Replies to Reviewer #1.

an explicit numerical scheme is used in the prediction phase: In the prediction phase, we have employed both the proposed discrete gradient (see "Eq. (6)" in Tables 2-3 in Section 4 and Figs. A2–A4 in Appendix F) and explicit numerical schemes (see also RK and ada.DP). With the discrete gradient method, the system's energy is strictly conserved/dissipated, and with the explicit numerical schemes, as you indicate, it is not; however, the latter would be preferable when the learned models are used in existing physics simulators (e.g., Matlab). One can choose either scheme depending on the situations, and this is one of the advantages of our proposed method.

artificially varying the time-step: We appreciate the very interesting comments. With varying time-step, symplectic integrators (the leapfrog integrator in our study) are known to lose symplecticity and cannot preserve the system energy just like non-symplectic Runge–Kutta methods, while the proposed discrete gradient method can. In particular, depending on the learned parameters, the proposed method can choose the time-step while preserving the energy.

Replies to Reviewer #3.

momentum conservation: A discrete gradient method can have the property of conjugate symplecticity, which guarantees the preservation of the momentum with high accuracy (E. Hairer, *Journal of Numerical Analysis*, 2010). Moreover, if a certain quantity should be strictly preserved, one can design a modified discrete gradient by a method proposed by M. Dahlby et al. *Journal of Physics A*, 2011, so that both the energy and the quantity are preserved. We will discuss these potential extensions as future work in the final version.

higher-order systems: Single precision (32-bit floating-point numbers) was used for ODE datasets, and double precision (64-bit) for PDE, as stated in lines 236 and 298. Although, as stated in line 205, the proposed discrete gradient method is a second-order method, higher-order methods can be designed using the composition method (using multiple sub-steps) and the higher-order temporal difference (using multiple steps) as introduced in [12]. We will add the above explanation in the final version. Moreover, as shown in Tables 2–3, the proposed discrete gradient method works with a smaller error than the Dormand–Prince method with the adaptive time-stepping, which is a fourth-order method.

novelty in the math: Firstly, the automatic discrete differentiation algorithm is a novel mathematical idea. Secondly, dissipative SymODEN in [35] introduced a matrix similar to G in our paper as the standard skew-symmetric matrix minus a symmetric positive semi-definite matrix. From a geometric point of view, this means that the systems are defined on cotangent bundles, while our approach is formulated on general symplectic/Riemannian manifolds. This extension enables our method to handle the PDE systems (e.g., the Cahn–Hilliard equation and the KdV equation).

Lagrange / VIN / VIGN, particularly, VIN isn't benchmarked: Our study focuses on both energy-conservative and -dissipative systems. To this end, the Hamiltonian formalism is suitable because it has the equation of the same form as the dissipative gradient flows on Riemannian manifolds. In fact, it can express many kinds of conservative and dissipative ODEs and PDEs in a single unified way as shown in Eq. (1) in our paper, emphasizing the impact of the proposed discrete gradient. Conversely, Lagrangian mechanics is not obviously extendable to general dissipative systems (e.g., the Cahn–Hilliard equation), and the variational integrator used by VIN is a second–order symplectic integrator derived from Lagrangian mechanics. Instead, we have evaluated the leapfrog integrator, which is also a second–order symplectic integrator and is applicable to Hamiltonian formalism. We will add these explanations in the final version along with the suitable references for the Lagrangian approaches.

SympODEN dissipative is not benchmarked: SympODEN dissipative employed a dissipative term (in the same way as Eq. (3) in our paper) and a control input (not needed in our experiments), and learned from finite difference using a Runge–Kutta method (in the same way as our experiments). Hence, all proposals of SympODEN dissipative were already considered or out of scope. In the final version, we will explain this fact clearly.

extendable to N particle systems: Our proposed automatic discrete differentiation algorithm is defined using the Fréchet derivative, and thus it is applicable to any kind of network architectures. This means that the proposed DGNet is extendable to operations on a graph and thereby to N particle systems in the same way as VIN is extended to VIGN.

The computational overhead: For training, the computational cost of the proposed scheme in Eq. (6) is no more than twice of the HNN with the Euler method and tens times smaller than that with the adaptive Dormand–Prince method. Besides, the Ge–Marsden theorem shows that no method can be both symplectic and strictly energy-preserving (see G. Zhong and J. Marsden, Physics Letters A, 1988). Hence DGNet does not conflict with but complements VIN. One can choose a preferable one depending on targeted tasks.

control might be incorporated: In the same way as dissipative SymODEN, DGNet can be extended to a port-Hamiltonian system that accepts the control input. We will add some comments on this in the final version.

Additional: We thank you for the comments on our bobtailed explanations. We will clarify them in the final version.

Reply to Reviewer #4: We deeply thank you for your review and appreciation.