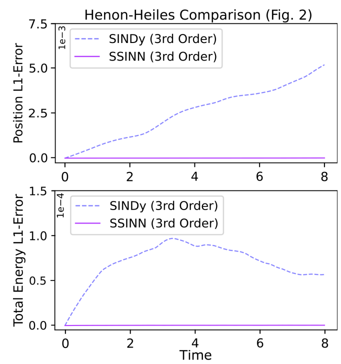
We thank all reviewers for their valuable comments and suggestions. We will release our codebase, create a supplement with pseudocode for our integrator, and make all requested textual changes such as adjusting overly strong wording, clarifying the state-of-the-art, and including all suggested references. Our submission is the first to use a symplectic bias with sparse identification and was described by reviewers as "very sensible" and "of interest to the NeurIPS community." We believe concerns can be addressed within the review cycle via textual changes and further results.

Comparison to SINDy (R2, R3) We have obtained and will incorporate a comparison to SINDy, along with its experimental details. SINDy may be applied by learning a system of ODEs, constructing a Hamiltonian, and predicting with a symplectic integrator if the Hamiltonian is separable. There were no instances where SINDy outperformed our approach; the SINDy mass-spring Hamiltonian was not even separable. SINDy learned the Henon-Heiles Hamiltonian to 10^{-2} precision (10^{-5} for an SSINN) and the coupled oscillator Hamiltonian to 10^{-1} precision (10^{-6} for an SSINN).



Hamiltonians not in the basis (R2, R3) We will incorporate an additional result into Section 7 using a polynomial basis to approximate the pendulum Hamiltonian. As the cosine term is not present, this SSINN instead converged to the Taylor polynomial of cosine (with the degree depending on the basis used). We assume that Hamiltonians not included in the basis can usually be well-approximated by large polynomial or trigonometric bases. In these instances, sparsity likely remains beneficial, as Taylor polynomials are generally sparse in a polynomial function space. Although simple, this additional example offers some experimental support.

Non-(p, q) data (R1, R2) Rather than incorporating examples with autoencoders, we chose to instead emphasize other common real-world limitations like small and noisy data; many works since Greydanus, et al. have taken a similar approach. We will add lack of non-(p, q) data to Section 8 (Scope and Limitations), along with relevant citations.

Architecture clarifications (R1, R2) SSINNs are presented in Section 3.2 as a single fully-connected linear layer with no bias. This layer learns the necessary terms of the basis, meaning that the trainable parameters become the coefficients and are linear with respect to each term in the basis. However, the SSINN used with Equation 9 in Section 7 is slightly more complicated, including an additional layer with bias for the trainable parameters inside of the sine function (which are learned non-linearly with respect to the model output). We will improve Section 3.2 and Figure 1 by clarifying that, although a single layer with no bias is used for a polynomial basis, more complicated bases (such as Equation 9) may require additional layers with bias terms.

Experimental clarifications (R1) The novel trajectories referenced in the paper were completely held out from the training code and only used for assessing the performance of already trained models. Hyperparameter tuning was slight, consisting of a small grid-search (LR=[10^{-2}, 10^{-3}], L1=[10^{-4}, 10^{-3}]), sometimes followed by additional tweaking of regularization. When referring to convergence to the true governing equation, we generally tried to include the decimal precision of convergence (although we used the right sparsity pattern for Section 6). Training and inference speed is comparable to SRNNs but lags behind SINDy due to substantial overhead from the symplectic integrator. We will clarify these details in the paper. We did not incorporate thresholding but like the idea and will mention it in Section 3.2.

Integrator clarifications (R2) Symplectic integrators do maintain conserved quantities, but small oscillations commonly arise in practice, often from truncation error. These oscillations are more apparent for learned Hamiltonians because their trainable parameters tend to have much longer decimal components than the coefficients in the true Hamiltonians–as a result, they are more strongly affected by truncation error. All experiments used $\Delta t = 0.1$; Figure 2 shows long-term integration results (80 steps). We will clarify this in the final paper and reword line 317.

Drivers of performance (R2) The most important driver of performance increases over SRNNs is the simplicity of the chosen basis (and presumed sparsity). Our SRNNs employed the same symplectic integrator as our SSINNs but were still often outperformed by large margins. That said, RK4-trained SSINN Hamiltonians are heavily degraded due to noise from the less accurate integrator, chaos, and lack of symplectic preservation in training. Henon-Heiles 0.1-step test predictions from an RK4-trained Hamiltonian, even when computed with a symplectic integrator, had an average L1-error 4 orders of magnitude larger than the SI-trained Hamiltonian. We will incorporate this discussion.

Interpretability (R2) Throughout the paper, we refer to interpretability in the sense of being able to discover the true Hamiltonian if it is included in the basis, which is not possible for previous approaches with symplectic biases. For clarity, we will adjust our wording and include a definition of what we mean by black-box approaches.