OPERATOR LEARNING FOR ALGORITHMIC EQUIVALENCE: FROM MIRROR FLOWS TO FINITE STEPS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mirror descent and Euclidean gradient methods use different geometries, but in continuous time they can sometimes describe the same dynamics after a nonlinear change of coordinates. We ask how far this equivalence extends when both methods take finite steps of the same size. For a mirror regularizer R, we first construct the continuous coordinate map q_R and study operator learning of the map $R \mapsto q_R$ across regularizer families. At finite step size, requiring one objective-independent nonlinear coordinate map to match the two updates for a neighborhood of gradient directions forces the map to be affine and the mirror geometry to be quadratic. We therefore allow the finite-step coordinate map to depend on the objective. For fixed R, f, and step size η, we solve for $q_{R,f}(\eta, \cdot)$; across problem families, we study operator learning of $(R, f) \mapsto q_{R,f}(\eta, \cdot)$. We then study how the continuous and finite-step coordinate maps are related as η varies. Finally, we ask whether the objective or regularizer can be modified so that the continuous and finite-step formulations produce the same coordinate map. Our numerical experiments include one-dimensional examples with exact solutions and genuinely coupled two-dimensional function families.

1 INTRODUCTION

Mirror descent and Euclidean gradient descent minimize the same objective using different geometries. Euclidean gradient descent uses the ordinary Euclidean geometry. Mirror descent (Nemirovsky & Yudin, 1983; Beck & Teboulle, 2003) instead uses a regularizer R, whose curvature changes how a step is measured (Bregman, 1967). This raises a basic question: can the two algorithms describe the same dynamics after a change of coordinates?

In continuous time, the answer is sometimes yes. A nonlinear coordinate map can transform mirror flow into Euclidean gradient flow when it flattens the geometry induced by R (Amid & Warmuth, 2020; Li et al., 2022). We denote this continuous coordinate map by q_R. Our first goal is to construct q_R for a given regularizer and then learn how this map varies across a family of regularizers. The equivalence condition gives an equation for q_R. For individual regularizers, we solve this equation numerically; across regularizer families, we study operator learning (Kovachki et al., 2023) of

$$R \longmapsto q_R.$$

Finite steps lead to a different question. A coordinate map that makes the two continuous flows agree does not necessarily make a single mirror-descent step match a single Euclidean gradient step (Gunasekar et al., 2021; Ghai et al., 2022). Moreover, a single nonlinear coordinate map that is independent of the objective cannot, in general, match the two finite-step updates for all nearby gradient directions. This motivates allowing the coordinate map itself to depend on the objective. For a regularizer R, objective f, and step size η, we therefore seek a coordinate map

$$q_{R,f}(\eta, \cdot)$$

that makes one mirror-descent step exactly match one Euclidean gradient step with the same η. We solve this equation for fixed problems and study operator learning of

$$(R, f) \longmapsto q_{R,f}(\eta, \cdot)$$

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.

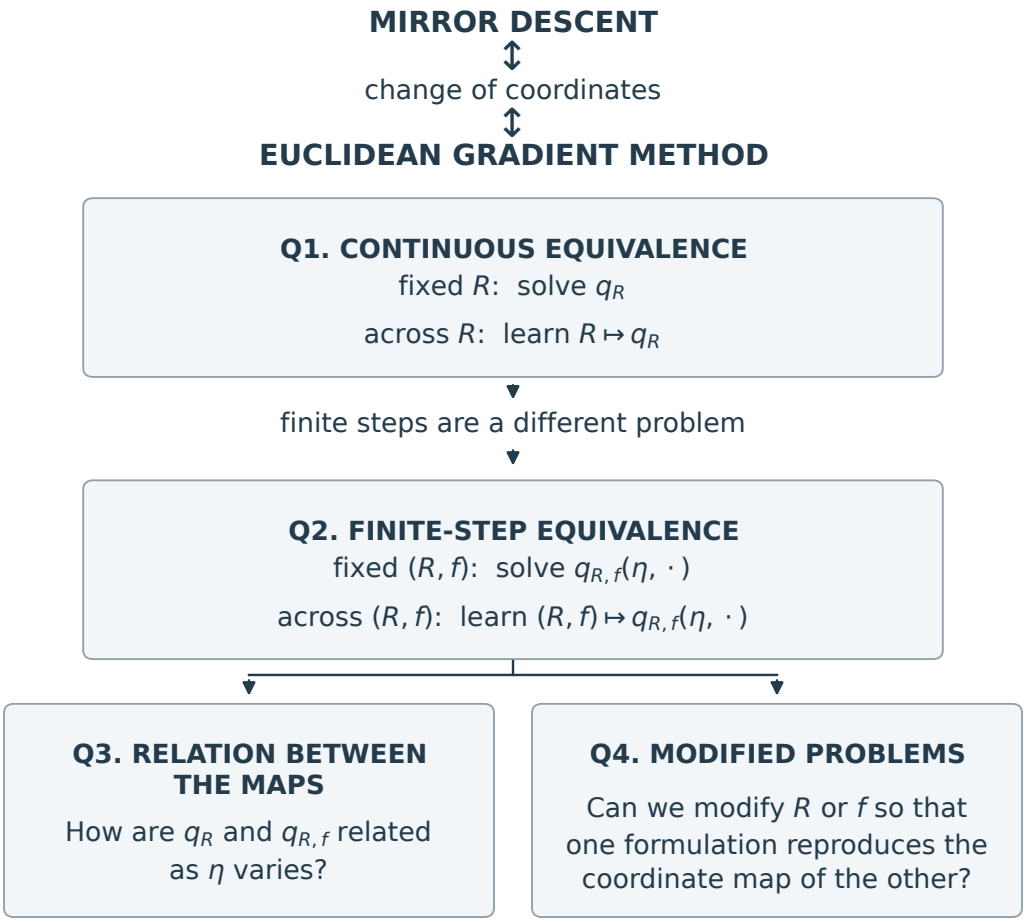


Figure 1: **Overview of the paper.** We first construct and learn coordinate maps for continuous equivalence. We then study exact equivalence of finite steps, where the coordinate map generally depends on both the regularizer and the objective. Next, we study how the continuous and finite-step coordinate maps are related as the step size varies. Finally, we ask whether modifying the objective or regularizer can make one formulation reproduce the coordinate map of the other.

across problem families.

Once the continuous and finite-step coordinate maps have been constructed, we ask how they are related. In particular, we ask whether there is another diffeomorphism $\Phi_{R,f,\eta}$ such that

$$q_R = q_{R,f}(\eta, \cdot) \circ \Phi_{R,f,\eta},$$

and how this relation changes with η.

Finally, we ask whether the problem itself can be modified so that the continuous and finite-step formulations produce the same coordinate map. In particular, we study whether changing R or f can make one formulation reproduce the coordinate map produced by the other.

Figure 1 summarizes the four questions and their relations. The rest of the paper follows these four questions. Section 2 studies continuous equivalence, the construction of q_R, and operator learning of $R \mapsto q_R$. Section 3 studies finite-step equivalence, the objective-dependent coordinate map $q_{R,f}(\eta, \cdot)$, and operator learning across regularizers and objectives. Section 4 studies the relation between the continuous and finite-step coordinate maps as η varies. Section 5 asks whether changing R or f can make the continuous and finite-step formulations produce the same coordinate map. Section 6 concludes the paper.

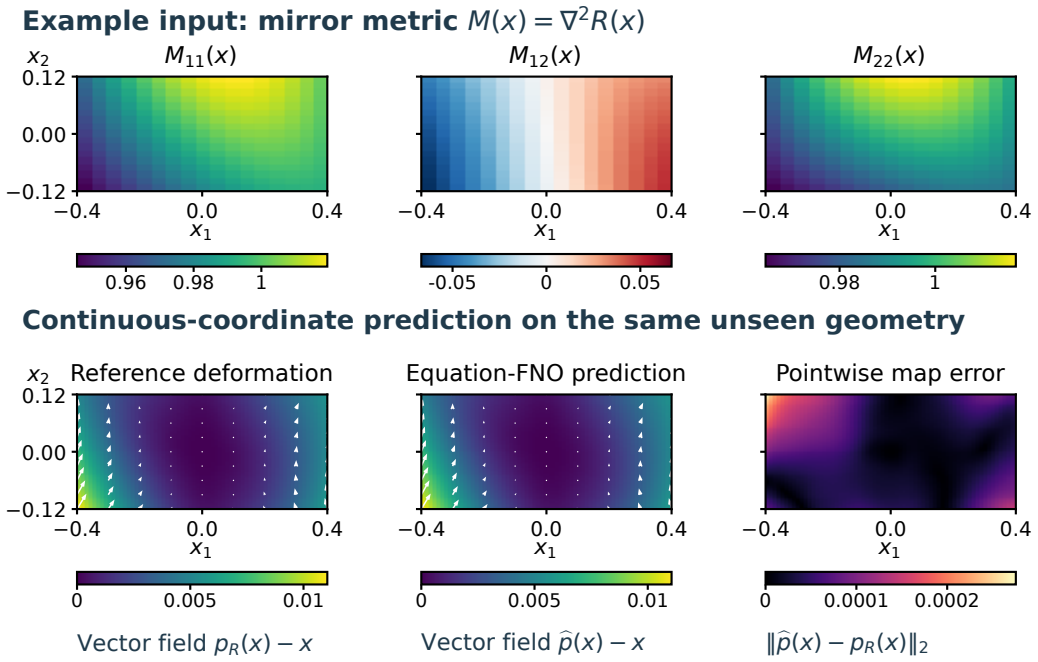


Figure 2: **Learning a vector coordinate map on an unseen geometry.** The input is the mirror metric at physical coordinates $x = (x_1, x_2)$, shown through its three independent components. Reference and equation-trained FNO deformations subtract the identity to reveal what is learned; the output remains a vector map. The final panel shows pointwise vector-map error. The case was specified before viewing the fields (Appendix C.1.1); population results use full-map errors.

2 Continuous Equivalence

We first consider the continuous-time problem. Let x denote the state of mirror flow and let u denote Euclidean coordinates, related by

$$x = q(u).$$

For a mirror regularizer R, define

$$M(x) = \nabla^2 R(x).$$

Mirror flow for an objective f is (Alvarez et al., 2004)

$$\dot{x} = -M(x)^{-1} \nabla f(x).$$

In the u-coordinates, Euclidean gradient flow applied to the objective $f(q(u))$ is

$$\dot{u} = -Dq(u)^T \nabla f(q(u)).$$

Since $x = q(u)$, differentiating x with respect to time shows that the two flows agree for every objective f exactly when

$$Dq(u)^T M(q(u)) Dq(u) = I. \tag{1}$$

We denote a coordinate map satisfying (1) by q_R. Equation (1) depends on the regularizer through $M = \nabla^2 R$, but not on the objective f. This is why the continuous coordinate map depends on R alone.

In one dimension, equation (1) can be solved directly. In higher dimensions, such a coordinate map does not exist for every positive-definite M; the metric must be locally flat (Duistermaat, 2001; Shima, 2007). We therefore use families of regularizers for which this condition holds and focus on how to construct and learn the corresponding maps.

2.1 SOLVING THE COORDINATE MAP FOR A FIXED REGULARIZER

For a fixed R, we solve (1) numerically. It is convenient to work with the inverse coordinate map

$$p_R = q_R^{-1}.$$

In the physical coordinates x, equation (1) becomes

$$Dp_R(x)^T Dp_R(x) = M(x). \tag{2}$$

We parameterize p_R so that it remains invertible on the domain and minimize the residual of (2). The point and derivative of the map at one reference point are fixed to remove the freedom to translate or rotate the Euclidean coordinates.

2.2 LEARNING ACROSS REGULARIZERS

Solving (2) separately for every new regularizer can be expensive. We therefore learn the dependence of the coordinate map on the regularizer across a family of problems. Since equation (2) depends on R through $M = \nabla^2 R$, the numerical model takes the sampled field $M(x)$ as input and predicts the corresponding coordinate map $p_R(x)$.

Figure 2 illustrates this operator-learning task on one unseen regularizer. The input is the matrix-valued function $M(x)$, and the family-trained FNO predicts the corresponding two-dimensional coordinate map $p_R(x)$. The reference and predicted maps are shown through their displacement from the identity, which makes the nonlinear deformation easier to see.

We compare a Fourier neural operator (FNO) (Li et al., 2021), DeepONet (Lu et al., 2021), and a multilayer perceptron (MLP) (Hornik et al., 1989) that receives the same sampled input fields. We also include PCA-Net (Bhattacharya et al., 2021) and POD-DeepONet (Lu et al., 2022) as reduced-basis comparisons. All models are evaluated on regularizers that are not used for training.

Our main experiments use genuinely coupled two-dimensional geometries. This family cannot be reduced to two independent one-dimensional problems by any fixed affine change of variables. We evaluate three groups of unseen regularizers: an in-distribution group, a higher-frequency group with 32 spatial modes rather than the 4, 8, and 16 modes used in training, and a geometry-scale group with larger variation in the metric.

Among the equation-trained models, FNO gives the lowest mean relative H^1 error in all three groups. Its errors are

$$0.12\%, \qquad 0.18\%, \qquad 0.30\%,$$

compared with

$$0.30\%, \qquad 0.35\%, \qquad 0.59\%$$

for the MLP. Here the H^1 error measures both the coordinate map and its first spatial derivatives. Figure 3(a) shows the complete equation-trained comparison.

The same trained FNO can also be evaluated on finer input grids without retraining. Its in-distribution relative H^1 errors are

$$0.120\%, \qquad 0.141\%, \qquad 0.121\%, \qquad 0.121\%$$

at 17, 25, 33, and 65 input points per axis, respectively. Figure 3(b) shows this same-weight resolution transfer. This is consistent with the discretization-independent motivation of neural operators (Kovachki et al., 2023). The other models are trained on a fixed input grid and are evaluated at finer resolutions after resampling the inputs to that grid. Their results therefore do not test native resolution transfer in the same sense.

These results show that the continuous coordinate map can be recovered accurately from the mirror geometry and learned across a nonseparable family of regularizers. They also show that the same FNO remains accurate across the tested input resolutions without retraining.

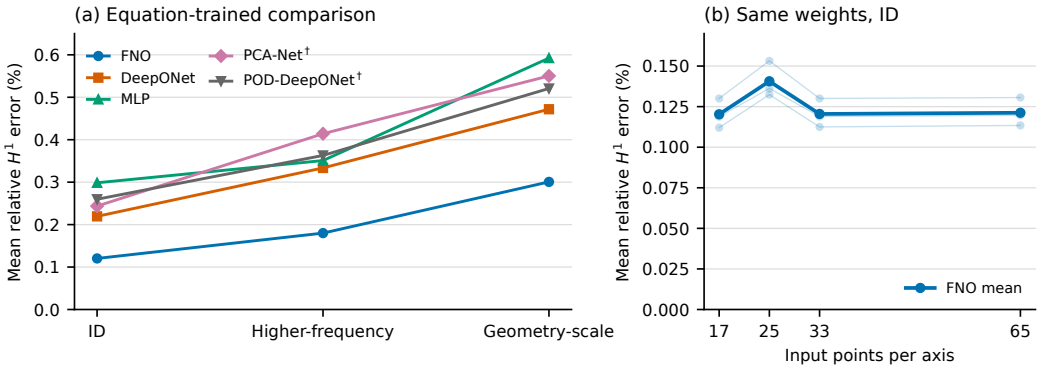


Figure 3: **Continuous operator learning and same-weight resolution transfer.** (a) Mean relative H^1 error of the equation-trained models on three groups of unseen coupled geometries. Each group contains 66 test regularizers. PCA-Net and POD-DeepONet use output bases constructed from training solutions. (b) The same equation-trained FNO weights evaluated on the in-distribution group at four input resolutions, without retraining. The other models resample finer inputs to their fixed training grid and are therefore not shown in the native resolution-transfer panel.

3 Finite-Step Equivalence

We now return to the discrete algorithms. Let $x = q(u)$. Starting from x, one mirror-descent step with step size η is

$$x^+ = (\nabla R)^{-1}\big(\nabla R(x) - \eta \nabla f(x)\big).$$

Starting from u, one Euclidean gradient step for the objective $f(q(u))$ is

$$u^+ = u - \eta Dq(u)^T \nabla f(q(u)).$$

Exact finite-step equivalence means that these two updates correspond under the same coordinate map:

$$q(u^+) = x^+.$$

Equivalently,

$$q\big(u - \eta Dq(u)^T \nabla f(q(u))\big) = (\nabla R)^{-1}\left(\nabla R(q(u)) - \eta \nabla f(q(u))\right).$$

Both algorithms use the same step size η. Unlike the continuous equation in Section 2, this equation depends explicitly on the objective f.

3.1 Why the finite-step map depends on the objective

One might first ask whether the continuous coordinate map q_R can also satisfy the finite-step equation. More generally, could there be a nonlinear coordinate map depending only on R that works for every objective?

In general, the answer is no. Suppose the same coordinate map satisfies the finite-step equation for every gradient direction in a neighborhood of zero. Then the coordinate map must be affine, and $\nabla^2 R$ must be constant. Thus a nonlinear coordinate map depending only on R cannot give exact finite-step equivalence for a general class of objectives.

This motivates allowing the finite-step coordinate map to depend on the objective. We write

$$q_{R,f}(\eta, \cdot)$$

for a coordinate map that satisfies the finite-step equation for a fixed regularizer R, objective f, and step size η.

Nonlinear exact finite-step maps can nevertheless exist for a fixed pair (R, f). For example, suppose x_0 is a stationary point of f and

$$\nabla f(x_0) = 0, \qquad \nabla^2 f(x_0) = c\,\nabla^2 R(x_0), \qquad c > 0.$$

Under the regularity assumptions stated in Appendix A.2.2, if $0 < \eta c < 1$, there is a unique local analytic coordinate map satisfying the finite-step equation once its value and derivative at the reference point are fixed. This gives one setting in which an exact objective-dependent finite-step coordinate map is guaranteed to exist.

3.2 Solving the exact finite-step map

For a fixed pair (R, f), we solve the finite-step equation directly. The unknown is now the function

$$(\eta, u) \longmapsto q_{R,f}(\eta, u).$$

We represent this function with an invertible coordinate-map model and minimize the residual of the finite-step equation over states u and step sizes η. In this way, η is an input to the same finite-step map rather than a separate problem that must be solved from scratch for every step size.

3.3 Learning exact finite-step maps across problem families

Solving the finite-step equation separately for every new pair (R, f) can again be expensive. We therefore study operator learning of

$$(R, f) \longmapsto q_{R,f}(\eta, \cdot).$$

A single model is trained across a family of regularizers and objectives. Given a new pair (R, f), it predicts the corresponding coordinate map as a function of both the state and the step size.

Our family-level benchmark for exact finite-step maps uses one-dimensional problems. This setting allows the regularizer and objective to vary independently while the exact finite-step equation remains tractable over a broad range of states and step sizes. The learned models are trained across these problem pairs and evaluated on unseen (R, f).

On the in-distribution test set, the MLP gives the lowest errors among the three scalar operator models. Its normalized equation RMS is 5.63×10^{-3}, its map RMSE is 6.23×10^{-4}, and its relative derivative RMS is 3.19×10^{-3}. Table 1 compares the three architectures. The full distribution-shift results are reported in Appendix C.2.1.

Table 1: **Exact scalar finite-step operator learning on in-distribution test problems.** Values are means across five independent training runs. All three models use the same positive-derivative output representation.

Model	Equation RMS	Map RMSE	Rel. derivative RMS
FNO	6.56×10^{-3}	7.15×10^{-4}	3.39×10^{-3}
MLP	$\mathbf{5.63 \times 10^{-3}}$	$\mathbf{6.23 \times 10^{-4}}$	$\mathbf{3.19 \times 10^{-3}}$
DeepONet	7.87×10^{-3}	8.96×10^{-4}	4.50×10^{-3}

These results show that exact finite-step coordinate maps can be learned across varying regularizers and objectives rather than recomputed independently for every new problem.

3.4 Small-step approximation in coupled two-dimensional problems

The previous subsection learns the exact finite-step map directly. To study genuinely coupled two-dimensional problems, we also consider the relation between the finite-step and continuous coordinate maps when η is small.

Let

$$p_R = q_R^{-1}.$$

For small η, we approximate the inverse finite-step coordinate map by

$$\widehat{p}_{R,f,\eta} = p_R + \eta a_{R,f}.$$

Substituting this form into the finite-step equation and matching the leading error gives an equation for the first correction $a_{R,f}$. We solve this equation for individual problems and learn the map

$$(R, f) \longmapsto a_{R,f}$$

across the coupled two-dimensional family.

Using the continuous coordinates alone generally leaves a one-step error of order $O(\eta^2)$. Under the assumptions in Appendix A.2.3, including the first correction reduces this error to $O(\eta^3)$.

On the correction field itself, equation-trained FNO has mean relative H^1 errors of 8.32% on the in-distribution group and 10.20% under the geometry-scale shift, the lowest mean H^1 errors among the five equation-trained models on these two complete groups.

The correction nevertheless has a much stronger effect on finite-step algorithmic agreement. At $\eta = 0.02$ over fixed time $T = 0.04$, the mean physical disagreement decreases from 3.58×10^{-5} to 6.95×10^{-7} on the in-distribution group, a factor of approximately 51.5. Under the geometry-scale shift it decreases from 5.52×10^{-5} to 1.61×10^{-6}, a factor of approximately 34.4.

The exact finite-step map and the small-step approximation serve different purposes. The former solves the finite-step equation directly for $q_{R,f}(\eta, \cdot)$. The latter describes how the finite-step coordinate map departs from q_R when the step size is small. In the next section, we study the relation between the continuous and finite-step coordinate maps more directly.

4 Relating the Continuous and Finite-Step Coordinate Maps

The continuous map q_R and the finite-step map $q_{R,f}(\eta, \cdot)$ solve different equations. When both maps are invertible and cover the same physical region, however, their coordinate systems can be compared directly.

Take a point u in the continuous coordinate system. The corresponding physical point is

$$x = q_R(u).$$

The same physical point has a finite-step coordinate

$$\widetilde{u} = q_{R,f}(\eta, \cdot)^{-1}(x).$$

We therefore define

$$\Phi_{R,f,\eta} = q_{R,f}(\eta, \cdot)^{-1} \circ q_R,$$

so that

$$q_R = q_{R,f}(\eta, \cdot) \circ \Phi_{R,f,\eta}.$$

Thus $\Phi_{R,f,\eta}$ simply converts continuous coordinates into finite-step coordinates for the same physical state.

The dependence on η describes how the finite-step coordinate system moves away from the continuous one. For small step sizes, Section 3 gives the approximate connecting map

$$(p_R + \eta a_{R,f}) \circ q_R.$$

For larger step sizes, we can compute the finite-step map directly and compare the two coordinate systems wherever both maps remain defined and invertible on the same region.

We examine this relation numerically through the displacement between the two coordinate systems, the conditioning of $\Phi_{R,f,\eta}$, and the region on which both maps are simultaneously defined. Representative exact scalar examples are reported in Appendix C.3.1. Figure 4 shows a nonlinear numerical example in which the connecting map bends away from the identity as η increases; see Appendix C.3 for the exact scalar controls and additional details, and Appendix B.1.2 for the numerical reference construction.

This relation gives a simple way to study the transition from continuous to finite-step equivalence. Rather than comparing the two algorithms only through their trajectories, we can compare the coordinate systems that make them equivalent. In the next section, we ask a different question: whether changing the objective or regularizer can make the two formulations produce the same coordinate map.

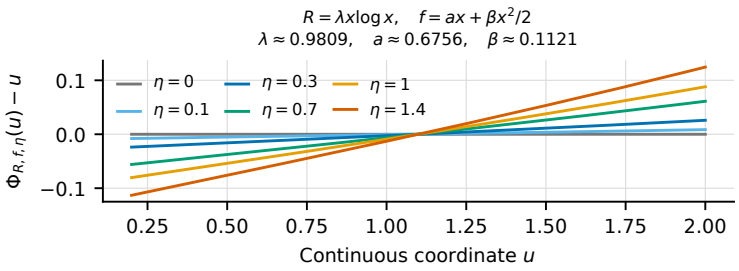


Figure 4: **Connecting coordinates in a nonlinear scalar example.** The numerical displacement $\Phi_{R,f,\eta}(u) - u$ varies nonlinearly with the continuous coordinate. The curves use the first held-out problem from the scalar confirmation study, independently computed finite-step reference maps, and the same source interval $[0.2, 2]$. All displayed connecting maps are increasing.

5 Changing the Objective or Regularizer

So far, R and f have been fixed while the coordinate map changes. We now reverse the question. Suppose a coordinate map is given. Can we change the objective or regularizer so that the same map describes both the continuous and finite-step formulations?

There are two natural directions. First, starting from a continuous coordinate map q_R, we can ask whether a modified objective $\widetilde{f}$, a modified regularizer $\widetilde{R}$, or both can make q_R satisfy the finite-step equation. Changing only the objective is not always sufficient to obtain a nontrivial finite-step equivalence. Allowing the regularizer to change gives a larger class of solutions; Appendix A.4 gives scalar examples and the corresponding equations.

Conversely, suppose a finite-step coordinate map

$$Q = q_{R,f}(\eta, \cdot)$$

is given. In one dimension, a continuous regularizer that produces the same coordinate map must satisfy

$$\widetilde{R}''(x) = \left[Q'\big(Q^{-1}(x)\big)\right]^{-2}.$$

Thus the target coordinate map directly determines the curvature of the modified regularizer. In higher dimensions, the analogous matrix field need not be the Hessian of any scalar regularizer, so not every coordinate map can be reproduced in this way.

We also test the one-dimensional reconstruction numerically. In the direction from a finite-step target map Q to a modified continuous regularizer, all 320 learned predictions are successfully reconstructed by an independent continuous solve. The mean transform RMSE is

$$(1.65 \pm 0.36) \times 10^{-6}$$

across five independent training runs, compared with 3.09×10^{-7} for the cubic numerical baseline. Results for the reverse modification are reported in Appendix C.4.

These constructions show that disagreement between the continuous and finite-step coordinate maps can sometimes be moved from the coordinates into the underlying optimization problem. We give the detailed constructions and numerical examples in Appendices A.4 and C.4. The main results of the paper, however, concern the original R and f studied in Sections 2–4.

6 Discussion and Conclusion

This paper separates two questions that are easy to conflate. In continuous time, mirror descent and Euclidean gradient flow can sometimes be related by a coordinate map determined by the mirror geometry. At finite step size, the problem changes: a nonlinear coordinate map that works independently of the objective is generally impossible, and exact equivalence instead leads naturally to an objective-dependent coordinate map. This distinction is the main conceptual message of the paper.

The same viewpoint leads to a family of computational problems. For individual regularizers and objectives, the coordinate maps can be obtained by solving their defining equations. Across problem families, these constructions become operator-learning problems. Our experiments show that continuous coordinate maps can be learned accurately on genuinely coupled two-dimensional geometries. In one dimension, we learn an approximation to the operator

$$(R, f) \mapsto q_{R,f}(\eta, \cdot).$$

For coupled two-dimensional problems, the small-step approximation provides a practical way to connect the continuous and finite-step maps. The preferred architecture is problem dependent: FNO gives the lowest mean H^1 error on the coupled two-dimensional continuous and first-correction benchmarks, while the MLP gives the lowest in-distribution errors in the scalar exact finite-step benchmark. The same FNO weights also remain accurate across the tested input resolutions without retraining.

Several questions remain open. Extending exact finite-step maps to higher-dimensional coupled problems remains an open direction. Their existence may depend on the step size and the region of state space under consideration. The relation between continuous and finite-step coordinate systems also deserves a more complete study beyond the local and numerical settings considered here. More broadly, the results suggest that algorithmic equivalence is not a single notion: continuous-time equivalence, exact finite-step equivalence, and equivalence after modifying the underlying problem are distinct questions that should be studied separately.

REPRODUCIBILITY STATEMENT

We provide complete proofs of the theoretical results and additional technical details in the appendix. The experimental setup, implementation details, hyperparameters, and additional numerical results are also documented in the appendix.

AI USE STATEMENT

Generative AI tools were used to assist with software implementation and debugging, as well as with drafting and editing portions of the manuscript for clarity and presentation. The core research questions, methodology, theoretical results, and scientific conclusions were developed by the authors. All AI-assisted code and text were reviewed and verified by the authors, who take full responsibility for the final content of the paper.

REFERENCES

Felipe Alvarez, Jérôme Bolte, and Olivier Brahic. Hessian Riemannian gradient flows in convex programming. *SIAM Journal on Control and Optimization*, 43(2):477–501, 2004. doi: 10.1137/S0363012902419977. URL https://epubs.siam.org/doi/10.1137/S0363012902419977.

Ehsan Amid and Manfred K. Warmuth. Reparameterizing mirror descent as gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8430–8439, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/604b37ea63ea51fa5fb3d8a89ec056e6-Abstract.html.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. doi: 10.1016/S0167-6377(02)00231-6. URL https://www.sciencedirect.com/science/article/pii/S0167637702002316.

Mario Bertero and Patrizia Boccacci. *Introduction to Inverse Problems in Imaging*. Institute of Physics Publishing, 1998. ISBN 9780750304351. doi: 10.1201/9781439822067. URL https://doi.org/10.1201/9781439822067.

Kaushik Bhattacharya, Bamdad Hosseini, Nikola B. Kovachki, and Andrew M. Stuart. Model reduction and neural networks for parametric PDEs. *SMAI Journal of Computational Mathematics*, 7:121–157, 2021. doi: 10.5802/smai-jcm.74. URL https://smai-jcm.centre-mersenne.org/item/?id=SMAI-JCM_2021__7__121_0.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. doi: 10.1016/0041-5553(67)90040-7. URL https://www.sciencedirect.com/science/article/pii/0041555367900407.

Robert L. Bryant. Lecture 2: Some classical PDE background and generalizations. Eilenberg lecture notes, February 2026. URL https://sites.math.duke.edu/~bryant/Eilenberg/Lecture02.pdf. Draft version 1.0, February 3, 2026.

Philippe Chartier, Ernst Hairer, and Gilles Vilmart. Numerical integrators based on modified differential equations. *Mathematics of Computation*, 76(260):1941–1953, 2007. doi: 10.1090/S0025-5718-07-01967-9. URL https://doi.org/10.1090/S0025-5718-07-01967-9.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4):911–917, 1995. doi: 10.1109/72.392253. URL https://doi.org/10.1109/72.392253.

J. J. Duistermaat. On Hessian Riemannian structures. *Asian Journal of Mathematics*, 5 (1):79–91, 2001. doi: 10.4310/AJM.2001.v5.n1.a6. URL https://archive.intlpress.com/site/pub/files/_fulltext/journals/ajm/2001/0005/0001/AJM-2001-0005-0001-a006.pdf.

Udaya Ghai, Zhou Lu, and Elad Hazan. Non-convex online learning via algorithmic equivalence. In *Advances in Neural Information Processing Systems*, volume 35, pp. 22161–22172, 2022. doi: 10.52202/068431-1610.

Suriya Gunasekar, Blake Woodworth, and Nathan Srebro. Mirrorless mirror descent: A natural derivation of mirror descent. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2305–2313. PMLR, 2021. URL https://proceedings.mlr.press/v130/gunasekar21a.html.

Ernst Hairer, Christian Lubich, and Gerhard Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, volume 31 of *Springer Series in Computational Mathematics*. Springer, 2 edition, 2006. ISBN 978-3-540-30663-4. doi: 10.1007/3-540-30666-8. URL https://link.springer.com/book/10.1007/3-540-30666-8.

J. S. Hesthaven and S. Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018. doi: 10.1016/j.jcp.2018.02.037. URL https://www.sciencedirect.com/science/article/pii/S0021999118301190.

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. doi: 10.1016/0893-6080(89)90020-8. URL https://doi.org/10.1016/0893-6080(89)90020-8.

Pengzhan Jin, Shuai Meng, and Lu Lu. MIONet: Learning multiple-input operators via tensor product. *SIAM Journal on Scientific Computing*, 44(6):A3490–A3514, 2022. doi: 10.1137/22M1477751. URL https://epubs.siam.org/doi/10.1137/22M1477751.

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for Fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021. URL https://jmlr.org/papers/v22/21-0806.html.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023. URL https://jmlr.org/papers/v24/21-1524.html.

Samuel Lanthaler. Operator learning with PCA-Net: upper and lower complexity bounds. *Journal of Machine Learning Research*, 24(318):1–67, 2023. URL https://jmlr.org/papers/v24/23-0478.html.

Samuel Lanthaler, Siddhartha Mishra, and George Em Karniadakis. Error estimates for DeepONets: a deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022. doi: 10.1093/imatrm/tnac001. URL https://academic.oup.com/imatrm/article/6/1/tnac001/6542709.

Zhiyuan Li, Tianhao Wang, Jason D. Lee, and Sanjeev Arora. Implicit bias of gradient descent on reparametrized models: On equivalence to mirror descent. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34626–34640, 2022. doi: 10.52202/068431-2509.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *ACM / IMS Journal of Data Science*, 1(3):1–27, 2024. doi: 10.1145/3648506. URL https://doi.org/10.1145/3648506.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3:218–229, 2021. doi: 10.1038/s42256-021-00302-5. URL https://www.nature.com/articles/s42256-021-00302-5.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on FAIR data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022. doi: 10.1016/j.cma.2022.114778. URL https://doi.org/10.1016/j.cma.2022.114778.

Arkadii Semenovich Nemirovsky and David Borisovich Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, 1983. ISBN 0471103454. URL https://www2.isye.gatech.edu/~nemirovs/Nemirovskii_Yudin_1983.pdf. Translated by E. R. Dawson.

Hirohiko Shima. *The Geometry of Hessian Structures*. World Scientific, 2007. ISBN 9789812700315. doi: 10.1142/6241. URL https://www.worldscientific.com/worldscibooks/10.1142/6241.

Burt Totaro. The curvature of a Hessian metric. *International Journal of Mathematics*, 15:369–391, 2004. URL https://www.math.ucla.edu/~totaro/papers/public_html/hessian.pdf.

Sifan Wang, Hanwen Wang, and Paris Perdikaris. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets. *Science Advances*, 7(40):eabi8605, 2021. doi: 10.1126/sciadv.abi8605. URL https://www.science.org/doi/10.1126/sciadv.abi8605.

Aiqing Zhu, Tom Bertalan, Beibei Zhu, Yifa Tang, and Ioannis G. Kevrekidis. Implementation and (inverse modified) error analysis for implicitly templated ODE-Nets. *SIAM Journal on Applied Dynamical Systems*, 23(4):2643–2669, 2024. doi: 10.1137/23M1564596. URL https://epubs.siam.org/doi/10.1137/23M1564596.

A Mathematical Foundations

This appendix gives the mathematical details behind the four questions in the main paper. We follow the same order as the main text. We first study continuous equivalence, then finite-step equivalence, the relation between the two coordinate systems, and finally the effect of changing the objective or regularizer.

A.1 Continuous Equivalence

We begin with the continuous-time problem from Section 2. There are two questions to address. First, when can a change of coordinates turn mirror flow into Euclidean gradient flow? Second, how can we construct a nontrivial multidimensional family for which such coordinates exist? The first question leads to a flatness condition on the mirror geometry; the second motivates the coupled two-dimensional family used in our experiments.

A.1.1 Coordinate Equation and Flatness

Let $X \subset \mathbb{R}^d$ be open and let

$$M(x) = \nabla^2 R(x) \succ 0$$

be the metric defined by the mirror regularizer R. A coordinate map q_R relates Euclidean coordinates u to the physical state x through

$$x = q_R(u).$$

As shown in Section 2, mirror flow and Euclidean gradient flow agree under this change of coordinates when

$$Dq_R(u)^T M(q_R(u)) Dq_R(u) = I.$$

It is often more convenient to use the inverse coordinate map

$$p_R = q_R^{-1}.$$

In the physical coordinates x, the previous equation is equivalent to

$$Dp_R(x)^T Dp_R(x) = M(x).$$

Thus p_R gives Euclidean coordinates for the metric M.

In one dimension, this equation can always be solved locally. Choosing the increasing orientation gives

$$p_R'(x) = \sqrt{R''(x)},$$

and therefore

$$p_R(x) = p_R(x_0) + \int_{x_0}^{x} \sqrt{R''(s)}\, ds.$$

This formula gives the exact coordinate maps used in our one-dimensional examples.

The higher-dimensional problem is different. Positive definiteness of M alone does not imply that such coordinates exist. They can exist locally only when the Riemannian metric defined by M is flat (Duistermaat, 2001; Shima, 2007). Conversely, on a sufficiently small simply connected flat region, one can introduce local Euclidean coordinates.

The coordinate map is not unique without specifying its origin and orientation. We therefore prescribe $p_R(x_0)$ and $Dp_R(x_0)$ at one reference point. This removes translations and orthogonal changes of the Euclidean coordinates.

The flatness requirement means that an arbitrary positive-definite Hessian cannot be used in our multidimensional experiments. We therefore construct a function-valued family for which the metric is a Hessian, positive definite, and flat by construction.

A.1.2 A COUPLED TWO-DIMENSIONAL FLAT FAMILY

We now construct the two-dimensional geometries used in Section 2. The construction must satisfy three requirements: M must be a Hessian metric, it must be positive definite, and it must be flat. We also want the two coordinates to interact genuinely rather than reduce to two independent one-dimensional examples.

Let a_0, z_0, and ψ be analytic functions, with

$$a_0 > 0, \qquad \psi(t) > t^2$$

on the interval of interest. We solve

$$\partial_2 a = \partial_1 b, \qquad \partial_2 b = \partial_1 [a\, \psi(b/a)]\,,$$

with initial data

$$a(s,0) = a_0(s), \qquad b(s,0) = a_0(s) z_0(s).$$

Define

$$M(x) = \begin{pmatrix} a & b \\ b & a\,\psi(b/a) \end{pmatrix}.$$

We first verify that M is a Hessian. Writing

$$c = a\,\psi(b/a),$$

the two equations above become

$$\partial_2 M_{11} = \partial_1 M_{12}, \qquad \partial_2 M_{12} = \partial_1 M_{22}.$$

These are the compatibility conditions for the entries of M to arise as second derivatives of a scalar function. Hence, locally,

$$M = \nabla^2 R$$

for a regularizer R, unique up to an affine function.

Positive definiteness follows from

$$\det M = a^2 \left[\psi(b/a) - (b/a)^2\right] > 0.$$

It remains to verify flatness. Set

$$t = b/a, \qquad \nu(t) = (1, t, \psi(t)).$$

Then

$$(M_{11}, M_{12}, M_{22}) = a\,\nu(t).$$

The derivatives of this vector lie in the span of $\nu(t)$ and $\nu'(t)$. The determinant appearing in the curvature condition for a two-dimensional Hessian metric therefore vanishes, so the metric is flat (Totaro, 2004).

Finally, we show that the family is genuinely coupled. Suppose ψ is strictly convex and z_0 takes at least three distinct values. The corresponding matrices

$$\begin{pmatrix} 1 & t \\ t & \psi(t) \end{pmatrix}$$

then span the three-dimensional space of symmetric 2×2 matrices. By contrast, a fixed affine mixing of two independent one-dimensional regularizers produces Hessians in the span of only two fixed rank-one matrices. Therefore the family above cannot be reduced to two independent one-dimensional problems by a fixed affine change of coordinates.

This conclusion rules out fixed affine separation; it does not claim that no other nonlinear representation exists.

We have therefore obtained a function-valued family of genuinely coupled two-dimensional mirror geometries for which the continuous coordinate equation is well defined. Appendix B explains how the corresponding coordinate maps are computed numerically and how they are learned across the family.

A.2 FINITE-STEP EQUIVALENCE

We now turn from continuous flow to finite steps. The exact finite-step equation is more restrictive because the coordinate map appears not only in the physical state $q(u)$, but also in the Euclidean update through $Dq(u)^T \nabla f(q(u))$. This section develops the theory in three steps. We first show that one nonlinear coordinate map cannot work for a general class of objectives. We then give a setting in which an objective-dependent exact coordinate map exists locally. Finally, for more general problems and small step size, we derive corrections to the continuous coordinate map.

A.2.1 AN OBJECTIVE-INDEPENDENT FINITE-STEP MAP MUST BE AFFINE

Fix a step size $\eta > 0$. Suppose we seek a single coordinate map q that gives exact finite-step equivalence without depending on the objective. At a state u, allowing the objective gradient to vary is equivalent to allowing the vector

$$g = \nabla f(q(u))$$

to vary locally. The exact finite-step equation then becomes

$$q\big(u - \eta Dq(u)^T g\big) = (\nabla R)^{-1} \left(\nabla R(q(u)) - \eta g\right).$$

The following result shows that requiring this identity for a neighborhood of gradient directions leaves only the affine case.

Proposition 1 (An objective-independent finite-step map must be affine)**.** *Let $U \subset \mathbb{R}^d$ be connected and open, and let*

$$q : U \to V \subset X$$

be a C^1 diffeomorphism between domains of the same dimension. Let $R \in C^2(X)$ satisfy

$$\nabla^2 R \succ 0.$$

Suppose that, for every $u \in U$, the finite-step identity above holds for every g in a neighborhood of zero for which both updates remain in their domains. Then

$$q(u) = Au + b$$

for a constant invertible matrix A, and

$$\nabla^2 R(x) = A^{-T} A^{-1}$$

on V. Conversely, these forms satisfy the finite-step equation whenever the updates are defined.

Proof. Define

$$P = \nabla R \circ q.$$

Since $Dq(u)$ is invertible, varying g in a neighborhood of zero is equivalent to varying

$$v = -\eta Dq(u)^T g$$

in a neighborhood of zero. Applying ∇R to both sides of the finite-step equation gives

$$P(u + v) = P(u) + Dq(u)^{-T} v.$$

For each fixed u, this equation shows that P is affine on a neighborhood of u. Hence DP is locally constant. Because U is connected, DP is constant throughout U.

Evaluating the derivative with respect to v at $v = 0$ gives

$$DP(u) = Dq(u)^{-T}.$$

Since DP is constant, Dq is also constant. Therefore

$$q(u) = Au + b$$

for some constant invertible matrix A.

Finally,

$$DP(u) = \nabla^2 R(q(u))A = A^{-T},$$

and hence

$$\nabla^2 R(q(u)) = A^{-T} A^{-1}.$$

Since $q(U) = V$, the Hessian is constant on V. Direct substitution proves the converse. □

This proposition does not say that the objective f must be quadratic, nor does it rule out nonlinear finite-step coordinate maps for a fixed objective. It says that one nonlinear coordinate map cannot work for a whole neighborhood of objective gradients. This is why the finite-step map in the remainder of the paper is allowed to depend on f.

A.2.2 Local Existence of Objective-Dependent Finite-Step Coordinates

Once the coordinate map is allowed to depend on f, nonlinear exact finite-step equivalence becomes possible. We first give a local result near a stationary point. The assumptions are restrictive, but they provide an explicit setting in which the exact finite-step equation has a well-defined solution.

Let x_0 satisfy

$$\nabla f(x_0) = 0,$$

and define

$$M_0 = \nabla^2 R(x_0).$$

We assume that the objective and mirror Hessians are proportional at x_0:

$$\nabla^2 f(x_0) = cM_0, \qquad c > 0.$$

Theorem 2 (Local existence of objective-dependent finite-step coordinates)**.** *Suppose R and f are real analytic near x_0, with*

$$M_0 = \nabla^2 R(x_0) \succ 0, \qquad \nabla f(x_0) = 0, \qquad \nabla^2 f(x_0) = cM_0, \qquad c > 0.$$

For each fixed step size satisfying

$$0 < \eta c < 1,$$

there is a unique local real-analytic solution q_η, among solutions with the following point and frame, of the exact finite-step equation:

$$q_\eta(0) = x_0, \qquad Dq_\eta(0) = M_0^{-1/2}.$$

After restricting to sufficiently small neighborhoods, q_η is a diffeomorphism and both the mirror and Euclidean updates remain inside their respective domains.

Proof. We first normalize the stationary point and rewrite the exact finite-step equation in terms of the inverse coordinate map

$$p_\eta = q_\eta^{-1}.$$

We then solve the resulting analytic equation by a contraction argument and finally verify local invertibility and uniqueness.

Normalization and the equation for the inverse coordinate map. Apply the fixed affine change of physical coordinates

$$y = M_0^{1/2}(x - x_0).$$

This sends the stationary point to the origin and the mirror Hessian there to the identity. The mirror-descent equation and the pulled-back Euclidean gradient retain the same form under this transformation. We therefore work in the normalized coordinates and write them again as x. Thus

$$x_0 = 0, \qquad \nabla^2 R(0) = I, \qquad Dq_\eta(0) = I.$$

Let

$$m(x) = (\nabla R)^{-1}\big(\nabla R(x) - \eta \nabla f(x)\big)$$

denote one mirror step. Since

$$\nabla f(0) = 0, \qquad \nabla^2 f(0) = cI,$$

we have

$$m(x) = \lambda x + \mu(x), \qquad \lambda = 1 - \eta c \in (0, 1), \qquad \mu(x) = O(|x|^2).$$

Set

$$\delta = 1 - \lambda = \eta c.$$

We also write

$$\eta \nabla f(x) = \delta x + e(x), \qquad e(x) = O(|x|^2),$$

and define

$$b(x) = \mu(x) + e(x).$$

Let

$$p_\eta = q_\eta^{-1}.$$

Exact finite-step equivalence is equivalent to

$$Dp_\eta(x)^T \left[p_\eta(x) - p_\eta(m(x))\right] = \eta \nabla f(x).$$

We seek a solution of the form

$$p_\eta(x) = x + r(x), \qquad r(0) = 0, \qquad Dr(0) = 0.$$

Define the linear operator

$$Lr = r - r(\lambda x) + \delta Dr(x)^T x.$$

Substituting the preceding form into the inverse-coordinate equation gives the exact equation

$$Lr = b + [r \circ m - r(\lambda x)] - D\mu^T r - D(r \circ m)^T r + \nabla S(r),$$

where

$$S(r) = r \cdot \mu - \frac{1}{2} r \cdot r + r \cdot (r \circ m).$$

The advantage of this form is that the derivative of the unknown r at the unshifted point appears only inside the gradient term $\nabla S(r)$. The other derivative of r is evaluated at the contracting map m.

The linear equation and its inverse. We now introduce the analytic norm used in the contraction argument. For a scalar power series

$$s(x) = \sum_\alpha s_\alpha x^\alpha,$$

define

$$\|s\|_\rho = \sum_\alpha |s_\alpha| \rho^{|\alpha|}.$$

For a vector-valued series we sum the component norms. This norm is submultiplicative.

Let $\mathcal{A}_\rho$ be the Banach space of real-coefficient analytic vector fields with finite norm above and with zero constant and linear terms. For a homogeneous vector polynomial r_n of degree $n \geq 2$, define

$$P_n r_n = \frac{\nabla(x \cdot r_n)}{n+1}, \qquad Q_n = I - P_n.$$

Euler's identity gives

$$P_n^2 = P_n,$$

and

$$\|P_n\| \leq 1$$

in the coefficient norm.

On degree n, the operator L decomposes as

$$L_n = \alpha_n Q_n + \beta_n P_n,$$

with

$$\alpha_n = \lambda - \lambda^n, \qquad \beta_n = \alpha_n + \delta(n+1).$$

Because $0 < \lambda < 1$,

$$\alpha_n \geq \lambda\delta > 0, \qquad \beta_n \geq \delta(n+1) > 0.$$

Thus every L_n is invertible.

Moreover,

$$\|L^{-1} v\|_\rho \leq C_0 \|v\|_\rho, \qquad C_0 = \frac{2}{\lambda\delta}.$$

We also need a stronger estimate for gradient terms. If the scalar series s has order at least three, then

$$\|L^{-1}\nabla s\|_\rho \le \frac{\|s\|_\rho}{\delta\rho}.$$

This estimate is important because differentiation by itself is not bounded in the boundary norm. The factor β_n, which grows linearly with n, compensates for the derivative.

Nonlinear estimates and contraction. Define

$$m_\rho = \|\mu\|_\rho, \qquad d_\rho = \max_i \sum_j \|\partial_j \mu_i\|_\rho, \qquad B_\rho = \|b\|_\rho.$$

Since μ and b vanish to second order,

$$m_\rho = O(\rho^2), \qquad d_\rho = O(\rho), \qquad B_\rho = O(\rho^2).$$

Let

$$\sigma = \frac{1+\lambda}{2}, \qquad K_\sigma = (1-\sigma)^{-1},$$

and choose ρ small enough that

$$m_\rho \le (\sigma - \lambda)\rho = \frac{\delta\rho}{2}.$$

Then both $m(x)$ and the segment between λx and $m(x)$ remain inside the smaller analytic polydisc controlled by $\sigma\rho$.

If $A = \|r\|_\rho$, standard power-series composition estimates give

$$\|r \circ m\|_\rho \le A,$$

$$\|r \circ m - r(\lambda x)\|_\rho \le K_\sigma \frac{m_\rho}{\rho} A,$$

and

$$\|D(r \circ m)^T r\|_\rho \le K_\sigma \frac{\lambda + d_\rho}{\rho} A^2.$$

The scalar term satisfies

$$\|S(r)\|_\rho \le m_\rho A + \frac{3}{2}A^2.$$

On a ball

$$\|r\|_\rho, \|v\|_\rho \le r_*,$$

we also have

$$\|S(r) - S(v)\|_\rho \le (m_\rho + 3r_*)\|r - v\|_\rho.$$

Let $T(r)$ be L^{-1} applied to the right-hand side of the exact equation for r, with the gradient term estimated using the stronger inverse-gradient estimate above. Define

$$\ell_\rho = C_0 \left(K_\sigma \frac{m_\rho}{\rho} + d_\rho \right) + \frac{m_\rho}{\delta\rho},$$

$$Q_\rho = C_0 K_\sigma \frac{\lambda + d_\rho}{\rho} + \frac{3}{2\delta\rho},$$

and

$$r_* = 2C_0 B_\rho + \rho^2.$$

The preceding estimates imply

$$\|T(r)\|_\rho \le C_0 B_\rho + \ell_\rho A + Q_\rho A^2,$$

and

$$\|T(r) - T(v)\|_\rho \le (\ell_\rho + 2Q_\rho r_*)\,\|r - v\|_\rho.$$

Because

$$\ell_\rho = O(\rho), \qquad Q_\rho = O(\rho^{-1}), \qquad r_* = O(\rho^2),$$

we may choose ρ sufficiently small that, in addition to the composition condition above,

$$\ell_\rho + 2Q_\rho r_* \le \frac{1}{2}, \qquad \frac{2r_*}{\rho} < \frac{1}{2}.$$

The closed ball

$$\{r \in \mathcal{A}_\rho : \|r\|_\rho \le r_*\}$$

is then mapped into itself by T, and T is a contraction on this ball. The contraction principle gives a real-analytic solution r of the exact equation, hence an analytic solution

$$p_\eta = I + r$$

of the inverse finite-step equation.

Invertibility, admissible updates, and uniqueness. It remains to verify that the analytic solution defines a valid local coordinate map. On the real box

$$X_\rho = \{x : |x_i| < \rho/2\},$$

the coefficient norm gives

$$\sup_{X_\rho} \|Dr(x)\|_2 \le \frac{2r_*}{\rho} < \frac{1}{2}.$$

Therefore

$$p_\eta(x) = x + r(x)$$

has nonsingular Jacobian throughout X_ρ. Since r is Lipschitz there with constant less than one, p_η is also injective on the convex box. Hence

$$q_\eta = p_\eta^{-1}$$

is a real-analytic diffeomorphism between sufficiently small neighborhoods.

The mirror update also remains inside the physical neighborhood. Since

$$m(x) = \lambda x + \mu(x)$$

and μ vanishes to second order, the preceding choice of ρ gives

$$m(X_\rho) \subset X_\rho$$

after shrinking ρ if necessary. The inverse-coordinate equation then shows that the corresponding Euclidean source update remains inside $p_\eta(X_\rho)$.

Finally, suppose two analytic solutions satisfy the same point and frame conditions. If their Taylor expansions first differ at homogeneous degree $n \ge 2$, the first nonzero difference must lie in the kernel of L_n. But L_n is invertible. Therefore no such first degree exists, and the two local analytic solutions agree.

Undoing the initial affine normalization proves the theorem in the original coordinates. □

The proportional-Hessian assumption is local. The theorem does not require the mirror metric to be flat on a surrounding neighborhood. Flatness is needed for the objective-independent continuous coordinate map in Appendix A.1, but not for this fixed-objective finite-step result.

Exact finite-step maps need not be unique outside this setting. The uniqueness above depends on the stationary analytic assumptions and the specified point and frame. In general, exact finite-step coordinate maps need not be unique even after fixing the point and frame.

For example, let

$$R(x) = \frac{1}{2}|x|^2, \qquad f(x) = x_1.$$

For a fixed sufficiently small η, both

$$q_\eta(u_1, u_2) = (u_1, u_2)$$

and

$$q_\eta(u_1, u_2) = \left(u_1,\ u_2 + \eta u_2^3\right)$$

satisfy the exact finite-step equation on suitable local domains. They have the same point and frame at the origin,

$$q_\eta(0) = 0, \qquad Dq_\eta(0) = I,$$

and both converge to the identity as $\eta \to 0$. Thus point and frame normalization together with the continuous limit do not select a unique exact finite-step map in general.

A.2.3 SMALL-STEP CORRECTIONS

The previous theorem gives an exact finite-step coordinate map under a particular stationary-point condition. Away from that setting, we instead ask a local small-step question: starting from a known continuous coordinate map q_R, how must it change so that the finite-step equation is satisfied to higher order in η?

Let

$$J = Dq_R, \qquad v = -J^T \nabla f(q_R).$$

Here v is the Euclidean-coordinate velocity of the continuous flow. We look for a corrected coordinate map of the form

$$\widehat{q}_\eta = q_R \circ (\mathrm{id} + \eta z).$$

Define

$$d = \frac{1}{2} J^{-1} D^2 q_R[v, v].$$

Expanding the finite-step equation in η gives

$$r_\eta(\widehat{q}_\eta) = \eta^2 J \left[(Dz + Dz^T)v - d \right] + O(\eta^3),$$

where r_η denotes the difference between the mapped Euclidean step and the mirror step.

Thus the leading $O(\eta^2)$ error is removed when

$$(Dz + Dz^T)v = \frac{1}{2} J^{-1} D^2 q_R[v, v].$$

Theorem 3 (Finite-step corrections to arbitrary order)**.** *Suppose R, f, and q_R are real analytic on an interior region where*

$$\nabla^2 R \succ 0$$

and q_R is a diffeomorphism. Let $v \neq 0$ at the point under consideration, and let Σ be an analytic hypersurface with normal n satisfying

$$v \cdot n \neq 0.$$

An analytic value of z prescribed on Σ determines a unique local analytic solution of the first-correction equation.

More generally, for every fixed N, prescribed analytic boundary values determine functions

$$z_1, \ldots, z_N$$

such that

$$\widehat{q}_\eta^{[N]} = q_R \circ \left(\mathrm{id} + \sum_{j=1}^{N} \eta^j z_j \right)$$

is a diffeomorphism on a sufficiently small bounded convex neighborhood for sufficiently small η, and its one-step finite-step residual is

$$O(\eta^{N+2})$$

on interior compact subsets. The neighborhood and admissible range of η may depend on N.

Proof. The first-correction equation is a first-order system for z. Along a hypersurface with normal n, the coefficient multiplying the normal derivative is

$$(v \cdot n) I + n v^T.$$

Its determinant is

$$2(v \cdot n)^d,$$

which is nonzero under the stated condition. The hypersurface is therefore noncharacteristic, and the analytic Cauchy–Kovalevskaya theorem gives a unique local analytic solution for the prescribed boundary values (Bryant, 2026).

For higher orders, substitute the finite expansion above into the exact finite-step equation and expand in powers of η. After $z_1, \ldots, z_{j-1}$ have been determined, the next coefficient z_j enters through the same first-order operator as in the first-correction equation; all remaining terms depend only on the

previously determined coefficients. The same noncharacteristic argument therefore determines z_j. Repeating this construction gives $z_1, \dots, z_N$.

Because

$$\mathrm{id} + \sum_{j=1}^{N} \eta^j z_j$$

is a sufficiently small C^1 perturbation of the identity on a small bounded convex neighborhood, it is a diffeomorphism for sufficiently small η. Composing with q_R preserves this property. Matching the coefficients through order $N+1$ leaves a one-step residual of order $O(\eta^{N+2})$. □

For $N = 1$, reusing q_R alone leaves an $O(\eta^2)$ one-step error, while the first corrected map has $O(\eta^3)$ one-step error.

The main experiments use inverse coordinates. To first order, the inverse of the corrected map is

$$\widehat{p}_\eta = p_R + \eta a_{R,f}, \qquad a_{R,f} = -z \circ p_R.$$

This is the correction field learned in the coupled two-dimensional experiments.

If the two discrete trajectories start at the same physical state and remain in a common compact interior region, the physical one-step residual is bounded by $C\eta^3$ there, and the mirror update has Lipschitz constant at most $1 + L\eta$ there, with C, L independent of η, then the disagreement is $O(\eta^2)$ for $k\eta \le T$ with T fixed. This is the order statement used to interpret the independently computed first correction; a learned approximation to the correction need not reproduce the same asymptotic order exactly.

A.3 Relation Between the Continuous and Finite-Step Coordinate Maps

The continuous coordinate map q_R and the finite-step coordinate map $q_{R,f}(\eta, \cdot)$ solve different equations. When both maps are available, we can compare the two coordinate systems by asking how the same physical point is represented in each one.

Let

$$q_R : U_R \to X_R$$

and

$$q_{R,f}(\eta, \cdot) : U_{R,f,\eta} \to X_{R,f,\eta}$$

be invertible coordinate maps. If

$$q_R(u) \in X_R \cap X_{R,f,\eta},$$

then the continuous coordinate u and the corresponding finite-step coordinate are connected by

$$\Phi_{R,f,\eta}(u) = q_{R,f}(\eta, \cdot)^{-1}(q_R(u)).$$

Equivalently,

$$\Phi_{R,f,\eta} = q_{R,f}(\eta, \cdot)^{-1} \circ q_R,$$

and therefore

$$q_R = q_{R,f}(\eta, \cdot) \circ \Phi_{R,f,\eta}$$

wherever the composition is defined.

Thus the connecting map $\Phi_{R,f,\eta}$ does not introduce a third optimization problem. It simply maps the continuous coordinate of a physical state to the corresponding finite-step coordinate.

A.3.1 Where the Connecting Map Is Defined

The connecting map does not require the two coordinate maps to have identical images. It is defined on

$$U_\Phi = \{u \in U_R : q_R(u) \in X_{R,f,\eta}\}.$$

In other words, the physical point $q_R(u)$ must also lie in the image of the finite-step coordinate map.

This image condition is different from requiring a mirror-descent or Euclidean update to remain inside its admissible domain. A finite-step update may be well defined even when a particular physical point lies outside the common image of the two coordinate maps. Conversely, the two coordinate systems may overlap even though a later update leaves the region on which they were constructed.

We therefore treat image overlap and update admissibility as separate conditions. Numerical examples are given in Appendix C.3.

A.3.2 Connecting Map from the Small-Step Correction

The correction derived in Appendix A.2.3 also defines a corresponding change between the continuous coordinate system and the corrected finite-step coordinate system.

The correction construction gives the inverse map

$$\widehat{p}_{R,f,\eta} = p_R + \eta a_{R,f}.$$

Composing this corrected inverse map with the continuous coordinate map gives

$$\widehat{\Phi}_{R,f,\eta} = (p_R + \eta a_{R,f}) \circ q_R.$$

Since

$$p_R \circ q_R = \mathrm{id},$$

we obtain

$$\widehat{\Phi}_{R,f,\eta} = \mathrm{id} + \eta\, a_{R,f} \circ q_R.$$

Using

$$a_{R,f} = -z \circ p_R,$$

we may also write

$$\widehat{\Phi}_{R,f,\eta} = \mathrm{id} - \eta z.$$

These equations define the connecting map associated with the first-order corrected coordinate system. They do not imply that every exact connecting map has the same first-order expansion, because exact finite-step coordinate maps need not be unique.

The exact connecting map is still defined by

$$\Phi_{R,f,\eta} = q_{R,f}(\eta, \cdot)^{-1} \circ q_R$$

whenever a particular exact finite-step coordinate map has been selected and the two coordinate-map images overlap.

As η changes, $\Phi_{R,f,\eta}$ describes how the finite-step coordinate system differs from the continuous one. The experiments in Appendix C.3 examine this relation for exact scalar examples, including how much the finite-step coordinates differ from the continuous coordinates and where the exact connecting map is defined.

A.4 Changing the Objective or Regularizer

The previous sections keep the regularizer R and objective f fixed and ask how the coordinate map changes. We now reverse the question. Suppose a coordinate map is given. Can we change the objective, the regularizer, or both so that the same map describes a different form of equivalence?

There are two natural directions. Starting from a continuous coordinate map q_R, we can ask whether changing the objective or regularizer makes the same map satisfy the finite-step equation. Conversely, starting from a finite-step coordinate map, we can ask whether another regularizer makes the same map a continuous coordinate map.

These two directions are conceptually analogous to modified- and inverse-modified-equation viewpoints in numerical analysis, where one relates a numerical scheme to a modified underlying differential equation, or seeks an underlying equation whose discretization reproduces prescribed dynamics (Hairer et al., 2006; Chartier et al., 2007; Zhu et al., 2024).

More broadly, both directions have the form of inverse problems: the coordinate map is treated as objective, and one reconstructs the underlying objective, regularizer, or both so that the desired equivalence holds. This viewpoint is analogous in spirit to inverse formulations in imaging (Bertero & Boccacci, 1998).

A.4.1 Making a Continuous Coordinate Map Work at Finite Step

Let

$$Q = q_R$$

be a continuous coordinate map. We first keep the regularizer R fixed and ask whether changing only the objective can make Q satisfy the finite-step equation.

This is not always possible in a nontrivial way. A simple example is the entropy regularizer

$$R(x) = x \log x,$$

whose continuous coordinate map can be written, on the positive branch and after choosing the origin and orientation, as

$$Q(u) = \frac{u^2}{4}.$$

Let $\widetilde{f}$ be a modified objective and write

$$g = \widetilde{f}'(x), \qquad z = \eta g.$$

Requiring the same map Q to give exact finite-step equivalence reduces to

$$\left(1 - \frac{z}{2}\right)^2 = e^{-z}, \qquad 1 - \frac{z}{2} > 0.$$

Taking the positive square root gives

$$1 - \frac{z}{2} = e^{-z/2}.$$

The exponential lies strictly above its tangent except at $z = 0$. Hence

$$z = 0,$$

and therefore

$$\widetilde{f}'(x) = 0.$$

Thus, on a connected interval, changing only the objective produces only a constant objective in this example. This shows that changing the objective alone is not always sufficient to obtain a nontrivial finite-step equivalence.

Allowing the regularizer to change gives more freedom. Keep Q and the original objective f fixed, and define the physical state produced by one Euclidean step through Q:

$$F_Q(x) = Q\big(Q^{-1}(x) - \eta Q'(Q^{-1}(x)) f'(x)\big).$$

We seek a modified regularizer $\widetilde{R}$ whose mirror step produces exactly the same state $F_Q(x)$. Writing

$$H = \widetilde{R}',$$

this requirement becomes

$$H(F_Q(x)) - H(x) = -\eta f'(x), \qquad H'(x) > 0.$$

This equation is both necessary and sufficient on an interval where all quantities are defined. If H satisfies it and is strictly increasing, then H^{-1} gives the mirror update generated by $\widetilde{R}$.

The modified regularizer need not be unique. For the entropy example with $f'(x) = a > 0$, define

$$\chi_\eta = \left(1 - \frac{\eta a}{2}\right)^2, \qquad \alpha = -\frac{\eta a}{\log \chi_\eta}.$$

Then

$$H(x) = \alpha \log x$$

solves the difference equation. Moreover,

$$H_\varepsilon(x) = \alpha \log x + \varepsilon \sin\left(\frac{2\pi \log x}{\log \chi_\eta}\right)$$

has the same difference along

$$F_Q(x) = \chi_\eta x,$$

because the added term is unchanged when x is multiplied by χ_η. For sufficiently small $|\varepsilon|$, $H'_\varepsilon(x) > 0$. Thus distinct strictly convex modified regularizers can produce the same finite-step equivalence.

A.4.2 Making a Finite-Step Coordinate Map Continuous

We now consider the reverse direction. Suppose

$$Q = q_{R,f}(\eta, \cdot)$$

is a finite-step coordinate map. Can Q also be the continuous coordinate map of another regularizer $\widetilde{R}$?

In one dimension, the answer is immediate. Continuous equivalence for Q requires

$$Q'(u)^2\, \widetilde{R}''(Q(u)) = 1.$$

Setting $x = Q(u)$ gives

$$\widetilde{R}''(x) = \left[Q'(Q^{-1}(x))\right]^{-2}.$$

If $Q' > 0$, the right-hand side is positive. Integrating twice therefore gives a strictly convex regularizer $\widetilde{R}$, unique up to an affine function.

The higher-dimensional problem is more restrictive. For a diffeomorphism

$$Q : U \to X,$$

the continuous coordinate equation determines the candidate metric

$$G(x) = DQ(Q^{-1}(x))^{-T} DQ(Q^{-1}(x))^{-1}.$$

This matrix is positive definite, but a mirror regularizer requires more: G must be the Hessian of a scalar function.

Locally, this requires the compatibility conditions

$$\partial_k G_{ij} = \partial_j G_{ik}.$$

These conditions need not hold for an arbitrary coordinate map.

For example, consider

$$Q(u_1, u_2) = \left(u_1,\, u_2 + u_1^2\right).$$

The candidate metric is

$$G(x) = \begin{pmatrix} 1 + 4x_1^2 & -2x_1 \\ -2x_1 & 1 \end{pmatrix}.$$

But

$$\partial_{x_2} G_{11} = 0, \qquad \partial_{x_1} G_{12} = -2,$$

so the compatibility condition fails. No scalar regularizer can therefore have G as its Hessian, even locally.

In one dimension, an increasing coordinate map therefore determines a continuous mirror geometry directly through the scalar formula above. In higher dimensions, the corresponding metric must also satisfy the Hessian compatibility conditions.

These constructions change the underlying optimization problem. They are therefore different from the exact finite-step equivalence studied in Appendix A.2, where the original R and f remain fixed.

B Numerical Construction and Operator Learning

Appendix A describes the coordinate maps mathematically. We now explain how they are computed and represented in the numerical experiments, and how a single model is trained across a family of regularizers or objectives. We first describe the independent numerical solutions used for evaluation. We then give the common invertible representation of the coordinate maps, the operator-learning models, and the training and evaluation criteria.

B.1 Computing Numerical Reference Maps

To evaluate a learned coordinate map, we need a numerical solution that is computed independently of the learned model. We call this a numerical reference map. The reference construction depends on the problem: continuous coordinate maps can be obtained directly from the mirror geometry, whereas exact finite-step maps require solving the finite-step equation for the specified regularizer, objective, and step size.

These reference solutions are used for evaluation, not as prediction inputs. Equation-trained models therefore do not have access to the reference coordinate maps at test time.

B.1.1 Continuous Coordinate Maps

In one dimension, the inverse continuous coordinate map follows directly from

$$p_R'(x) = \sqrt{R''(x)}.$$

We therefore compute p_R by numerical quadrature and obtain q_R by monotone inversion. Analytic examples provide additional comparisons when closed-form expressions are available.

In the coupled two-dimensional family, we use the flat geometry established in Appendix A.1.2. Let Γ_i denote the matrices of the Levi-Civita connection of the metric M. For a Hessian metric,

$$\Gamma_i = \frac{1}{2} M^{-1} \partial_i M.$$

We first transport a matrix field B by

$$\partial_i B = B\Gamma_i, \qquad B(x_0) = B_0, \qquad B_0^T B_0 = M(x_0).$$

The inverse coordinate map is then obtained from

$$\partial_i p_R = Be_i, \qquad p_R(x_0) = 0.$$

On a simply connected flat region, the transport is path independent and the resulting map satisfies

$$Dp_R^T Dp_R = M.$$

We use this construction independently for each metric in the family. It is a per-problem numerical solver, not part of the learned operator.

The Levi-Civita connection is standard in Riemannian geometry. Here it gives the differential equations for transporting the matrix field B satisfying $B^T B = M$, from which p_R is recovered (Duistermaat, 2001; Shima, 2007).

Numerical accuracy is assessed in two complementary ways. We recompute the map at finer numerical resolutions and compare the resulting maps and derivatives. We also integrate the matrix and coordinate equations along different paths. Agreement between these computations indicates that the numerical reference is well resolved and consistent with the flat geometry. These comparisons measure numerical consistency; they do not by themselves prove global injectivity on an arbitrary domain.

B.1.2 Exact Finite-Step Coordinate Maps

For a fixed regularizer R, objective f, and step size η, an exact finite-step reference map must satisfy the same finite-step equation used by the learned model. We compute these references independently of the operator learner and use them only for evaluation.

In one dimension, it is convenient to work with the inverse coordinate map

$$p = q^{-1}$$

and its derivative

$$w = p' > 0.$$

Define the mirror update

$$M_\eta(x) = (R')^{-1}\left(R'(x) - \eta f'(x)\right).$$

Exact finite-step equivalence is then equivalent to

$$p(M_\eta(x)) = p(x) - \frac{\eta f'(x)}{w(x)}.$$

Since

$$p(x) - p(M_\eta(x)) = \int_{M_\eta(x)}^{x} w(t)\,dt,$$

the same equation can be written as

$$w(x) \int_{M_\eta(x)}^{x} w(t)\,dt = \eta f'(x).$$

Conversely, a positive w satisfying this equation, together with one value of p, determines an increasing inverse coordinate map that satisfies the original finite-step equation.

We therefore construct scalar finite-step references by solving for a positive function w. The additive constant in p is fixed by prescribing

$$p(x_a) = u_a.$$

Once w has been obtained, p follows by integration and q by monotone inversion.

Positivity of w guarantees that p is increasing, but it does not by itself establish that the finite-step equation has been solved accurately. We therefore evaluate the original same-η equation independently after constructing the map. We also compare solutions obtained with richer numerical representations and retain only references whose map, derivative, and equation residuals are stable under refinement.

The exact scalar family used for operator learning is deliberately restricted so that these references can be computed reliably across varying regularizers, objectives, states, and step sizes. This provides accurate targets for evaluating family-level learning without assuming that every finite-step problem admits a global exact coordinate map.

B.1.3 First-Correction Fields

The coupled two-dimensional small-step experiments do not require exact finite-step coordinate maps. Instead, they use the first correction derived in Appendix A.2.3.

Let

$$p_R = q_R^{-1}, \qquad B = Dp_R, \qquad M = \nabla^2 R, \qquad F = -M^{-1}\nabla f.$$

In physical coordinates, the first inverse-map correction $a_{R,f}$ satisfies

$$\big(Da^T B + B^T Da\big)F = \frac{1}{2}B^T D^2 p_R[F,F] = \frac{1}{4}\nabla^3 R[F,F].$$

A boundary value for a on a noncharacteristic curve selects the particular correction used in the experiment. In our two-dimensional family we impose

$$a(0, x_2) = 0.$$

We solve the correction equation numerically by polynomial collocation. No exact finite-step coordinate maps are used as labels in this construction.

The corrected inverse map is then

$$\widehat{p}_{R,f,\eta} = p_R + \eta a_{R,f}.$$

Before using it in trajectory comparisons, we verify that the derivative remains nonsingular on the declared domain and that the required source points lie in its image. These are separate questions from the accuracy of the correction equation itself.

As with the continuous references, numerical accuracy is assessed independently of the learned operator. We solve the correction equation at increasing polynomial orders and compare the resulting fields and derivatives. For selected cases, we also compare independent constructions of the underlying continuous map. These comparisons establish that the numerical reference fields are well resolved; they do not turn the first-order correction into an exact finite-step map.

B.2 Representing the Coordinate Maps

The numerical methods above produce functions, but the learned models must represent those functions with finitely many parameters. We choose the representation so that the geometric constraints are enforced by construction whenever possible.

For the coupled two-dimensional continuous problem, we represent the inverse coordinate map p_R and enforce its point, derivative, and invertibility conditions directly. The first-correction field uses the same spatial basis but is not itself required to be invertible. For the scalar exact finite-step problem, monotonicity is enforced by representing the positive derivative of the inverse map.

B.2.1 Coupled Two-Dimensional Coordinate Maps

For the two-dimensional continuous problem, we represent the inverse coordinate map rather than q_R itself. Let x_0 be the reference point and define

$$B_0 = M(x_0)^{1/2}.$$

We write

$$p_\theta(x) = B_0 \left[x - x_0 + r_\theta(x)\right],$$

where r_θ is represented by a tensor-product cubic spline.

The point and derivative normalization are imposed through

$$r_\theta(x_0) = 0, \qquad Dr_\theta(x_0) = 0.$$

Hence

$$p_\theta(x_0) = 0, \qquad Dp_\theta(x_0) = B_0.$$

These are the same normalization conditions used to identify the numerical reference map.

To make the output map invertible on the whole physical box, we constrain the nonlinear part to satisfy

$$\sup_X \|Dr_\theta(x)\|_2 < \kappa, \qquad 0 < \kappa < 1.$$

The spline coefficients are transformed deterministically so that this bound holds for every model output. Therefore

$$\sigma_{\min}(Dp_\theta) \geq \sigma_{\min}(B_0)(1 - \kappa) > 0.$$

On a convex domain, the map

$$x \mapsto x + r_\theta(x)$$

is injective because it is a Lipschitz perturbation of the identity with Lipschitz constant below one. Thus p_θ is a diffeomorphism onto its image, and

$$q_\theta = p_\theta^{-1}$$

is well defined there.

This guarantees invertibility of each predicted map on the physical domain. It does not guarantee that its image contains every source point requested in an experiment. We therefore verify source-domain coverage separately.

The operator-learning model does not predict $p_\theta(x)$ independently at every query point. Instead, it predicts the spline coefficients defining r_θ. The same continuous map can then be evaluated, differentiated, and inverted at arbitrary points inside the declared domain.

B.2.2 Representing the First Correction

The first correction $a_{R,f}$ is a vector field, not a coordinate map. It therefore does not need to be invertible by itself.

We represent a_θ with the same tensor-product cubic spline basis used for the continuous inverse map. The boundary condition from Appendix A.2.3 is imposed directly in the coefficients. For the coupled two-dimensional family,

$$a_\theta(0, x_2) = 0.$$

The operator therefore predicts only the remaining free spline coefficients.

Given a continuous inverse coordinate map p_R, the corrected inverse map is

$$\widehat{p_\theta} = p_R + \eta a_\theta.$$

It is this sum, rather than a_θ alone, that must remain invertible.

We therefore control the derivative of the correction and verify nonsingularity of $D\widehat{p_\theta}$ on the declared evaluation domain. We also verify that the requested source points lie in the image of $\widehat{p_\theta}$. These conditions are evaluated separately from the residual of the correction equation.

This representation lets every operator architecture predict the same correction field. Differences between FNO (Li et al., 2021), DeepONet (Lu et al., 2021), MLP (Hornik et al., 1989), and reduced-basis models therefore come from how the input functions are encoded, not from using different output geometries.

B.2.3 Scalar Exact Finite-Step Maps

The scalar exact finite-step problem uses a different representation. Here invertibility is equivalent to monotonicity, so we represent the positive derivative of the inverse coordinate map

$$w(x) = p'(x) > 0$$

rather than p itself.

Once w is known, the inverse map is obtained from

$$p(x) = u_a + \int_{x_a}^{x} w(t)\, dt,$$

and the finite-step coordinate map follows by monotone inversion:

$$q = p^{-1}.$$

Positivity of w therefore guarantees that every represented map is invertible.

In the exact scalar benchmark, w is represented by a positive mixture of smooth basis functions. The coefficients are constrained to be positive, and the representation is normalized at the reference point. The resulting p is increasing for every model output, so invertibility is structural rather than imposed through a penalty.

The step size η enters the same representation. Thus a model represents the joint dependence of the finite-step coordinate map on the physical variable and the step size, rather than fitting unrelated maps at separate step sizes.

At $\eta = 0$, we train using the continuous compatibility residual obtained from the normalized finite-step limit. The same η-conditioned representation is evaluated at $\eta = 0$, but agreement with the continuous coordinate map is learned rather than enforced by the architecture.

This positive-derivative representation is specific to the one-dimensional benchmark. In higher dimensions, positivity of a scalar derivative is no longer sufficient, which is why the coupled two-dimensional experiments use the structured inverse-map representation above.

B.3 Operator-Learning Models

The previous section specifies the output representation. We now describe the models that map the input functions to the coefficients of that representation. The models differ in how they process the input fields, but within each task they use the same output representation and the same physical information (for function-space operator learning, see Chen & Chen, 1995; Kovachki et al., 2023).

For the coupled two-dimensional continuous problem, the inputs are the sampled components of the mirror metric M. For the first-correction problem, the sampled objective gradient ∇f is added (for multiple-input operator learning, see Jin et al., 2022). The hidden parameters used to generate the functions are never provided to the models.

B.3.1 COUPLED TWO-DIMENSIONAL MODELS

We compare the Fourier neural operator (FNO), DeepONet, a multilayer perceptron (MLP) that receives the same sampled fields, PCA-Net (Bhattacharya et al., 2021), and POD-DeepONet (Lu et al., 2022).

In all cases, the model predicts the free coefficients of the representation in Appendix B.2; the coordinate map or correction field is then obtained from the common decoder. Thus the comparison concerns how the input functions are encoded, not different choices of output geometry.

The Fourier neural operator treats the sampled inputs as spatial fields (for approximation theory, see Kovachki et al., 2021). Each layer combines a Fourier convolution with a local pointwise transformation, so the model can use both long-range and local spatial structure. The final feature field is converted to the coefficients of the common spline representation.

In the continuous problem, the FNO receives the metric components together with spatial coordinates. In the correction problem, the components of ∇f are added as input channels. The architecture itself does not receive the parameters used to generate either R or f.

DeepONet represents the input function and the output coordinates separately (for approximation and error analysis, see Lanthaler et al., 2022). Its branch network encodes the sampled physical fields, while its trunk network represents the spatial location at which output coefficients are evaluated. Their outputs are combined to produce the coefficients of the same spline representation used by the other models.

This gives DeepONet a continuous coordinate-query structure, whereas the FNO processes the input primarily through its spatial grid. Both models receive the same physical fields.

The MLP provides a finite-dimensional baseline. It flattens the same sampled fields into a vector and maps this vector directly to the free decoder coefficients.

Unlike FNO and DeepONet, it does not explicitly preserve the spatial organization of the input field. It is nevertheless an important comparison because the sensor grid is fixed during training, so the functional input can also be viewed as a finite-dimensional vector.

The remaining two models introduce a reduced representation of the solution space (Hesthaven & Ubbiali, 2018). PCA-Net first compresses the sampled input using principal components, applies a nonlinear map between the reduced input and output coordinates, and reconstructs the output from a fixed reduced basis (Lanthaler, 2023). POD-DeepONet uses a learned branch network together with a fixed reduced output basis.

These two models therefore use additional information that the FNO, DeepONet, and MLP do not use in equation-only training: their output bases are constructed from solutions of training problems. We report them as useful reduced-basis comparisons, but not as models with exactly the same information as the first three. No basis or normalization is fitted using validation or test functions.

The five models are chosen so that differences in prediction quality can be attributed primarily to the treatment of the input functions. Within each task, they predict the same type of output coefficients and are evaluated through the same coordinate-map or correction-field representation.

Exact widths, ranks, Fourier modes, and parameter counts are summarized in Table 2. We keep these implementation details in one table rather than repeating them in the text.

B.3.2 SCALAR EXACT FINITE-STEP MODELS

The one-dimensional exact finite-step benchmark uses the positive-derivative representation from Appendix B.2.3. Here the operator-learning model predicts the parameters of the positive function

$$w(x) = p'(x) > 0,$$

together with its dependence on the step size η (Table 3). The inverse coordinate map is obtained by integration, and the finite-step coordinate map by monotone inversion.

We compare FNO, DeepONet, and MLP using the same sampled representation of the regularizer and objective. All three models use the same positive decoder, so every predicted scalar map is increasing by construction. The hidden coefficients used to generate the functional parts of the

Table 2: Coupled two-dimensional architectures. Counts are trainable real parameters; complex Fourier parameters count twice. Width/depth describe hidden layers. All hidden activations are GELU, with no dropout. Only PCA-Net and POD-DeepONet use an output basis constructed from training solutions; these bases and normalization buffers are not trainable parameters. Parameter counts do not imply equal approximation power.

Model	Continuous	Correction	Settings (continuous / correction)
FNO	114,266	114,314	Width 24; 3 blocks; 4 modes; head $24 \to 64 \to 2$
DeepONet	121,850	125,370	Width 72 / 56, depth 2 in branch and trunk; rank 32
MLP	123,484	118,908	Width 64 / 48, depth 2
PCA-Net	111,808	115,392	Width 224, depth 3; input rank 16 / 32; output rank 32
POD-DeepONet	116,960	121,552	Width 72 / 56, depth 3; output rank 32

Common output: 16×8-cell cubic splines; 412 / 396 free scalar coefficients; derivative-bound order 8; $\kappa = .85$ / 1; raw coefficient scale .01. Inputs have 5 / 7 channels on 17^2 sensors.

Table 3: Scalar exact finite-step architectures. Counts include the encoder and 173 shared decoder parameters: 24 centers, 24 widths, and 5×25 coefficient weights. Each encoder outputs 125 coefficients. The common decoder has 24 even-Gaussian components, a positive constant term, density floor .9, and degree-four Chebyshev dependence on $\eta \in [0, 1.4]$.

Model	Parameters	Settings
FNO	46,738	Width 24; 3 blocks; 12 modes; GELU; spatial mean and linear $24 \to 125$ head
DeepONet	47,710	Branch $768 \to 48 \to 64 \to 65$; trunk $2 \to 32 \to 32 \to 64$; rank 64; tanh hidden layers
MLP	48,346	$768 \to 48 \to 64 \to 125$; tanh hidden layers

All models receive the same six channels at 128 sensors: sampled r, sampled v, position, and constant channels for λ, a, and β. The Fourier coefficients generating r and v are not inputs. The anchor is $p(.3025) = 1.1$.

regularizer and objective families are not provided as inputs. The step size η enters the same model, allowing one model to represent a range of step sizes rather than fitting separate models for separate values of η.

The positive decoder uses a fixed collection of smooth basis functions with positive amplitudes and a positive constant term. This restricts the represented class of monotone maps, but it guarantees invertibility for every model output. The same decoder is used by all three architectures, so their comparison again concerns how the input functions are encoded rather than a change in the output representation.

In this scalar setting, the input functions are observed on a fixed one-dimensional grid and the final output is a comparatively low-dimensional set of decoder coefficients. For this reason, the MLP is a particularly strong baseline rather than merely a weak control. The numerical results in Appendix C.2.1 (Table 5) compare the three architectures on the same family-level task.

B.4 Training and Evaluation

The primary models are trained from the equations that define the desired coordinate maps or corrections (for equation-based operator learning, see Li et al., 2024; Wang et al., 2021). Numerical reference maps from Appendix B.1 are used for evaluation, except in the explicitly supervised comparisons.

B.4.1 Equation-Based and Supervised Training

For the continuous two-dimensional problem, let

$$B_\theta = Dp_\theta, \qquad W = M^{-1/2}.$$

We minimize

$$\mathcal{L}_{\mathrm{cont}} = \left\langle \frac{1}{d} \left\| W(B_\theta^T B_\theta - M)W \right\|_F^2 \right\rangle_X ,$$

where $\langle\cdot\rangle_X$ denotes numerical integration over the physical domain. The normalization by $M^{-1/2}$ makes the residual dimensionless across metrics of different scales.

For the first-correction problem, with

$$B_R = Dp_R, \qquad F = -M^{-1}\nabla f,$$

the residual is

$$\mathcal{R}_{\mathrm{corr}} = (Da_\theta^T B_R + B_R^T Da_\theta)F - \frac{1}{4}\nabla^3 R[F,F].$$

Training minimizes its squared physical norm, normalized by the squared norm of the right-hand side for each input. The required continuous coordinate map and metric derivatives are computed independently; first-correction reference fields are not used in this loss.

For the scalar exact finite-step problem, training uses the inverse equation from Appendix B.1.2. For $\eta > 0$,

$$w(x) \int_{M_\eta(x)}^{x} w(t)\, dt = \eta f'(x),$$

and at $\eta = 0$ we use its continuous extension. The mirror update M_η is computed independently of the predicted coordinate map.

For comparison, we also train the same architectures directly on the numerical reference solutions. These supervised models use the same output representation but minimize relative physical H^1 error to the reference fields. They provide a comparison between learning from the defining equation and learning directly from numerical solutions.

PCA-Net and POD-DeepONet differ from the other equation-trained models in one important respect: their output bases are constructed from numerical solutions of the training problems. FNO, DeepONet, and MLP do not use such solution-derived output bases during equation training.

B.4.2 Optimization and Model Selection

For the coupled two-dimensional experiments, all architectures within a task use the same training and validation functions and the same minibatch sequence within each independent run. We use AdamW (Loshchilov & Hutter, 2019) with batch size eight, an initial learning rate of 10^{-3}, weight decay 10^{-6}, and cosine learning-rate decay. Continuous-coordinate models are trained for at most 4000 updates, while the correction models use at most 16000.

We evaluate the validation objective periodically during training and select the earliest parameter state attaining its minimum. Training stops after ten consecutive validation evaluations without improvement. Three independent initializations are used for each coupled-model comparison. Test functions are not used for model selection, and no per-problem optimization is performed at test time.

The scalar exact finite-step experiment uses a separate training procedure adapted to its one-dimensional representation. Training and validation problem families are generated independently, and all compared architectures receive the same sampled inputs and minibatches. Model selection again uses only the finite-step equation on the validation family. The step size η is an input to the model, so a single model represents multiple step sizes (Table 4).

B.4.3 Accuracy Metrics

We evaluate coordinate maps in physical units. The L^2 error measures errors in function values, while the H^1 error also includes first derivatives. We report both because the defining equations depend directly on the Jacobian of the coordinate map.

We also evaluate the defining equation independently of the map error. For continuous coordinates this is the compatibility residual

$$Dp_\theta^T Dp_\theta - M;$$

Table 4: Training and model selection. Populations count functions or problem pairs. The same minibatches are used across architectures within each independent run. Scalar models use Adam; coupled models use AdamW. Validation includes initialization, and the earliest minimum selects the parameters.

Setting	2D continuous	2D correction	Scalar exact finite-step
Training population	96	384	128
Validation population	24	24	32
Batch size	8	8	8
Maximum updates	4,000	16,000	4,000
Validation interval (updates)	100	400	100
Independent initializations	3	3	5
Initial learning rate	10^{-3}	10^{-3}	.003
Final cosine learning rate	5×10^{-5}	5×10^{-5}	.0003
Weight decay	10^{-6}	10^{-6}	0
Early-stop patience (checks)	10	10	None
Input sensors	17^2	17^2	128
Equation sites	33^2	33^2	65 per η
Validation sites	65^2	65^2	73 per η

Scalar setting	Values
Training η	$0, .1, .4, .8, 1.1, 1.4$
Validation η	$0, .05, .25, .6, 1, 1.4$
Physical collocation	$x = s^2$, with 65 or 73 cosine-spaced $s \in [10^{-4}, \sqrt{2}]$
Source evaluation	129 points in $u \in [.2, 2]$
Independent-reference η	$0, .4, 1.4$
Reference degree pairs	$40/64$, then $96/128$, then $128/192$ for failed comparisons

for the first correction it is the residual above; and for exact finite-step maps it is the original same-η equation.

B.4.4 Invertibility, Coverage, and Numerical References

We evaluate invertibility and source-image coverage separately from approximation error. For the continuous representation, invertibility follows from the derivative bound in Appendix B.2.1. For corrected maps $p_R + \eta a_\theta$, we verify nonsingularity and source-image coverage numerically on the declared evaluation domain.

Numerical references are used in error summaries only when they satisfy the prescribed validity and numerical-refinement criteria. When some references are unavailable, we report both the requested and available population sizes.

C EXPERIMENTS AND ADDITIONAL RESULTS

This appendix gives the numerical results supporting the four questions in the main paper. We follow the same order as the main text: continuous coordinate learning, finite-step coordinate learning and correction, the relation between continuous and finite-step coordinates, and finally modified objectives and regularizers.

C.1 CONTINUOUS COORDINATE MAPS

C.1.1 COUPLED TWO-DIMENSIONAL OPERATOR LEARNING

We first evaluate the operator

$$M \longmapsto p_R$$

on the coupled two-dimensional family. The training set contains 96 metrics. We evaluate on three groups of 66 unseen metrics: an in-distribution group, a higher-frequency group with 32 spatial modes rather than the 4, 8, and 16 modes used in training, and a geometry-scale group with larger variation in the metric. All 198 numerical reference maps satisfy the prescribed refinement criteria.

At the native 17×17 input grid, the equation-trained FNO gives mean relative H^1 errors

$$1.203 \times 10^{-3}, \qquad 1.800 \times 10^{-3}, \qquad 3.006 \times 10^{-3}$$

on the in-distribution, higher-frequency, and geometry-scale groups, respectively. The corresponding MLP errors are

$$2.985 \times 10^{-3}, \qquad 3.509 \times 10^{-3}, \qquad 5.923 \times 10^{-3}.$$

Among the five equation-trained models, FNO has the lowest mean H^1 error in all three groups.

Direct supervision improves the FNO further on the in-distribution group, reducing its relative L^2/H^1 errors to

$$6.81 \times 10^{-4}/7.97 \times 10^{-4}.$$

Because p_R contains a large near-identity component, full-map relative errors are smaller than errors measured relative only to the nonlinear displacement. For equation-trained FNO, the displacement-relative L^2 errors are

$$8.51\%, \qquad 10.31\%, \qquad 14.44\%$$

across the three groups. We therefore report both full-map and deformation-sensitive errors when interpreting accuracy.

Figure 5 gives the full comparison.[1]

[1] Figure 2 uses the first zero-based ID test input and the first prescribed independent initialization (70), with the equation-trained FNO and the 17×17 input grid. This rule was fixed before viewing spatial arrays; no error ranking or visual screening was used.

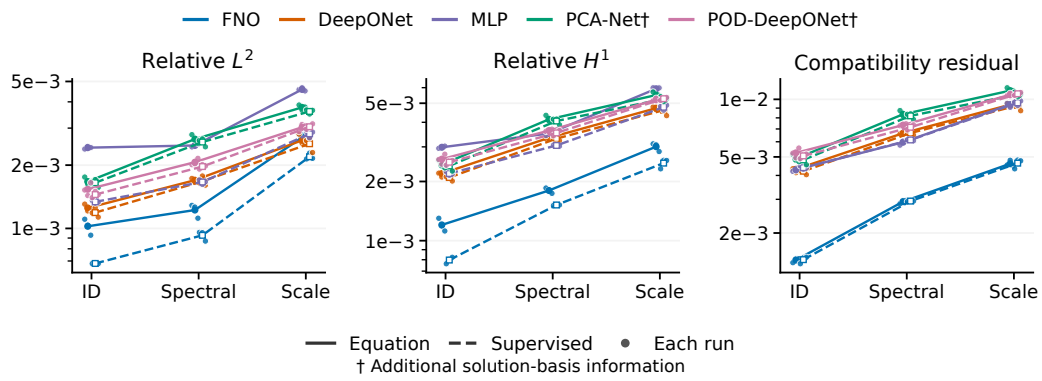


Figure 5: Continuous coordinate maps on unseen coupled geometries. All three independent training-run means are shown, each using 66 inputs per group. Equation-trained and supervised models use the same sampled fields and invertible spline representation. PCA-Net and POD-DeepONet also use output bases constructed from training solutions. Relative errors measure the full inverse coordinate map and its physical derivatives. Compatibility is an independently evaluated inverse-metric RMS residual, distinct from the training loss.

C.1.2 Effect of Input Resolution

We next evaluate the same trained models using finer samples of the input metric on the same physical domain. No model parameters are updated.

For equation-trained FNO, the in-distribution relative H^1 errors at 17, 25, 33, and 65 input points per axis are

$$0.001203, \qquad 0.001406, \qquad 0.001205, \qquad 0.001212.$$

Thus the same FNO remains accurate when the input field is sampled more finely, although the error does not decrease monotonically with resolution.

The other models are trained on a fixed input grid and are evaluated at finer resolutions after resampling the inputs to that grid. Their results therefore do not test native resolution transfer in the same sense.

Figure 6 compares both training objectives and both tasks.

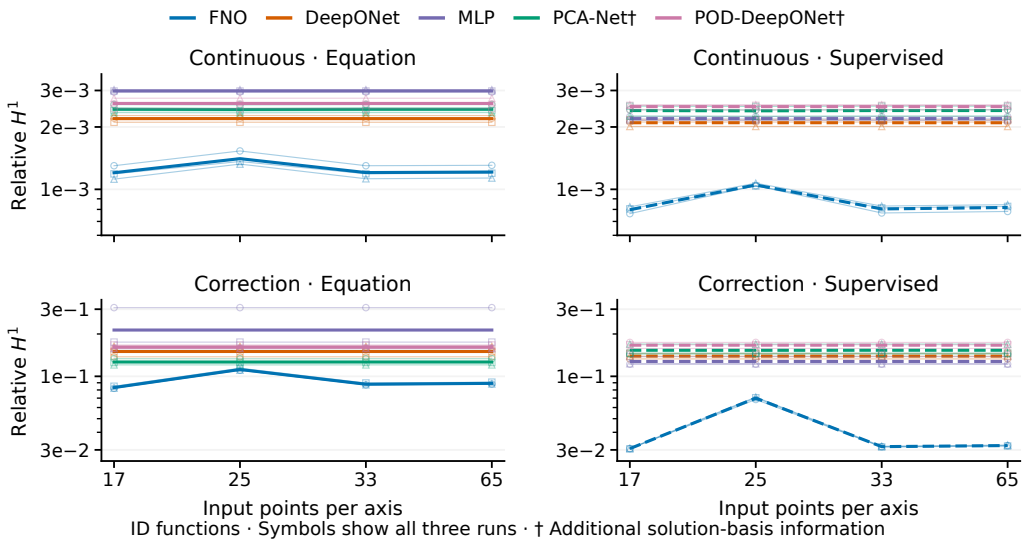


Figure 6: Same-weight resolution transfer on the in-distribution groups for continuous coordinate maps and first-correction fields. All three independent training runs are shown. FNO accepts each input grid directly; the other models resample to their training grid. The same physical domain and output evaluation grid are used throughout. Flat resampled curves do not demonstrate use of additional input information.

C.2 Finite-Step Coordinate Maps

C.2.1 Exact Finite-Step Operator Learning in One Dimension

We compare FNO, DeepONet, and MLP on the same family of exact scalar finite-step problems. Table 5 reports the primary-resolution results. The MLP gives the lowest ID equation, map, and derivative errors, while the relative ordering changes under distribution shift. All three models use the same positive-derivative output representation.

Distribution	Model	E RMS	q RMSE	Relative q' RMS
ID	FNO	6.555 ± 0.174	0.715 ± 0.032	3.392 ± 0.128
ID	MLP	5.634 ± 0.201	0.623 ± 0.104	3.193 ± 0.257
ID	DeepONet	7.869 ± 0.230	0.896 ± 0.012	4.501 ± 0.094
Shape shift	FNO	7.386 ± 0.089	0.879 ± 0.067	4.133 ± 0.203
Shape shift	MLP	6.523 ± 0.215	0.754 ± 0.081	3.969 ± 0.141
Shape shift	DeepONet	7.952 ± 0.241	0.826 ± 0.095	4.778 ± 0.121
Scale shift	FNO	7.220 ± 0.223	0.593 ± 0.075	3.307 ± 0.231
Scale shift	MLP	7.604 ± 0.423	0.796 ± 0.116	3.899 ± 0.294
Scale shift	DeepONet	9.117 ± 0.559	0.990 ± 0.182	4.743 ± 0.618

Table 5: Primary scalar exact finite-step operator-learning results. All three models use the same sampled problem inputs and positive-derivative output representation. All values are in 10^{-3} units and give the mean $\pm$ sample standard deviation across five independent training runs, each using the same 32 unseen functions per distribution and 128 input points. RMS values pool 129 source sites in $[.2, 2]$ and the specified function–step cases. Equation RMS normalizes the mapped-step discrepancy by the mirror-step displacement and pools $\eta = .05, .25, .6, 1, 1.4$. Map RMSE and relative derivative RMS use only $\eta = 1.4$, with independently computed reference maps and their derivatives; the relative derivative error is $q'_{\mathrm{pred}}/q'_{\mathrm{ref}} - 1$.

C.2.2 First-Correction Learning in Coupled Two Dimensions

We next learn the operator

$$(M, \nabla f) \longmapsto a_{R,f}$$

from the first-correction equation in Appendix A.2.3. Unlike the continuous problem, both the geometry and the objective vary across the family.

On the in-distribution group, equation-trained FNO gives relative L^2/H^1 errors

$$0.02768/0.08320,$$

while direct supervision gives

$$0.01907/0.03063.$$

On the geometry-scale group, the corresponding errors are

$$0.03784/0.10201$$

for equation training and

$$0.02718/0.04745$$

for supervision. FNO has the lowest mean L^2 and H^1 errors among the compared models in these two groups.

The field error and the correction-equation residual need not rank models identically. For example, on the in-distribution group the equation-trained FNO has a mean relative correction-equation residual of 0.02497, compared with 0.02616 for its supervised counterpart, even though supervision gives the smaller H^1 field error.

Figure 7 gives the full comparison.

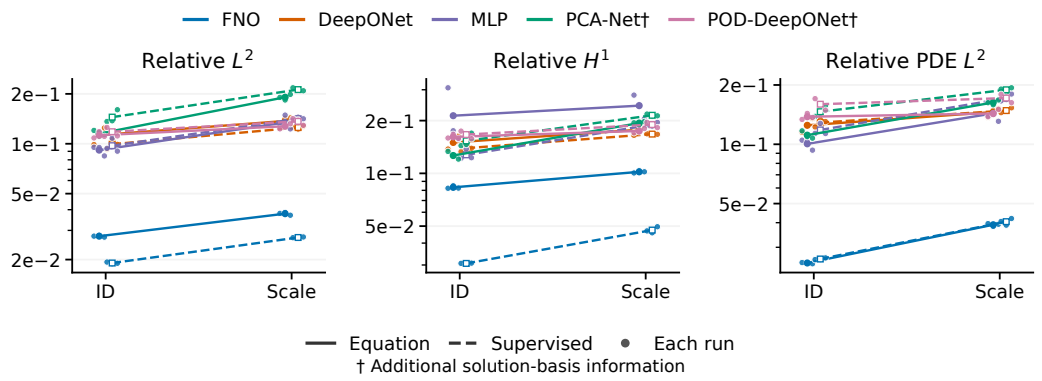


Figure 7: First-correction accuracy on the complete in-distribution and geometry-scale groups, with 66 unseen geometry/objective pairs per group and all three independent training runs. The equation residual is relative PDE L^2. PCA-Net and POD-DeepONet use output bases constructed from training solutions. For the higher-frequency group, one numerical reference remains unresolved, so a complete full-population accuracy summary is unavailable for that group (65 of 66 references available).

C.2.3 Finite-Step Algorithm Agreement

The correction field is useful only if it improves agreement between the two finite-step algorithms. We therefore compare mirror descent with Euclidean gradient descent using either the continuous coordinate map alone or the corrected coordinate map.

At $\eta = 0.02$ over fixed time $T = 0.04$, the mean physical disagreement on the in-distribution group decreases from

$$3.58 \times 10^{-5} \quad \text{to} \quad 6.95 \times 10^{-7},$$

when the learned correction is included. This is a factor of approximately 51.5.

Under the geometry-scale shift, the disagreement decreases from

$$5.52 \times 10^{-5} \quad \text{to} \quad 1.61 \times 10^{-6},$$

a factor of approximately 34.4.

Thus the learned first correction substantially improves finite-step agreement even though its field-level H^1 error is larger than that of the continuous coordinate map.

Figure 8 compares the learned corrections with continuous-coordinate reuse and the independently computed first correction.

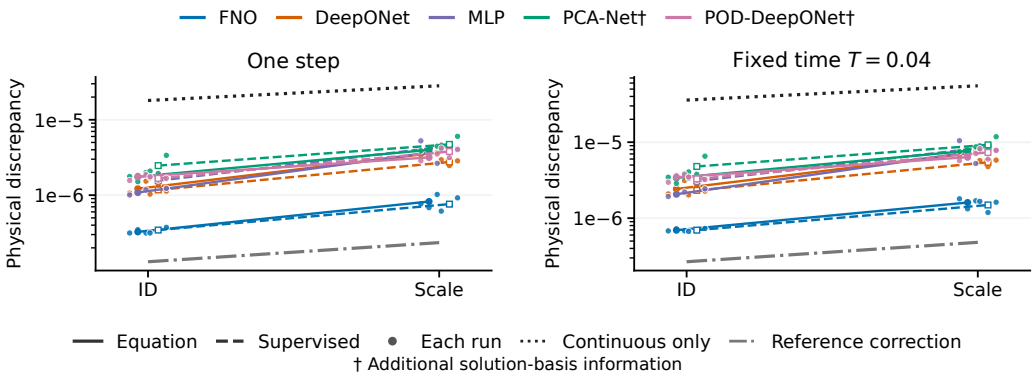


Figure 8: One-step and fixed-time physical disagreement at $\eta = .02$ and $T = .04$. Means of Euclidean endpoint distances use 12 unseen functions and three starting points per group. Both equation-trained and supervised models are shown, with all three independent training runs. Every correction uses an independently computed continuous base map. The independently computed first correction gives smaller disagreement than the learned corrections. These are error reductions, not computational speedups.

C.3 Relation Between the Continuous and Finite-Step Coordinates

C.3.1 Exact Scalar Examples

We first use scalar problems for which both coordinate maps are available in closed form. These examples separate changes in the connecting map from failures of either coordinate system.

For the entropy regularizer

$$R(x) = x \log x$$

with affine objective

$$f(x) = ax, \qquad a > 0,$$

the exact finite-step branch gives the connecting map

$$\Phi_{R,f,\eta}(u) = u_a + c_\eta^{-1/2}(u - u_a), \qquad c_\eta = \frac{2(1 - e^{-\eta a/2})}{\eta a},$$

with the continuous extension $c_0 = 1$.

Thus $\Phi_{R,f,\eta}$ is affine in u for every η. Both the continuous and finite-step coordinate maps remain diffeomorphisms on their natural intervals for every finite $\eta > 0$, although the slope of the connecting map moves progressively away from one as the step size increases.

This example shows that a large change between the two coordinate systems does not by itself imply a loss of exact finite-step equivalence.

A complementary example is provided by the log-barrier regularizer.[2] There the finite-step branch is admissible only while

$$1 + \eta a > 0.$$

When $a < 0$, reaching $1 + \eta a = 0$ corresponds to the mirror update leaving its domain rather than to a loss of invertibility of the connecting map.

C.3.2 A Nonlinear Connecting Map

The affine examples above are useful controls, but connecting maps need not be affine in general. We therefore also consider

$$R(x) = \lambda x \log x, \qquad f(x) = ax + \frac{\beta}{2}x^2,$$

[2]For $R(x) = -\log x$ and $f(x) = a \log x$, define $k_\eta^2 = \log(1 + \eta a)/(\eta a)$ and $q_\eta(u) = x_a e^{k_\eta(u - u_a)}$. The removable limit is $k_0 = 1$, and $\Phi_\eta(u) = u_a + (u - u_a)/k_\eta$.

for which the continuous and finite-step coordinate maps are computed independently.

For each step size, we form

$$\Phi_{R,f,\eta} = q_{R,f}(\eta, \cdot)^{-1} \circ q_R.$$

No additional model is fitted for $\Phi_{R,f,\eta}$.

Figure 4 shows the displacement

$$\Phi_{R,f,\eta}(u) - u$$

for

$$\eta \in \{0, 0.1, 0.3, 0.7, 1, 1.4\}$$

on the common source interval

$$u \in [0.2, 2].$$

Unlike the affine controls, the displacement varies nonlinearly with u. The observed curvature is substantially larger than the change produced by refining the numerical references, and all displayed connecting maps remain monotone on the common interval.

This example shows that the continuous and finite-step coordinate systems can be related by a genuinely nonlinear change of coordinates even while both coordinate maps remain well defined and invertible on the same region.

C.4 Modified Objectives and Regularizers

Appendix A.4 asks whether a prescribed coordinate map can be reproduced after changing the objective, the regularizer, or both. Here we test these constructions numerically.

In both directions, the learned model predicts the modified problem input. We then solve the corresponding forward coordinate equation independently and compare the reconstructed coordinate map with the prescribed target. The reconstruction step is therefore separate from the model that predicts the modified input.

C.4.1 Making a Finite-Step Map Continuous

We first start from a finite-step target map Q and learn a modified regularizer whose continuous coordinate map reproduces Q.

The test set contains 64 unseen target problems, evaluated after five independent training runs. For every predicted modified regularizer, we independently solve the continuous coordinate equation and compare the reconstructed map with the original target.

All

$$320/320$$

learned predictions reconstruct successfully. The mean per-instance transform RMSE is

$$(1.65 \pm 0.36) \times 10^{-6},$$

where the variation is across independent training runs. A cubic numerical baseline gives

$$3.09 \times 10^{-7}.$$

Thus the learned model produces valid modified regularizers throughout this test set, although the simple cubic baseline is more accurate on these one-dimensional problems.

C.4.2 Making a Continuous Map Work at Finite Step

We next reverse the construction. Starting from a continuous target coordinate map, the model predicts modified finite-step problem inputs, and we solve the finite-step coordinate equation independently.

Among the 320 learned predictions, 294 can be reconstructed numerically and 26 cannot. Repeating those 26 cases indicates that the failures arise from finite-precision limitations of the reconstruction

routine, not from an observed loss of invertibility. We therefore summarize this direction primarily by the reconstruction success count, rather than by an RMSE computed only on the successful cases.

The cubic baseline succeeds for all $64/64$ test problems and has mean transform RMSE

$$1.87 \times 10^{-6}.$$

The two directions are therefore numerically asymmetric in this experiment: reconstructing a continuous regularizer from a finite-step target is reliable across the full test set, whereas reconstructing the finite-step problem from a continuous target remains less robust.

Table 6 summarizes both directions.

Table 6: Independent coordinate-map reconstruction from modified problem inputs. Each direction uses 64 unseen target problems. Neural results use five independent training runs; the uncertainty is the sample standard deviation of run means. RMSE is the mean of per-instance transform RMSE. No successful-cases-only neural RMSE is reported for the incomplete finite-step reconstruction.

Reconstruction	Method	Successful / requested	Transform RMSE
Continuous	Neural	$320/320$	$(1.65 \pm 0.36) \times 10^{-6}$
Continuous	Cubic	$64/64$	3.09×10^{-7}
Finite-step	Neural	$294/320$	—
Finite-step	Cubic	$64/64$	1.87×10^{-6}