

1 We appreciate the reviewer’s valuable comments, and we were glad to read the positive comments regarding the
2 technical motivation, idea, and our results. We also appreciate the thorough feedback for further improvements. We will
3 address those issues in a future revision of our work.

4 **Review 1:** *What would be a real use case?* We believe our work can be applied to large variety of PDE simulations
5 where the reference can be computed, but is costly to obtain. A particularly interesting application would weather
6 prediction, where a simple differentiable solver could be augmented with a learned correction function to recover the
7 costly predictions of operational forecasting systems.

8 *What is trained in the PRE-approach?* The prior knowledge used for PRE models is usually problem-dependent and
9 makes use of a reduced version of the full PDE formulation. For example, the PRE model for the Navier-Stokes
10 equations makes use of a divergence-free constraint. For details of the *constrained least-squares method* used for this
11 model, we refer to Appendix A.2.

12 *Is there benefit in using the differentiable PDE solver?* It would be interesting to evaluate learned surrogate models that
13 replace the source PDE in our training setup, as neural networks can provide gradients by construction. However, any
14 errors introduced by the surrogate could yield sub-optimal gradient information, in turn deteriorating the quality of the
15 learned correction.

16 *Do steps of a differentiable simulator correspond to time steps?* Yes, in our text “step” typically means time step. We
17 use a normalized $\Delta t = 1$, so in Figure 1, t directly indicates the number of recurrent time steps that were calculated to
18 obtain the result shown. It’s a good idea to add a visual overview of the *recurrent blocks*. Note that for the *CG solver*
19 example, the steps correspond to the iteration of the CG solver.

20 For the “*look-ahead trajectory per iteration*”, the iteration denotes a single step of training. At each iteration, the
21 weights of our model receive gradients from all look-ahead steps of the solver.

22 **Review 2:** *The computational burden can be discussed more.* We would be happy to include measurements for the
23 other cases and discuss them in the main text. For example, the 3D example is particularly interesting but currently only
24 mentioned at the end of the appendix. In this case, the regular reference solver needs ca. 957 seconds, compared to 12.5
25 seconds for a simulation with SOL₁₆.

26 *How were the test datasets created?* We chose offset parameters w.r.t. training data set or shifted distributions for initial
27 conditions. Details of the test parametrizations are given towards the end of each first paragraph of the B.n sections in
28 the appendix.

29 Our *PDE solvers* cover a variety of advection-diffusion problems. The B.n sections of the appendix also give details of
30 the numerical methods we have implemented in a differentiable manner in our TensorFlow framework. We will also
31 revise our text regarding taking solver reactions into account and clarifying novelty w.r.t. previous work.

32 **Review 3:** *In principle, error should depend on the whole trajectory.* While the error certainly accumulates and typically
33 grows over the course of a full trajectory, our key hypothesis here is that a learned approach can nonetheless identify
34 and correct a large part of the error function based on information from a single phase-space input. We do not claim
35 that our method is able to perfectly correct the full error in each step, but our results demonstrate that a very significant
36 portion is learnable. Moreover, our tests with models using history information consistently did not yield significance
37 improvements. We are confident that this topic could be clarified easily in a revision.

38 A theoretical analysis for the highly non-linear cases we are targeting would be a interesting topic for future work, and
39 we hope our work will inspire further research in this direction. As these are the only negative points mentioned in the
40 review, we were surprised about the negative final assessment.

41 **Review 4:** Thank you for pointing out the *inconsistencies between main text and appendix*. We will correct this.

42 *It would be interesting to compare the cost of training versus a reference simulation.* Training a corrector is potentially
43 costly. Training complexity primarily scales with the cost for the differentiable solver and the number of look-ahead
44 steps. The complex SOL models can take more than a day of training time. However, we anticipate that the training cost
45 will in practice quickly amortize as our models generalize well and can be re-used for a large number of new simulation
46 runs. We will add training and reference simulation timings to Table 6 in the appendix.

47 *Why is the correction only defined as a function of the current state s_t ?* For the PDEs we consider, a single state actually
48 uniquely describes its future evolution. We have experimented with additionally providing varying numbers of previous
49 states s_{t-k}, \dots, s_{t-1} as input to our model, but our tests have not shown improvements. The tests indicate that the
50 components of the error function that are learned with our approach can be reliably inferred from a single state s_t . We
51 can include these additional experiments to illustrate that providing additional states has a negligible influence on the
52 learned corrector.