bulk fields. A potential value affects more than one thermal state, so learning it is not a collection of independent pointwise fits. Their Gaussian localization of the potential network helps spatially restricted changes coexist within a globally coupled problem. Their evaluation also goes beyond training loss: recovered potentials, bulk solutions, and equations of state are compared with their references, with multiple realizations shown rather than only the selected run.

Jimenez et al. (2026) extend this approach to large hierarchies and false vacua. An affine coordinate along the thermodynamic curve separates states that are nearly degenerate in entropy and temperature and permits more uniform sampling along that curve. They apply localization to this coordinate in the solution networks and use physical constraints inferred from ultraviolet and infrared thermodynamics. For false vacua, a two-branch procedure first reconstructs one part of the potential, freezes it, and trains the second part with a smooth matching construction. The resulting problem includes a region without corresponding thermal states in the supplied equation of state: reconstruction there depends on the equations and information from the adjoining regions, rather than on direct thermodynamic coverage.

These studies sharpen the interpretation of our experiments. A controller should be assessed by whether it locates the well, recovers its depth, and resolves the rising branch, as well as by whether it lowers a reported loss. They also suggest specific experiments on state representation and staged constraints. Our localization and data setup differ from theirs, as detailed below; their physical reconstruction results are methodological guidance, not validation of the RL controller evaluated here.

3 METHOD

3.1 CONTROLLING THE PINN TRAINING TRAJECTORY

Let θ_t contain the PINN and potential-network parameters after decision t. An action a_t specifies optimization settings for a block of inner updates,

$$\theta_{t+1} = \mathcal{U}(\theta_t, a_t; \mathcal{D}), \qquad a_t \sim \pi_\psi(\cdot \mid o_t), \tag{1}$$

where $\mathcal{D}$ comprises the collocation and reference data. Here $\pi_\psi(\cdot \mid o_t)$ is the learned controller policy: it maps the current observation to a distribution over optimization actions, and ψ denotes its trainable parameters. At each decision, the selected action is decoded into optimizer settings and applied for a fixed block of PINN updates. The complete training state also includes optimizer moments and sampling state. We use residual observations and stage context as a compact observation o_t; they need not constitute an exactly Markov state, so the policy acts on a partial description of the optimization process.

The environment returns seven residual channels evaluated on a grid. In the schedule experiments, four additional channels encode the cycle fraction, the start and end of the current training window, and the fraction of decisions remaining. Thus o_t has seven residual features and four stage features, allowing the same residual pattern to be interpreted differently at different stages. A cycle consists of collecting transitions, fitting the controller to the resulting replay data, and saving its weights for evaluation. The memory schedule determines the amount of experience available in each cycle; it is specified externally and is distinct from the optimization actions learned within a cycle.

The collection reward is a discounted log-loss improvement with a penalty for reducing residual scales:

$$r_t = 0.99^{t-1}\left[50\log_{10}\left(\frac{\ell_t + 10^{-10}}{\ell_{t+1} + 10^{-10}}\right) - 800\sum_{j=1}^{7}(1 - s_{t,j})\right], \tag{2}$$

where t counts decisions within an episode, $s_{t,j} \in [0, 1]$ scales residual channel j, and ℓ_t is the collection loss diagnostic. This penalty discourages suppressing equations to improve the numerical objective. It does not directly reward extrema or potential RMSE. The dependence of the recorded loss on residual scaling is discussed in Section 4.3; the reward is an optimization signal, not a physical accuracy certificate.

3.2 CONTROLLER FAMILIES AND ACTION REPRESENTATION

We use a shared environment interface for four controller families: a multi-head Q-network (`default`), a parameterized-action critic (`mpdqn`), a sequential action decoder (`seq-madac`), and a learned dynamics-and-reward planner (`flexplore`). The first two correspond to value-based control and parameterized action spaces, while the latter two are implementation variants for factorized decoding and short-horizon planning; these choices follow standard RL and dynamic-configuration formulations (Mnih et al., 2015; Hausknecht and Stone, 2015; Biedenkapp et al., 2020). The interface exposes optimizer choice, learning rate, Adam moment parameters, seven residual scales, and an inner-update budget. The parameterized implementations encode the learning rate, two moment parameters, and seven scales in a normalized ten-dimensional vector, with a separate discrete optimizer index.

The schedule experiments use FLEXplore. Its learned transition model predicts how observations change, while its reward model scores candidate short-horizon action sequences. During experience collection, the schedule path decodes relative changes to the learning rate, Adam moment parameters, and residual scales, and uses Adam for the executed updates. Thus, the broader interface's

Adam/LBFGS option is not an optimizer-switching ablation in these results. Each collection decision performs 50 PINN epochs. The controller is trained for up to 100 epochs per cycle with patience 10; a controller epoch is distinct from a PINN epoch or a collected transition. The other families define the shared comparison interface, while the quantitative analysis below concerns FLEXplore schedules.

3.3 Inverse model and localization

The inverse solver jointly represents six bulk or auxiliary fields with separate networks and learns the scalar potential $V(\phi)$ with another network. Static homogeneous gravitational equations reduce to an ordinary differential system in the radial coordinate $u \in [0, 1]$. At each collocation point, three residual channels enforce auxiliary derivative definitions and four enforce the remaining gravitational equations and constraint. The training objective has the form

$$\mathcal{L}(\theta; s) = \frac{1}{7N} \sum_{j=1}^{7} \sum_{i=1}^{N} \big(s_j E_j(\theta; u_i, b_i)\big)^2 + \lambda(q)\mathcal{L}_{\text{aux}}(\theta), \tag{3}$$

where b_i identifies the thermodynamic boundary condition associated with point i. The scales multiply the residuals before squaring, so the controller can change their relative contribution while the seven channels remain explicit. Boundary conditions on the solution fields are imposed through constrained output parameterizations.

To introduce locality without partitioning the domain, the localized networks multiply first-layer activations by Gaussian gates,

$$\widetilde{h}_j(x) = h_j(x) \exp\left[-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right]. \tag{4}$$

In our implementation, the gate coordinate is ϕ for the potential and u for the solution networks; centers and bounded widths are trainable. "Both/Gaussian" means that both network groups use these gates. The first-layer gated construction of Eq. 4 is due to Chawla and Jain (n.d.), who propose it for standard PINNs and screen localization-function families on forward problems. We take the Gaussian form from that work and apply it to both network groups in this inverse problem; it is not the affine-coordinate localization of Jimenez et al. (2026) or an exact reproduction of either gravitational-duals architecture. The shared motivation is to make local changes to the approximation easier without requiring every region to change together.

4 Experimental Setup

4.1 Data, computation, and evaluation

The experiments use the retained thermodynamic curve. The solver takes rows from index 45 onward and selects every seventh resulting boundary-condition pair. It also uses the supplied reference profiles of A, Σ, and ϕ through $\mathcal{L}_{\text{aux}}$, whose coefficient decays as $\lambda(q) = \exp(-0.001q)$ with the auxiliary-loss evaluation count q. These bulk profiles constitute additional supervision. The experiments therefore study optimization of this combined inverse setup, rather than reconstruction from thermodynamics alone.

All retained schedule runs use eight collection workers. The main results use 100,000-epoch greedy PINN rollouts driven by saved controllers (2,000 decisions of 50 epochs each). Here "inference" refers to applying a trained controller to optimize a fresh PINN; it still involves PINN updates. It is not a fixed network's forward pass. The evaluation path uses the unconditional action decoder with fixed stage context, whereas collection uses relative actions and changing stage context. This train–evaluation difference is part of the retained setup.

For Schedules 1–4 and 6, the reported endpoint is the final cycle's saved evaluation. Schedules 5, 7, and 8 have separately saved standard final-controller rollouts; their cycle-checkpoint evaluations are separate observations. We preserve this distinction when displaying cycle progress and final results. The eight schedule definitions and their memory targets and windows are summarized in Table 1. A later rerendering of the same controller need not reproduce the original standard rollout because it starts a new PINN optimization trajectory.

4.2 MEMORY SCHEDULES AND EXPLORATION

Table 1 specifies all eight localized schedules. A memory is a stored RL transition, not a PINN epoch; increasing a target chiefly changes the number of sampled trajectories within a training stage. The ordinary collection episode contains 200 decisions, corresponding to a 10,000-epoch window. Episodes restart from the cycle's starting state, while successive cycles advance that starting state. Schedule 5 uses 400 decisions and a 20,000-epoch window. The collection window and memory target thus control different aspects of experience.

Table 1: Localized FLEXplore schedule definitions. Memory targets and reset windows are in thousands (K), with targets listed in cycle order. All schedules use Both/Gaussian localization; [†]Schedule 5 has an initial exploratory phase followed by a greedy collection.

Schedule	Memory targets [K]	Cycles	Exploration ϵ	Window [K]
1	10/20/20/30/30	5	0.2	10
2	20/25/25/35/35	5	0.2	10
3	10/20/30/40/50	5	0.2	10
4	10/20/20/30/30/30	6	0.2; last 0.5	10
5	20/40/40/60/60	5	0.2; mixed collection[†]	20
6	100/200/200/300/300	5	0.2	10
7	5/10/10/20/30	5	0.2	10
8	30/30/30	3	1.0/0.2/0.2	10

Schedule 1 establishes the five-cycle allocation. Schedule 2 increases early experience, whereas Schedule 3 increases the amount collected in later cycles. Schedule 4 adds a sixth cycle and raises exploration to 0.5 in that cycle. Schedule 5 doubles Schedule 1's memory targets and window length: its first cycle is exploratory, and subsequent cycles collect a greedy block followed by a 20K block at $\epsilon = 0.2$. Schedule 6 increases Schedule 1's targets by an order of magnitude while retaining the 10K window. These alternatives examine whether additional experience is useful at different stages, but they do not all use equal compute.

Schedule 7 reduces early targets to 5K/10K/10K before increasing them to 20K/30K. Schedule 8 concentrates experience into three 30K cycles, begins with fully random exploration, and leaves residual scales free in every cycle. The other configurations normally replace scale actions by ones in their planned final cycle. The nominal scale baselines progress through 0.1/0.3/0.5/0.7/0.9, with later cycles using 0.9 before any final-cycle override. Schedule 1's first replicate originally planned six cycles but stopped after five; its effective label therefore does not imply an identical final-cycle constraint history to the other five-cycle runs.

Collection uses a single exploration phase except in Schedule 5. Cycle stopping is enabled for Schedules 1–3 and 6 and disabled for Schedules 4, 5, 7, and 8. Completed evidence comprises three replicates each for Schedules 1–4, 7, and 8, one each for Schedules 5 and 6, and one non-localized FLEXplore reference using 10K/20K/20K/30K/30K. These are comparisons of schedule configurations; changes in exploration, stopping, and scale constraints prevent attributing every difference to memory count alone.

4.3 METRICS AND CHECKPOINT INTERPRETATION

The primary numerical diagnostic is absolute potential RMSE on the 100 stored, evenly spaced points over $\phi \in [0, 1.8]$:

$$E_V = \left[\frac{1}{100} \sum_{i=1}^{100} \left(V_{\text{best}}(\phi_i) - V_{\text{ref}}(\phi_i) \right)^2 \right]^{1/2}. \tag{5}$$

The subscript "best" refers to the solver-selected state saved in the report, selected by its tracked loss, rather than the epoch with minimum potential RMSE. We report the final controller's evaluation for each replicate, not the minimum over controller cycles. The three highlighted runs are selected from these 20 endpoints; their sampled minima, depths, and endpoint rises are reported in Table 3. Earlier cycle evaluations can have smaller RMSEs and remain visible in the cycle analysis.

The loss column reproduces the archive's `best_unweighted` field. Its name is not sufficient to establish scale independence: the available evaluation code obtains residuals from equations that already multiply each channel by s_j, without explicitly resetting these scales to one. We therefore call this quantity the *reported best loss*, retain it as an optimization diagnostic, and do not interpret cross-schedule differences as a verified comparison of unweighted physical residuals. The saved potential arrays allow RMSE to be checked independently of that label.

We also examine the sampled minimum $(\phi_{\min}, V_{\min})$ and the endpoint rise $\Delta V = V(1.8) - V_{\min}$. These are descriptive diagnostics computed directly from the stored grid, without interpolation or derivative estimation. They identify displaced or shallow wells and an insufficient rebound, but do not establish the accuracy of all extrema or thermodynamic observables. Absolute RMSE here is also different from the relative RMS errors reported in the gravitational-duals papers.

5 Results

5.1 Aggregate performance and replicate variation

Table 2 gives the complete schedule-level results, with the source report's rounding preserved. Schedule 1 has the lowest mean endpoint RMSE (1.38), followed by Schedules 2 (1.45) and 7 (1.49). Their medians are 1.62, 1.63, and 1.45, respectively. The differing order under mean and median reflects the effect of individual strong runs, and makes a single schedule ranking inadequate.

Table 2: Standard final-controller evaluations. n is the number of replicates. Means and medians aggregate per-replicate metrics, not averaged potential curves. The loss diagnostic is defined in Section 4.3.

Configuration	Memory Targets [K]	n	Best Loss	Mean E_V	Median E_V
Non-localized reference	-	1	6.71×10^{-5}	1.62	1.62
Schedule 1	10/20/20/30/30	3	4.13×10^{-4}	1.38	1.62
Schedule 2	20/25/25/35/35	3	8.79×10^{-5}	1.45	1.63
Schedule 3	10/20/30/40/50	3	6.75×10^{-5}	1.56	1.64
Schedule 4	10/20/20/30/30/30	3	5.43×10^{-5}	1.75	1.73
Schedule 5	20/40/40/60/60	1	2.91×10^{-4}	1.66	1.66
Schedule 6	100/200/200/300/300	1	2.16×10^{-4}	2.96	2.96
Schedule 7	5/10/10/20/30	3	8.12×10^{-5}	1.49	1.45
Schedule 8	30/30/30	3	6.91×10^{-5}	4.50	1.89

The individual results are listed in Supplementary Material. Schedule 1 spans 0.770–1.75, Schedule 2 spans 0.761–1.97, and Schedule 7 spans 1.14–1.88. The single non-localized reference has RMSE 1.62, so several localized evaluations improve on it without establishing a reliable general effect. Schedule 8 is uneven (10.2, 1.44, and 1.89); its mean of 4.50 is dominated by the first replicate, while its median of 1.89 remains above those of Schedules 1, 2, and 7.

Larger targets do not consistently improve reconstruction. Schedule 3 increases later-cycle memory but reaches a mean of 1.56. The six-cycle Schedule 4 has a mean of 1.75 despite the lowest mean reported loss, 5.43×10^{-5}. Schedules 5 and 6 yield 1.66 and 2.96 in their single completed replicates. These observations do not support a monotonic relationship between memory allocation and potential accuracy. They also do not isolate why a configuration succeeds or fails, because some alternatives alter the collection mode and other settings simultaneously.

5.2 Progress across controller cycles

Figure 1 recomputes RMSE from all 97 saved cycle-potential reports. The points are separate controller evaluations, not samples from one continuous PINN optimization trajectory. Potential arrays are available at these checkpoints; epoch-by-epoch numerical residual and loss histories are not available in the same reports, so the figure does not interpolate an unobserved learning curve.

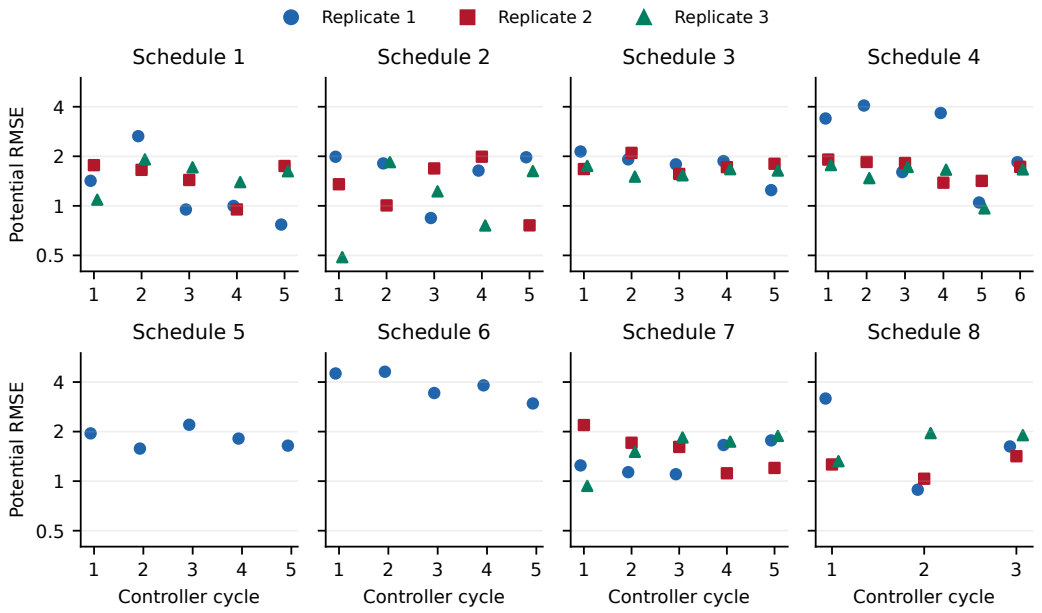

Figure 1: Potential RMSE of saved cycle evaluations, with all available replicates shown. Points are unconnected and slightly offset horizontally for visibility; the vertical axis is logarithmic. Schedules 5 and 6 have one replicate. Separate standard rollouts for Schedules 5, 7, and 8 are reported in Table 2, not substituted for their cycle points.

Progress is non-monotonic. Schedule 1 replicate 1 has cycle RMSEs 1.420, 2.650, 0.950, 0.998, and 0.770: the final result follows an early deterioration and subsequent recovery. Schedule 2 replicate 2 moves through 1.353, 1.006, 1.688, 1.989, and 0.761. Its strongest endpoint is reached only after two intervening cycles with worse reconstructions. This behavior is consistent with a controller whose changing experience alters the optimization trajectory; the observations alone do not identify which actions caused the changes.

The converse occurs in other replicates. Schedule 2 replicate 3 reaches 0.488 in its first saved cycle evaluation but ends at 1.626. The added sixth cycle of Schedule 4 worsens all three cycle-five errors: 1.048 to 1.845, 1.419 to 1.726, and 0.965 to 1.665. A longer schedule therefore does not guarantee preservation of previously useful behavior. The higher exploration probability in that cycle is a possible contributor, but is confounded with continued controller training and evaluation variability.

Schedule 7 replicate 2 improves from 2.191 to 1.117 by cycle four, followed by 1.203 at cycle five; its separately saved standard rollout gives 1.142. The distinction matters even more for Schedule 8 replicate 1: its last cycle evaluation gives 1.625, whereas the original standard rollout gives 10.165. We retain both. This discrepancy demonstrates sensitivity across evaluations of the saved controller and cannot be resolved by replacing the unfavorable endpoint with a later rerendering. The supplementary material shows the evolving potential shapes for the three selected endpoint runs.

5.3 Strong individual runs and recovered potential geometry

The three strongest standard final-controller results are Schedule 2 replicate 2 ($E_V = 0.761$), Schedule 1 replicate 1 (0.770), and Schedule 7 replicate 2 (1.14). Schedule 2 replicate 2 is slightly ahead of Schedule 1 replicate 1 in aggregate grid error. This ordering applies to the selected endpoint evaluations, not to every saved cycle or to the schedules as populations. Figure 2 shows both the recovered potentials and their pointwise errors.

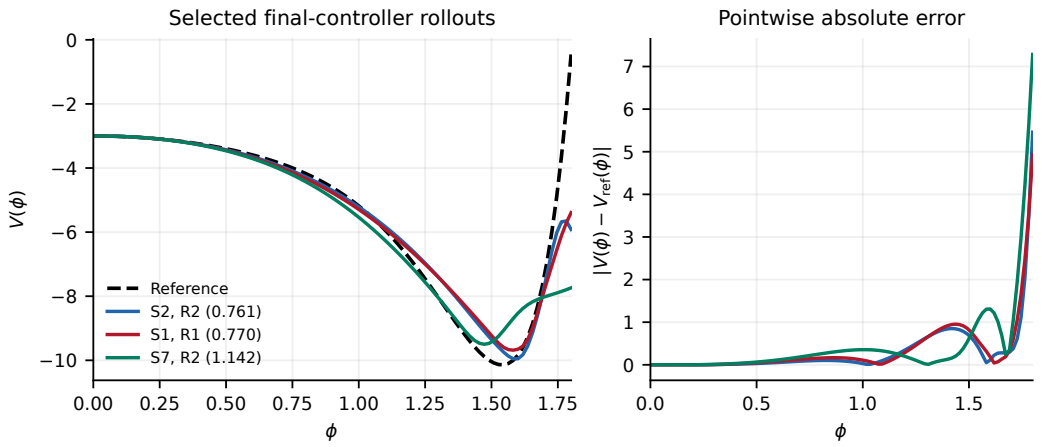

Figure 2: Three strongest individual standard endpoint evaluations, selected from the full replicate table. Left: stored solver-selected potentials against the reference. Right: absolute errors on the same 100-point grid. The legend gives RMSE. Recovery of the main well coexists with appreciable error on the rising branch.

The selected curves recover an interior well and a rebound. Table 3 quantifies what this agreement does and does not capture. The sampled reference minimum is at $\phi = 1.545$ with $V = -10.140$. Schedules 2 and 1 place it at 1.582, while Schedule 7 places it at 1.473. Schedule 2's selected well depth is closer to the reference than Schedule 1's, yet both substantially underestimate the rise from the well to the right endpoint. Their rises are 4.035 and 4.292 against the reference 9.684; Schedule 7's rise is 1.761. A small advantage in global RMSE therefore does not mean that all physically relevant features have improved together.

Table 3: Geometry recomputed from the stored standard-rollout potentials. Memory targets are listed in cycle order; "–" denotes the non-localized reference. Minima are grid minima with spacing approximately 0.0182; $\Delta V = V(1.8) - V_{\min}$ measures the endpoint rise. Values are not interpolated extrema or new experimental runs.

Potential	Memory targets [K]	$\phi_{\min}$	$V_{\min}$	ΔV
Reference	–	1.545	-10.140	9.684
Schedule 2, replicate 2	20/25/25/35/35	1.582	-9.951	4.035
Schedule 1, replicate 1	10/20/20/30/30	1.582	-9.684	4.292
Schedule 7, replicate 2	5/10/10/20/30	1.473	-9.500	1.761

The remaining replicates reinforce the distinction between scalar loss and potential geometry. Schedule 7 replicate 3 has a lower reported loss (2.14×10^{-5}) than replicate 2 (1.62×10^{-4}), but a worse RMSE (1.88 versus 1.14). Schedule 2 replicate 3 likewise has a lower reported loss than replicate 2, but a worse endpoint potential. These are empirical disagreements between the two diagnostics, not evidence that the residual equations are irrelevant. They indicate that the archived loss, checkpoint selection, and physical reconstruction must be examined together.

6 Discussion

What the schedules reveal. The schedule experiments show that useful final controllers can emerge from quite different allocations of early experience. Schedule 2's strongest run follows increased early collection, while Schedule 7's follows reduced early targets. Schedule 1 has the lowest mean endpoint error, whereas Schedule 7 has the lowest median. Together with the reversals across cycles, these results suggest that the composition and stage-wise placement of experience merit as much attention as its volume. Establishing a causal explanation will require matching initial conditions and budgets and relating action histories to subsequent changes in the potential.

The three highlighted runs illustrate partial geometric recovery: their wells lie near the reference minimum, but the rising branch remains insufficiently resolved. As emphasized by the gravitational-duals literature, a recovered function must support relevant physical structure, not merely match one average error. The well-plus-rebound shape is diagnostic of this stored reference; it does not demonstrate recovery of a false vacuum or the phase structure studied by Jimenez et al. (2026).

Data coverage and physical information. Thermodynamic samples constrain the bulk solution through boundary conditions and the gravitational equations, while the auxiliary profiles add bulk information. Neither source directly labels $V(\phi)$ densely, and the retained reports do not record coverage along the potential coordinate. The larger rising-branch error therefore cannot be assigned to a hidden or weakly sampled region from these artifacts alone; importing the thermodynamic-gap explanation of Jimenez et al. (2026) would conflate two settings.

Directions motivated by the gravitational-duals studies. Three extensions follow naturally: an affine thermodynamic coordinate for more uniform branch sampling, ultraviolet and infrared constraints treated as explicit training decisions, and branchwise freezing with smooth matching. These mechanisms have support in Bea et al. (2024); Jimenez et al. (2026) but were not evaluated here; each should first be compared separately before learning when to apply it.

7 Limitations and Future Work

The evidence concerns one inverse setup, with three replicates for six localized schedules and one for each larger-memory schedule and the non-localized reference. The three best results were selected after observing endpoint errors; we do not estimate significance or generalization. Schedule budgets and collection settings differ, and Schedule 1 includes an effective five-cycle run. Matched-seed comparisons against fixed schedules, adaptive weighting, and the other controller families are still needed.

The residual observation omits parts of the optimizer state, and collection and evaluation use different action decoding and context. Scale-dependent loss reporting also limits interpretation of reward and checkpoint selection. Setting all residual scales to one, fixing the collocation grid, and reporting auxiliary loss separately would make physical comparisons more informative; this analysis preserves the archived measurements.

Potential RMSE and sampled well geometry are useful but not a full physical validation. Future work should measure extrema and derivatives on a finer grid, map coverage in ϕ, and solve the direct gravitational problem using the reconstructed potential, following Bea et al. (2024), before extending to larger hierarchies or false-vacuum regimes. These checks could also provide a better stopping criterion than scalar loss alone.

RL adds collection and model-training steps, and an inadequately checked inverse solution can mislead downstream analysis. Reporting all replicates and discrepancies between numerical and physical diagnostics is therefore part of assessing practical value.

8 Conclusion

We formulated PINN training as sequential control and analyzed eight localized FLEXplore schedules. The strongest individual endpoint RMSEs were 0.761, 0.770, and 1.14, with substantial replicate variation and cycle reversals. Recovering the main well did not ensure an accurate rebound, motivating joint evaluation of optimization diagnostics, physical geometry, and repeatability.

Reproducibility Statement. Schedule definitions, retained endpoint tables, evaluation conventions, and supplementary figures are included with this submission. The analyses use archived evaluations and stored 100-point potential arrays. The source code is available on Git repository https://anonymous.4open.science/r/rl-pinn-7E2E/

AI Use Statement. AI assistants supported language editing and formatting checks. The authors designed, ran, and verified the experiments and take responsibility for the manuscript and results.

REFERENCES

M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019. doi:10.1016/j.jcp.2018.10.045.

Y. Bea, R. Jimenez, D. Mateos, S. Liu, P. Protopapas, P. Tarancón-Álvarez, and P. Tejerina-Pérez. Gravitational duals from equations of state. *Journal of High Energy Physics*, 07:087, 2024. doi:10.1007/JHEP07(2024)087.

R. Jimenez, D. Mateos, P. Protopapas, P. Solé-Vilaró, P. Tarancón-Álvarez, and P. Tejerina-Pérez. Gravitational Duals from Equations of State II: Large Hierarchies and False Vacua. arXiv:2606.30117, 2026.

S. Wang, Y. Teng, and P. Perdikaris. Understanding and mitigating gradient pathologies in physics-informed neural networks. arXiv:2001.04536, 2020.

A. S. Krishnapriyan, A. Gholami, S. Zhe, R. M. Kirby, and M. W. Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 2021. arXiv:2109.01050.

L. McClenny and U. Braga-Neto. Self-adaptive physics-informed neural networks. *Journal of Computational Physics*, 474:111722, 2023. doi:10.1016/j.jcp.2022.111722.

S. Wang, S. Sankaran, and P. Perdikaris. Respecting causality is all you need for training physics-informed neural networks. arXiv:2203.07404, 2022.

Lakshay Chawla and Hardik Jain. LPINNs: First-Layer Gated Localization for Physics-Informed Neural Networks. arxiv:2609.22984, 2026.

A. Bihlo. Improving physics-informed neural networks with meta-learned optimization. *Journal of Machine Learning Research*, 25(14):1–26, 2024. https://jmlr.org/papers/v25/23-0356.html.

A. Biedenkapp, F. H. Bozkurt, T. Eimer, F. Hutter, and M. Lindauer. Dynamic Algorithm Configuration: Foundation of a New Meta-Algorithmic Framework. In *Proceedings of ECAI*, pages 427–434, 2020. https://www.ai.uni-hannover.de/fileadmin/ai/paper/Eimer/DAC2020.pdf.

V. Mnih et al. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. doi:10.1038/nature14236.

M. Hausknecht and P. Stone. Deep reinforcement learning in parameterized action space. arXiv:1511.04143, 2015.

A. Mukherjee and J. Liu. Bridging Physics-Informed Neural Networks with Reinforcement Learning: Hamilton-Jacobi-Bellman Proximal Policy Optimization (HJBPPO). arXiv:2302.00237, 2023.