ConCerNet: A Contrastive Learning Based Framework for Automated Conservation Law Discovery and Trustworthy Dynamical System Prediction

Wang Zhang [1] **Tsui-Wei Weng** [2] **Subhro Das** [3] **Alexandre Megretski** [1] **Luca Daniel** [1] **Lam M. Nguyen** [4 3]

Abstract

Deep neural networks (DNN) have shown great capacity of modeling a dynamical system; nevertheless, they usually do not obey physics constraints such as conservation laws. This paper proposes a new learning framework named **ConCerNet** to improve the trustworthiness of the DNN based dynamics modeling to endow the invariant properties. **ConCerNet** consists of two steps: (i) a contrastive learning method to automatically capture the system invariants (i.e. conservation properties) along the trajectory observations; (ii) a neural projection layer to guarantee that the learned dynamics models preserve the learned invariants. We theoretically prove the functional relationship between the learned latent representation and the unknown system invariant function. Experiments show that our method consistently outperforms the baseline neural networks in both coordinate error and conservation metrics by a large margin. With neural network based parameterization and no dependence on prior knowledge, our method can be extended to complex and large-scale dynamics by leveraging an autoencoder.

1. Introduction

Many critical discoveries in the world of physics were driven by distilling the invariants from observations. For instance, the Kepler laws were found by analyzing and fitting parameters for the astronomical observations, and the mass conservation law was first carried out by a series of experiments. However, such discoveries usually require extensive human insights and customized strategies for specific problems. This naturally raises a question:

> **Q1:** *Can we learn conservation laws from real-world data in an automated fashion?*

Recent works on automatic discovery of scientific laws try to answer the above question by proposing symbolic regression (Udrescu & Tegmark, 2020). The idea is to recursively fit the data to different combinations of pre-defined function operators. However, these methods' implicit dependence on human knowledge (i.e. function class and complexity) and high computational cost limit their application to small physics equations, making them less scalable and incapable of handling general systems with complicated dynamics.

On the other hand, the approach of data-driven dynamical modeling tries to learn a dynamical system from data, which often generate models that are prone to violation of physics laws as demonstrated in (Greydanus et al., 2019). This motivates a second question:

> **Q2:** *Can we learn a dynamical system that is trustworthy (i.e. obey physics laws)?*

There has been a recent line of work trying to answer this question by actively constructing dynamics models that obey physical constraints. For example, Greydanus et al. (2019) enforces the Hamiltonian to be conserved in Hamiltonian systems, Cranmer et al. (2020) further extends it to Lagrangian dynamics. As these works focus on specific systems, there is yet a method to be designed for preserving the conservation quantities in *general* dynamical systems.

In this work, we aim at answering the two questions **Q1** and **Q2** jointly by introducing a new framework **ConCerNet**[1], which is a novel pipeline to learn trustworthy dynamical systems $f(x)$ based on the automatic discovery of the system's invariants $H(x)$ from data. **ConCerNet** consists of two steps: (i) contrastively learning $H(x)$ as low-dimensional representation as the invariant quantity for the system; (ii) learning $f(x)$ to approximate the system dynamics with a correction step. This ensures the learned dynamics to automatically preserve the learned $H(x)$. An overview of ConCerNet method pipeline is shown in Figure 1.

[1]Massachusetts Institute of Technology, Cambridge, MA, USA [2]University of California, San Diego, CA, USA [3]MIT-IBM Watson AI Lab, IBM Research, Cambridge, MA, USA [4]IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Correspondence to: Wang Zhang <wzhang16@mit.edu>, Lam M. Nguyen <LamNguyen.MLTD@ibm.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

[1]Source code available at https://github.com/wz16/concernet.

We summarize our main contributions as follows:

1. We provide a novel contrastive learning perspective of dynamical system trajectory data to capture their invariants. We design a **Square Ratio Loss** function as the contrastive learning metric. To the best of the authors' knowledge, this is the first work that studies the discovery of conservation laws for general dynamical systems through contrastive learning.
2. We propose a **Projection Layer** to impose conservation of the invariant function for dynamical system trajectory prediction, preserving the conservation quantity during dynamics modeling.
3. Leveraging the above components, we establish a generic learning framework for dynamical system modeling named **ConCerNet** (CONtrastive ConsERved Network). It provides robustness in prediction outcomes and flexibility for application in a wide range of dynamical systems that mandate conservation properties.
4. Under mild conditions, we prove the local minimum property of **Square Ratio Loss**, theoretically bridging the relationship between the learned latent representation and original conservation function.
5. We conduct extensive experiments to demonstrate the efficacy of **ConCerNet**. Our contrastive learning method automatically discovers physical conservation laws, and the coordinate error/conservation violation of **ConCerNet** are much smaller than the baseline neural networks in dynamics prediction.

2. Background and Related Work

2.1. Contrastive Learning

Unlike discriminative models that explicitly learn the data mappings, contrastive learning aims to extract the data representation implicitly by comparing among examples. The early idea dates back to the 1990s (Bromley et al., 1993) and has been widely adopted in many areas. One related field to our work is metric learning (Chopra et al., 2005; Harwood et al., 2017; Sohn, 2016), where the goal is to learn a distance function or latent space to cluster similar examples and separate the dis-similar ones.

Contrastive learning has been a popular choice for self-supervised learning (SSL) tasks recently, as it demonstrated its performance in many applications such as computer vision (Chen et al., 2020a; He et al., 2020; Ho & Nvasconcelos, 2020; Tian et al., 2020) and natural language processing (Wu et al., 2020b; Gao et al.). There are many existing works related to the design of contrastive loss (Oord et al., 2018; Chen et al., 2020a;b). Prior to our work to use contrastive learning in exploiting temporal correlation, Hyvarinen & Morioka (2016; 2017) propose time contrastive learning to discriminate time segments and Eysenbach et al. (2022) estimates Q-function in reinforcement learning by comparing state-action pairs.

2.2. Deep Learning based Dynamical System Modeling

Constructing dynamical system models from observed data is a long-standing research problem with numerous applications such as forecasting, inference and control. System identification (SYSID) (Ljung et al., 1974; Ljung, 1998; Keesman & Keesman, 2011) was introduced 5-6 decades ago and designed to fit the system input-output behavior with choice of lightweight basis functions. In recent years, neural networks became increasingly popular in dynamical system modeling due to its representation power. In this paper, we consider the following neural network based learning task to model an (autonomous and continuous time) dynamical system:

$$f_\theta(x(t)) \sim \dot{x}(t) \equiv \frac{dx(t)}{dt} \tag{1}$$

where $x \in \mathbb{R}^n$ is the system state and $\dot{x}$ is its time derivative. $f_\theta : \mathbb{R}^n \to \mathbb{R}^n$ denotes the neural network model f with parameter θ, and f_θ is used to approximate dynamics evolution.

The vanilla neural networks learn the physics through data by minimizing the step prediction error, without a purposely designed feature to honor other metrics such as conservation laws. One path to address this issue is to include an additional loss in the training (Singh et al., 2021; Wu et al., 2020a; Richards et al., 2018; Wang et al., 2020); however, the soft Lagrangian treatment does not guarantee the model performance during testing. Imposing hard constraints upon the neural network structures is a more desirable approach, where the built-in design naturally respect certain property regardless of input data. Existing work includes: Kolter & Manek (2019) learns the dynamical system and a Lyapunov function to ensure the exponential stability of predicted system; Hamiltonian neural network (HNN, Greydanus et al. (2019)) targets at the Hamiltonian mechanics, directly learns the Hamiltonian and uses the symplectic vector field to approximate the dynamics; Lagrangian neural network (LNN, Cranmer et al. (2020)) extends the work of HNN to Lagrangian mechanics. Although the above models are able to capture certain conservation laws under specific problem formulations, they are not applicable to general conserved quantities (e.g. mass conservation). This motivates our work in this paper to propose a contrastive learning framework in a more generic form that is compatible of working with arbitrary conservation.

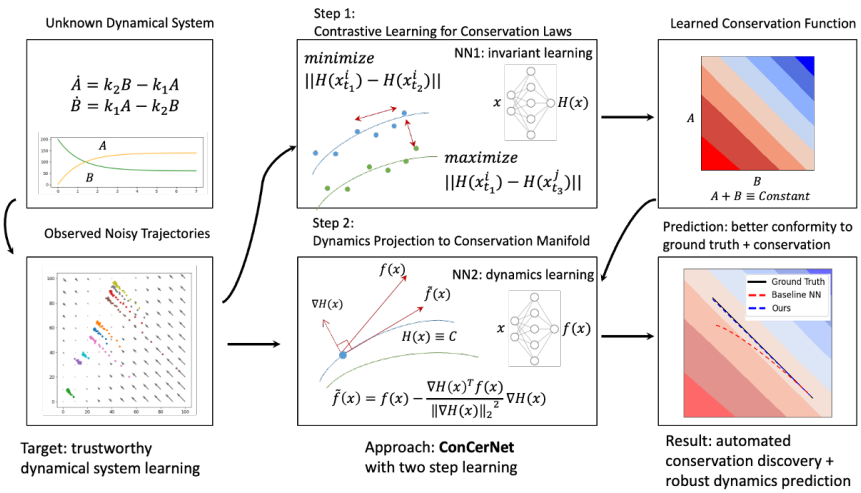

Figure 1. **ConCerNet** pipeline to learn the dynamical system conservation and enforce it in simulation. A contrastive learning framework is proposed to extract the invariants across trajectory observations, then the dynamical model is projected to the invariant manifold to ensure the conservation property.

2.3. Learning with Conserved Properties

Automated scientific discovery from data without prior knowledge has attracted great interest to both communities in physics and machine learning. Besides the above-mentioned HNN and LNN, a few recent works (Zhang et al., 2018; Liu & Tegmark, 2021; Ha & Jeong, 2021; Liu et al., 2022; Udrescu & Tegmark, 2020) have explored automated approaches to extract the conservation laws from data. Despite the promising results, the existing methods are mostly based on symbolic regression, and they suffer from limitations including poor data sampling efficiency and rely on prior knowledge and artificial preprocessing. Often they use search algorithms on pre-defined function classes, hereby are difficult to extend to larger and more general systems. We show the method applicability comparison in Table 1. To the best of our knowledge, our work is the first time to study automated conservation law discovery: 1. for general dynamical system with arbitrary conservation property without any prior knowledge 2. through the lens of contrastive learning.

3. Proposed Methods: ConCerNet

3.1. Step 1: Learning Conservation Property from Contrastive Learning

In the practice of dynamical system learning, the dynamics data is usually observed as a set of trajectories of system state $\{x^i_t \in \mathbb{R}^n\}^{N,T}_{i=1,t=1}$, where i denotes the trajectory index of total trajectory number N and t is the time step with the number of total time step T. Let $\{x^i_1\}^N_{i=1}$ be the initial conditions and assume they have different conservation values. Let $H_{\theta_c} : \mathbb{R}^n \to \mathbb{R}^m$ be the m-dimensional latent mapping function parameterized by the neural network. Following the convention of classical contrastive learning, the system states within the same trajectory are considered "in the same class" and with the same value of conservation properties. Since the invariants are well conserved along the trajectory and differ among different trajectories, we aim to find the conservation laws that naturally serve as each class's latent representation.

Table 1. Applicability comparison of automatic methods to recover conservation laws.

Method	Applicability	Do not need pre- processing /prior knowledge
Symbolic Reg.	Simple conservation	×
HNN/LNN	Hamiltonians/Lagrangians	√
ConCerNet	General conservation	√

As a metric to persuade similar latent representation within the same trajectory and encourage discrepancy between different trajectories, we introduce the **Square Ratio Loss** (SRL) defined in the following as the contrastive loss function:

$$\mathcal{L}_{\text{SR}} = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t_1=1}^{T} \left[\frac{\sum_{t_2=1}^{T} \|H_{\theta_c}(x^i_{t_1}) - H_{\theta_c}(x^i_{t_2})\|^2}{\sum_{j=1}^{N} \sum_{t_3=1}^{T} \|H_{\theta_c}(x^i_{t_1}) - H_{\theta_c}(x^j_{t_3})\|^2} \right]. \quad (2)$$

In each fraction, the denominator summarizes the squared Euclidean norm in the latent space between the anchor point and all other points, the numerator only summarizes between the anchor point and points within the same trajectory. Therefore, we call this loss function design "square

ratio loss". Intuitively, minimizing this loss function will decrease the latent discrepancy between points within the same trajectory and increase the distance between different trajectories.

To notice, in common contrastive learning settings like SimCLR (Chen et al., 2020a), the latent space is usually measured by Cosine distance between point pairs, rather than the Euclidean distance. Besides, contrastive learning is a classification task while our metric is pure value comparison. Further, we compare one point to a group of points in the same class, this is similar to the NaCl loss in (Ko et al., 2022) with many similar points to the anchor point. We choose SRL as metric for a few reasons: 1. In contrast with classification objects with discretized sampling distribution space, the dynamical system lives in a continuous space, Euclidean distance comparison intuitively works better. 2. SRL achieves good experimental performance, as we will show in Section 5. 3. Through the latter theoretical analysis in Theorem 4.9, we can prove the optimization property of SRL that draws the relationship between the learned representation and the exact conservation law.

3.2. Step 2: Enforcing Conservation Invariants in Dynamical Modeling

Once the conservation term $H_{\theta_c}(x) \in \mathbb{R}^m$ is contrastively learned or given from prior knowledge, we attempt to enforce the predicted trajectory along the conservation manifold in the simulation stage, $s.t.\ \frac{dH_{\theta_c}(x)}{dt} = 0$. In the continuous dynamical system like Equation (1), we can project the nominal neural network output $f_{\theta_d}(x)$ onto the conservation manifold by eliminating its parallel component to the normal direction of the invariant planes (i.e. $\nabla_x H_{\theta_c}(x)$). Let $G = \nabla_x H_{\theta_c}(x) \in \mathbb{R}^{n\times m}$, we define the projected dynamical model $\tilde{f}_{\theta_d}(x)$ as following:

$$\begin{aligned}\tilde{f}_{\theta_d}(x) &\coloneqq \text{Projection}\left(f_{\theta_d}(x), \{f : G^\top f = 0\}\right)\\ &= f_{\theta_d}(x) - G(G^\top G)^{-1}G^\top f_{\theta_d}(x)\\ &= f_{\theta_d}(x) - \sum_{i=1}^{m}(G_i^\perp)^\top f_{\theta_d}(x) G_i^\perp,\end{aligned} \tag{3}$$

where the second equality is the standard orthogonal projection equation and the third equality indicates we use the Gram-Schmidt process to solve it in practice. Here $G^\perp$ denotes the orthonormalized matrix from G calculated by Gram–Schmidt process and $G_i^\perp$ is the ith component. The projected dynamics function naturally satisfies $(\nabla_x H_{\theta_c}(x))^\top f = 0$ and therefore guarantees $H_{\theta_c}(x)$ to be constant during prediction. The intuitive diagram of projection with one conservation term (i.e. $m = 1$) is shown in Figure 1.

For dynamical system learning, we assume the system time derivative is observable (with noise) for simplicity. The loss function is the mean square loss between the neural network prediction and the system time derivative, as shown in Equation (4). For real-world problems with only discretized state observations, we can approximate the derivative by time difference or bridge the continuous system and discretized data by leveraging neuralODE (Chen et al., 2018).

$$\mathcal{L}_{\text{dyn}} = \mathbb{E}_x\left[\|\tilde{f}_{\theta_d}(x) - \dot{x}\|^2\right]. \tag{4}$$

4. Theory

In this section, we rigorously formulate the contrastive learning problem from Section 3.1 and study the latent function property in the case of the SRL loss.

Definition 4.1. For function $g(\cdot) : \mathcal{X} \to \mathbb{R}^1$ defined on a compact and convex set $\mathcal{X} \subset \mathbb{R}^n$, $y \in \mathbb{R}^1$, $\epsilon > 0$, define the **preimage** $g_{\mathcal{X}}^{-1}(y, \epsilon) = \{x | x \in \mathcal{X}, ||g(x) - y|| \le \epsilon\}$. The preimage represents the set of elements in $\mathcal{X}$ mapped to the ϵ ball centered at y in the image space. ϵ is called image neighborhood diameter.

Definition 4.2. For a given implicit $g(\cdot)$ and state set $\mathcal{X}$, define the **Integral Square Ratio Loss (ISRL)** as a function of neighborhood diameter ϵ and target function $h(\cdot) : \mathbb{R}^n \to \mathbb{R}^1$:

$$\mathcal{L}_{ISR}(h(\cdot), \epsilon) = \int_{\mathcal{X}} \frac{\int_{g_{\mathcal{X}}^{-1}(g(x),\epsilon)}(h(x) - h(x'))^2 dx'}{\int_{\mathcal{X}}(h(x) - h(x''))^2 dx''} dx. \tag{5}$$

Remark 4.3. To facilitate the notation and problem formulation, we convert the discretized SRL from Equation (2) into the continuous and surrogate version of SRL in Equation (5). To notice, they are not equivalent: 1. In the discretized version, the numerator summation set is the points within the same trajectory, it is a "partial preimage" being a subset of the preimage integral area in the continuous loss. In many cases, we only have partial preimage due to observation interval or trajectory length. For the rest of the analysis, we consider the ideal case with access to full preimage. However, we can perform a similar analysis on the partial preimage case and the final functional relationship will be confined to the union of the observed partial preimage. 2. In practice, there exists noise in observation. The continuous version explicitly counts in the factor and summarizes over all the preimage within ϵ diameter to the anchor point. 3. The actual system might have more than one conservation law and **ConCerNet** allows more than one dimension for $H_{\theta_c}(\cdot)$. In this section, we only consider $g(\cdot)$ and $h(\cdot)$ being one-dimensional. For a system with multiple conservation laws, we can pick one conservation term. In fact, we cannot even guarantee to find the exact conservation function for such systems, as any combination of different conservation functions is conserved as well.

Definition 4.4. Let $\mathcal{X} \subset \mathbb{R}^n$ be a compact and convex set with positive volume (w.r.t. the Lebesgue measure). Let $\mathcal{F}$ denote the space of all square integrable[2] real-valued functions $f : \mathcal{X} \to \mathbb{R}$.
For function $f_1(\cdot), f_2(\cdot) \in \mathcal{F}$, we say that $f_2(\cdot)$ is **"functional injective"** with respect to $f_1(\cdot)$ when $\forall x_1, x_2 \in \mathcal{X}, f_1(x_1) = f_1(x_2) \implies f_2(x_1) = f_2(x_2)$.
Let $\mathcal{F}_{inj}(f_1)$ denote "functional injective set" with respect to $f_1(\cdot)$, the set of all the functions in $\mathcal{F}$ which are functional injective to $f_1(\cdot)$.

Remark 4.5. $f_2(\cdot)$ being "functional injective" to $f_1(\cdot)$ is equivalent to saying that there exists another function $f_3(\cdot)$, s.t. $f_2(\cdot)$ can be written as a composition of $f_3(\cdot)$ and $f_1(\cdot)$: $f_2(x) = f_3(f_1(x))$.

For functions $f_1(\cdot), f_2(\cdot) \in \mathcal{F}$, we say that $f_2(\cdot)$ is **"skew-symmetric functional injective"** with respect to $f_1(\cdot)$, we have $\mathbb{E}[f_2|f_1] = 0$.[3]
Let $\mathcal{F}_{skewsym\text{-}inj}(f_1)$ denote "skew-symmetric functional injective set" with respect to $f_1(\cdot)$, the set of all the functions in $\mathcal{F}$ which are skew-symmetric functional injective to $f_1(\cdot)$.

Proposition 4.6. *$\mathcal{F}$ defined in Definition 4.4 is a vector space over field $\mathbb{R}$. For square integrable function $f_1(\cdot) : \mathcal{X} \to \mathbb{R}^1$, $\mathcal{F}_{inj}(f_1)$ and $\mathcal{F}_{skewsym\text{-}inj}(f_1)$ are complemented subspaces of $\mathcal{F}$.*

Definition 4.7. Let $\mathcal{L}(\cdot) : \mathcal{F} \to \mathbb{R}^1$ be a function on $\mathcal{F}$. Let $\mathcal{F}_1$ and $\mathcal{F}_2$ be complemented subspaces of $\mathcal{F}$. We say $\mathcal{L}$ reaches a "directional local minimum" for any perturbation along $\mathcal{F}_2$ at $f_1(\cdot) \in \mathcal{F}_1$, if $\forall f_2 \in \mathcal{F}_2$ not being a zero function, $\exists \bar{\delta} > 0$, such that $\mathcal{L}(f_1(\cdot)+\delta f_2(\cdot)) > \mathcal{L}(f_1(\cdot)), \forall \delta < \bar{\delta}$.

Definition 4.8. For function $f_1(\cdot)$ and $f_2(\cdot)$ defined on $\mathcal{X}$, we say $f_2(\cdot)$ is "C-relative Lipschiz continuous" to $f_1(\cdot)$ if $\forall x_1, x_2 \in \mathcal{X}, ||f_2(x_1) - f_2(x_2)|| \leq C||f_1(x_1) - f_1(x_2)||$ for some constant C.

With the above preparation, we are ready to introduce the main theoretical result of this paper, which studies the local minimum property of the loss function when $h(\cdot)$ is "functional injective" to $g(\cdot)$. We make three moderate assumptions in the following. Assumption 4.10 caps the variation magnitude between $h(\cdot)$ and $g(\cdot)$. Assumption 4.11 and Assumption 4.12 require the image neighborhood diameter ϵ to be small. The proof of Theorem 4.9 is delayed to Appendix A.3.

Theorem 4.9. *For a compact and convex set $\mathcal{X} \subset \mathbb{R}^n$ of positive volume, consider a non-constant continuously differentiable (thus also square integrable) function $g(\cdot) : \mathcal{X} \to \mathbb{R}^1$ and corresponding preimage operator $g_{\mathcal{X}}^{-1}(\cdot, \epsilon)$ (defined in Definition 4.1) with diameter $\epsilon > 0$. Let $\mathcal{F}$ be the functional space of square integrable real-valued functions defined on $\mathcal{X}$. Let $\mathcal{L}_{ISR}$ be the integral square ratio loss from Definition 4.2. Assume $h(\cdot) \in \mathcal{F}_{inj}(g)$ and $\epsilon > 0$ satisfy Assumption 4.10, 4.11 and 4.12. Then $\mathcal{L}_{ISR}$ reaches a directional local minimum (defined in Definition 4.7) at $h(\cdot)$ for any perturbation from $\mathcal{F}_{skewsym\text{-}inj}(g)$ (defined in Definition 4.4) .*

Assumption 4.10. $h(\cdot)$ is C_1-relative Lipschiz continuous to $g(\cdot)$ and $g(\cdot)$ is C_1'-relative Lipschiz continuous to $h(\cdot)$.

Assumption 4.11. The image neighborhood diameter ϵ (defined in Definition 4.2) is small enough s.t. $\forall x \in \mathcal{X}$, $\epsilon < \sqrt{\frac{\int_{\mathcal{X}}(h(x)-h(x''))^2 dx''}{\int_{\mathcal{X}} 1 dx''}} / (C_1\sqrt{5C_2})$, where C_1 is the Lipschitz continuous constant defined in Assumption 4.10 and $C_2 = \frac{\max_{x_1} \int_{\mathcal{X}}(h(x_1)-h(x''))^2 dx''}{\min_{x_2} \int_{\mathcal{X}}(h(x_2)-h(x''))^2 dx''}$.

Assumption 4.12. ϵ is small enough s.t. for $\forall x \in \mathcal{X}$,

$$C_1 \epsilon \frac{\int_{x' \in g_{\mathcal{X}}^{-1}(g(x),\epsilon)} (h(x) - h(x'))dx'}{\int_{x' \in g_{\mathcal{X}}^{-1}(g(x),\epsilon)} (h(x) - h(x'))^2 dx'} < \int_{x \in \mathcal{X}} 1 dx.$$

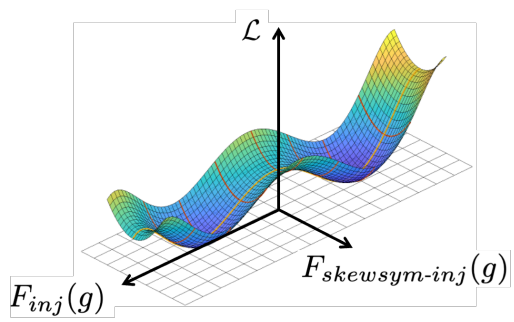

Figure 2. Illustrative Diagram for Theorem 4.9. The yellow curve denotes $h(\cdot) \in \mathcal{F}_{inj}(g)$, the red curves denote the perturbations in $\mathcal{F}_{skewsym\text{-}inj}(g)$.

Remark 4.13. Consider $h(\cdot)$ is parameterized on the functional space $\mathcal{F}$, recall the definition of "directional local minimum" from Definition 4.7, Theorem 4.9 claims if the optimization finds $h(\cdot) \in \mathcal{F}_{inj}(g)$ and it satisfies the above conditions, then any perturbation from $\mathcal{F}_{skewsym\text{-}inj}(g)$ results in an increase of the loss function. Figure 2 provides an illustrative diagram of such relationship. In practice, once the gradient descent optimizer finds a function injective $h(\cdot)$ to $g(\cdot)$, the optimizer will stay inside $\mathcal{F}_{inj}(g)$, as the gradient descent method reaches a local minimum point for any perturbation step along $\mathcal{F}_{skewsym\text{-}inj}(g)$.

[2]The element in L^2 space is an equivalent class of functions equals to each other almost everywhere, here we use one function f to represent the equivalent class, as all equivalent function have the same integral result on positive volume.

[3]With $\mathcal{X}$ as the sampling space of random variable x, we define the probability measure P, s.t. $\forall A \subset \mathcal{X}$, $P(A) = |A|/|\mathcal{X}|$, where $|\cdot|$ represents the n-dimensional Lebesgue measure. Consider f_1 and f_2 as two random variables, the conditional expectation $\mathbb{E}[f_2|f_1]$ is a function that takes the value of $\mathbb{E}[f_2|f_1 = r]$ whenever $f_1 = r$, i.e., $\phi(r) = \int f_2 p(f_2|f_1 = r) df_2 = \int_{f_1^{-1}(r)} f_2(x) p(f_2(x)|f_1(x) = r) dx$. $p(f_2(x)|f_1(x) = r)$ has the Dirac delta form if $P(f_1^{-1}(x) = r) = 0$.

As the major theoretical result of this paper, Theorem 4.9 narrates that through our proposed method, without knowing the exact invariant mapping function of the physics systems, the neural network is capable to recover it from the state neighborhood relationship in the image space. To the best of our knowledge, this is the first theoretical result on automated conservation law recovery, potentially building the bridge between AI and human-reliant scientific discovery work. Although the framework is used for dynamical system trajectory observations in this paper, the insight can be further extended to other physics or non-physics systems with inherent invariant properties.

5. Experiments

We first use two simple conservation examples to illustrate the ConCerNet procedures, then we discuss the model performance for a system with multiple conservation laws. In the end, we demonstrate the power of our method and extend the ConCerNet pipeline to high-dimensional problems by leveraging an autoencoder. As our task focuses on improving dynamical simulation trustworthiness on conservation properties, the conservation learning experiments are recorded in Table 2 and the dynamical model performance is recorded in Table 3. We mainly compare ConCerNet with a baseline neural network, which does not include the projection module but shares the exact dynamics network structure and learning scheme/loss function with ConCerNet. We also compare ConCerNet with one classical modeling method (SINDy, (Brunton et al., 2016)) and a DNN-based prior work (HNN, (Greydanus et al., 2019)) and delay the results to Appendix B.2, where ConCerNet shows similar performance but ConCerNet is more generally applicable. All the experiments are performed over 5 random seeds, more system and experiment details are listed in Appendix B.1.

5.1. Simple Conservation Examples

In this section, we introduce two simple examples: **Ideal spring mass system** under energy conservation ($x[1]^2 + x[2]^2$) and **Chemical reaction** under mass conservation ($x[1] + x[2]$). Both systems have 2D state space for easier visualization of the learned conservation function. Figure 3 shows the learned conservation compared with ground truth. The contrastive learning process captures the quadratic and linear functions, as the contour lines are drawn in circles and lines. To notice, the learned conservation function here is approximately the exact conservation function differing by some constant coefficient (we will further discuss their relationship in Section 6.1). This is a natural result because the SRL is invariant to linear transform of $H_{\theta_c}(\cdot)$; furthermore, the real-world conservation quantity is also a relative value instead of an absolute value. In Figure 4, we compare the two methods by showing the typical trajectory, conservation violation and coordinate error to the ground truth. The vanilla neural network is likely to quickly diverge from the conserved trajectory, and the error grows faster than our proposed method.

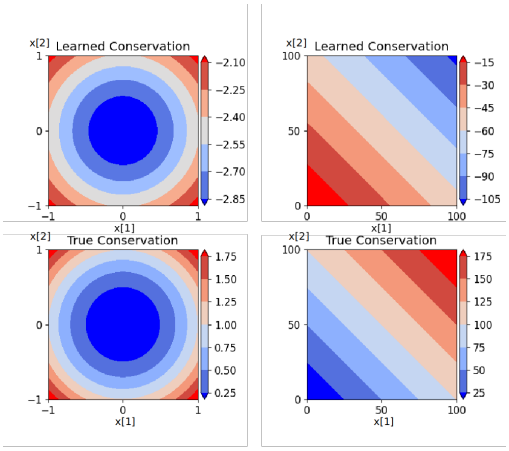

Figure 3. Learned conservation function vs ground truth, left: ideal spring mass, right: chemical kinetics

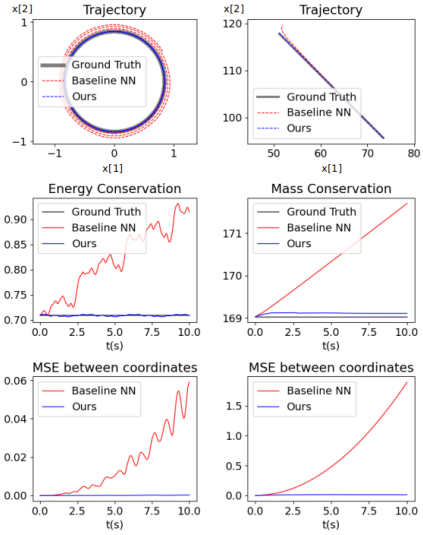

Figure 4. Simulation comparisons of two simple examples: left column: ideal spring mass system, right column: chemical kinematics. 1st row: state trajectories, 2nd row: violation of conservation laws to ground truth, 3rd row: mean square error to ground truth.

5.2. Systems with More Than One Conservation Functions

In this section, we tackle a more complex system with more than one conservation law. The **Kepler system** describes a planet orbiting around a star with elliptical trajectories. The planet's state is four-dimensional, including both coordinates in the 2D plane and the corresponding velocity. The

system has two conservation terms (energy conservation $g_1(x) = \frac{x[3]^2+x[4]^2}{2} - \frac{1}{\sqrt{x[1]^2+x[2]^2}}$ and angular momentum conservation $g_2(x) = x[1]x[4] - x[2]x[3]$, we neglect the Laplace–Runge–Lenz vector for simplicity in this paper).

In our prior discussion (Remark 4.3), we claim that if the system has more than 1 conservation property, our method does not guarantee to find the injective function of each individual conservation equation. Consider the Kepler example, any functional combination of $g_1(x)$ and $g_2(x)$ is also conserved. In practice, we found the learned function is likely to converge to some linear function of the simpler conservation function. As we cannot visualize functions with four dimension input, we randomly sample 10 trajectories and calculate the R^2 by linear regressing the learned function towards each conservation function. The results in Table 2 indicate that the learned function correlates with the angular momentum equation better than the energy equation which is more difficult to represent by the neural network. We also test the latent neural network with two-dimensional output ($m = 2$), the two outputs are likely to be linear to each other, which does not affect dynamical prediction or the square ratio loss. Despite not learning all the conservation laws, in the simulation stage, our method still outperforms the vanilla neural network by a large margin in both metrics with the learned invariant, as the results are shown in Table 3.

5.3. Larger System: Heat Equation

To further extend our model to larger systems, we test our method on the **Heat Equation** on a 1D rod. The 1D rod is given some initial temperature distribution and insulated boundary conditions on both ends. The temperature $U(y, t)$, as a function of coordinate and time, gradually evens up following the heat equation $\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial y^2}$. The total internal energy along the rod does not vary because the heat flow is blocked by the boundary. We use system state x consisting of overall 101 nodes to discretize the entire interval $[-5, 5]$ and compress system states to a 9 dimension latent space with an autoencoder pair $(E_{\theta_E}, D_{\theta_D})$. For both contrastive conservation learning and dynamical system learning, original space state and time derivative $(x, \dot{x})$ are mapped to the autoencoder latent space $(z, \dot{z})$ by

$$z = E_{\theta_E}(x), \dot{z} = \frac{\partial E_{\theta_E}(x)}{\partial x} \times \dot{x}, \tag{6}$$

where $\times$ denotes the matrix multiplication by chain rule, the partial derivative from latent space to original space can be calculated by auto-differentiation package. After simulation, the latent space trajectory will be mapped back to the original space by D_{θ_D}.

Table 2 shows the neural network is capable to capture an affine function of mass conservation, regardless of the reduced order and non-linear representation space. Figure 5 draws the simulation result and conservation metric comparison between the vanilla neural network and our method. For both methods, the initial conservation violation error was introduced by the autoencoder. In general, our method conforms to ground truth trajectory and conservation laws much better than the vanilla method.

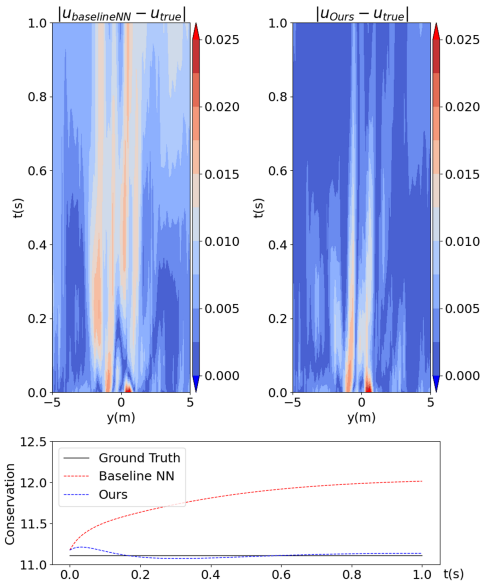

Figure 5. Heat Equation Simulation. Upper: Simulation coordinate error comparison: vanilla neural network and ConCerNet. Bottom: Conservation violation to ground truth.

5.4. Experiment Summary

We generalize the quantitative results for all the experiments above in Table 2 and Table 3. For conservation learning testing, we randomly sample 10 trajectories and linear regress the learned conservation function into each of the exact conservation laws and report the R^2 values. To test dynamics modeling, we sample 10 initial points and integrate the simulated trajectory with Runge-Kutta method, and calculate the trajectory state coordinate error and conservation violation to the ground truth. Our method outperforms the baseline neural networks as the error is often multiple times smaller. One may notice that the standard deviation is comparable with the error metric in the dynamics experiments, this is due to the instability of the dynamical systems. The trajectory tracking error will exponentially grow as a function of time, and the cases with outlier initialization are likely to dominate the averaged results and lead to large variance. In practice, our method is capable to control the tracking deviation better than the baseline method across almost all the cases.

Table 2. Average R^2 with linear regression between the learned function and exact conservations

Task	Conservation	R^2
Ideal spring mass system	$x[1]^2 + x[2]^2 = C$	0.997 ± 0.001
Chemical kinematics	$x[1] + x[2] = C$	$0.999998 \pm 1.9e-6$
Kepler system	$\frac{x[3]^2+x[4]^2}{2} - \frac{1}{\sqrt{x[1]^2+x[2]^2}} = C_1$	0.858 ± 0.104
	$x[1]x[4] - x[2]x[3] = C_2$	0.994 ± 0.001
Heat equation	$\int U dx = C$	0.9995 ± 0.0003

Table 3. Simulation error over the tasks

Task	Mean square error		Violation of conservation laws	
	Baseline NN	ConCerNet	Baseline NN	ConCerNet
Ideal spring mass system	0.022 ± 0.023	**9.2e-3 ± 5.6e-3**	0.012 ± 0.032	**1.4e-4 ± 7.4e-5**
Chemical reaction	9.4e-3 ± 5.8e-3	**5.9e-3± 3.7e-3**	0.019 ± 0.026	**5.6e-3 ± 5.4e-3**
Kepler system	0.794 ± 0.476	**0.140 ± 0.067**	0.016 ± 0.011	**5.2e-3 ± 4.2e-3**
Heat equation	7.34e-5 ± 2.62e-5	**2.79e-5 ± 1.28e-5**	0.094 ± 0.070	**0.013 ± 0.008**

6. Discussions

6.1. Learned Invariants vs. Exact Conservation laws

From the theoretical side, for systems with only one conservation law, we prove through Theorem 4.9 that the optimizer converges to a composition function of the conservation function $g(\cdot)$. For systems with more than one conservation laws, it is difficult for the system to retrieve even one of the conservation equations, as any combination of individual conservation functions is also conserved, and Theorem 4.9 only guarantees the convergence to one of these combinations. Some prior work claiming to find the exact conservation functions depend on the prior human knowledge of conservation functions (i.e. using symbolic regression or limiting the fitting function classes). However, our paper takes a generic route using neural network parameterization to represent general functions, and therefore it can be extended to more complex systems or even non-linear representations like the heat equation case. In this paper, we do not involve symbolic regression based work in conservation learning comparison baselines as they are dependent on application-specific knowledge.

In practice, we found $h(\cdot)$ is likely to be a linear function of $g(\cdot)$, as Table 2 shows the R_2 coefficients under linear regression, we can also tell the linear relationship from the contours in Figure 3.

Despite the interesting experimental finding on the linear relationship, we yet have any theoretical ground for it. Our intuitive explanation is that the linear function is relatively easy for the neural network to express. We would like to emphasize that our paper focuses on improving the conservation performance for dynamical modeling, and we leave the task of finding the exact conservation laws or proving convergence to linear functions to future work. Regardless of the mapping between the learned function and the exact conservation function, the simulation is guaranteed to preserve the conservation property if the mapping is injective.

6.2. ConCerNet Robustness to Noisy Observation and Contrastive Learning Data Efficiency

To test the overall ConCerNet method performance under noisy observations, we perform experiments with different noise settings and compare the simulation error. The result is recorded in Table 8 in appendix. In general, ConCerNet is much less vulnerable to observation noise compared with baseline neural networks, as the projection layer provides a robust guarantee that the system will not deviate far away from the conservation manifold.

To test the contrastive conservation learning data efficiency, we conduct experiments with different observed trajectory numbers and points per trajectory and record the R^2 coefficient between the learned function and exact conservation law in Table 10 in Appendix B.4. In general, we usually need 100 to 1000 points to reach $R^2 \approx 1$ for the above four examples, the contrastive conservation learning is relatively data efficient.

7. Conclusion

In this paper, we propose ConCerNet, a generic framework to learn the dynamical system with designed features to preserve the invariant properties along the simulation trajectory. We first learn the conservation manifold in the state space from a contrastive learning perspective, then purposely enforce the dynamical system in the desired subspace leveraging a projection module. We establish the theoretical bridge between the learned latent representation and the actual conservation laws and experimentally validate the advantage of ConCerNet in both simulation error and conservation metrics and the extensibility to complex systems. Despite the paper presenting an end-to-end approach, both contrastive learning on system invariants and projected dynamical system learning can be seen as an independent procedure and open up a different direction. We believe these ideas represent a promising route in automated invariant property discovery and practical dynamics modeling.

References

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6, 1993.

Brunton, S. L., Proctor, J. L., and Kutz, J. N. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020a.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020b.

Chopra, S., Hadsell, R., and LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pp. 539–546. IEEE, 2005.

Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P., Spergel, D., and Ho, S. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*, 2020.

Eysenbach, B., Zhang, T., Salakhutdinov, R., and Levine, S. Contrastive learning as goal-conditioned reinforcement learning. *arXiv preprint arXiv:2206.07568*, 2022.

Gao, T., Yao, X., and Chen, D. SimCSE: Simple contrastive learning of sentence embeddings.

Greydanus, S., Dzamba, M., and Yosinski, J. Hamiltonian neural networks. *Advances in neural information processing systems*, 32, 2019.

Ha, S. and Jeong, H. Discovering conservation laws from trajectories via machine learning. *arXiv preprint arXiv:2102.04008*, 2021.

Harwood, B., Kumar BG, V., Carneiro, G., Reid, I., and Drummond, T. Smart mining for deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2821–2829, 2017.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729–9738, 2020.

Ho, C.-H. and Nvasconcelos, N. Contrastive learning with adversarial examples. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 17081–17093. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c68c9c8258ea7d85472dd6fd0015f047-Paper.pdf.

Hyvarinen, A. and Morioka, H. Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in neural information processing systems*, 29, 2016.

Hyvarinen, A. and Morioka, H. Nonlinear ica of temporally dependent stationary sources. In *Artificial Intelligence and Statistics*, pp. 460–469. PMLR, 2017.

Kaptanoglu, A. A., de Silva, B. M., Fasel, U., Kaheman, K., Goldschmidt, A. J., Callaham, J. L., Delahunt, C. B., Nicolaou, Z. G., Champion, K., Loiseau, J.-C., et al. Pysindy: A comprehensive python package for robust sparse system identification. *arXiv preprint arXiv:2111.08481*, 2021.

Keesman, K. J. and Keesman, K. J. *System identification: an introduction*, volume 2. Springer, 2011.

Ko, C.-Y., Mohapatra, J., Liu, S., Chen, P.-Y., Daniel, L., and Weng, L. Revisiting contrastive learning through the lens of neighborhood component analysis: an integrated framework. In *International Conference on Machine Learning*, pp. 11387–11412. PMLR, 2022.

Kolter, J. Z. and Manek, G. Learning stable deep dynamics models. *Advances in neural information processing systems*, 32, 2019.

Liberzon, D. *Calculus of variations and optimal control theory: a concise introduction*. Princeton university press, 2011.

Liu, Z. and Tegmark, M. Machine learning conservation laws from trajectories. *Physical Review Letters*, 126(18): 180604, 2021.

Liu, Z., Madhavan, V., and Tegmark, M. Ai poincaré 2.0: Machine learning conservation laws from differential equations. *arXiv preprint arXiv:2203.12610*, 2022.

Ljung, L. System identification. In *Signal analysis and prediction*, pp. 163–173. Springer, 1998.

Ljung, L., Gustavsson, I., and Soderstrom, T. Identification of linear, multivariable systems operating under linear feedback control. *IEEE Transactions on Automatic Control*, 19(6):836–840, 1974. doi: 10.1109/TAC.1974.1100702.

Oord, A. v. d., Li, Y., and Vinyals, O. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Richards, S. M., Berkenkamp, F., and Krause, A. The lyapunov neural network: Adaptive stability certification for safe learning of dynamical systems. In *Conference on Robot Learning*, pp. 466–476. PMLR, 2018.

Singh, S., Richards, S. M., Sindhwani, V., Slotine, J.-J. E., and Pavone, M. Learning stabilizable nonlinear dynamics with contraction-based regularization. *The International Journal of Robotics Research*, 40(10-11):1123–1150, 2021.

Sohn, K. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020.

Udrescu, S.-M. and Tegmark, M. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Wang, R., Kashinath, K., Mustafa, M., Albert, A., and Yu, R. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1457–1466, 2020.

Wu, J.-L., Kashinath, K., Albert, A., Chirila, D., Xiao, H., et al. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *Journal of Computational Physics*, 406:109209, 2020a.

Wu, Z., Wang, S., Gu, J., Khabsa, M., Sun, F., and Ma, H. Clear: Contrastive learning for sentence representation. *arXiv preprint arXiv:2012.15466*, 2020b.

Zhang, P., Shen, H., and Zhai, H. Machine learning topological invariants with neural networks. *Physical review letters*, 120(6):066401, 2018.

A. Supplementary proofs

A.1. Proof of Proposition 4.6

Proposition 4.6 contains many claims, and we prove them one by one in the following. Recall $\mathcal{F}$ be the space of square integrable real-valued functions defined on a compact and convex set $\mathcal{X} \subset \mathbb{R}^n$ with positive volume.

1. $\mathcal{F}$ is a vector space over the field $\mathbb{R}$.

   Proof.
   It is obvious the elements in $\mathcal{F}$ and scalars in $\mathbb{R}$ satisfy the associativity and commutativity of vector addition and distributivity of scalar multiplication.
   Closure of vector addition: $\forall f_1(\cdot), f_2(\cdot) \in \mathcal{F}, f_1(\cdot) + f_2(\cdot) \in \mathcal{F}$, because square integrability is preserved under finite summation.
   Zero element of vector addition: there exists zero vector $z(\cdot) \equiv 0, z(\cdot) \in \mathcal{F}$, s.t. $\forall f(\cdot) \in \mathcal{F}, z(\cdot) + f(\cdot) = f(\cdot)$.
   Inverse elements of vector addition: $\forall f(\cdot) \in \mathcal{F}$, there exists $(-f(\cdot)) \in \mathcal{F}, (-f(\cdot)) + f(\cdot) = 0$.
   Compatibility of scalar multiplication with field multiplication: $\forall a, b \in \mathbb{R}, f(\cdot) \in \mathcal{F}, a(bf(\cdot)) = (ab)f(\cdot)$.
   Identity element of scalar multiplication: scalar $1 \in \mathbb{R}, \forall f(\cdot) \in \mathcal{F}, 1f(\cdot) = f(\cdot)$. $\mathcal{F}$ and $\mathbb{R}$ satisfy the axioms of vector space, this completes the proof. □

2. $\forall f_1(\cdot) \in \mathcal{F}, \mathcal{F}_{inj}(f_1)$ is a vector space over the field $\mathbb{R}$.

   Proof.
   Pick two elements $f_{2,a}(\cdot), f_{2,b}(\cdot)$ from $\mathcal{F}_{inj}(f_1)$, by Definition 4.4, $\forall x_i, x_j \in \mathcal{X}, f_1(x_i) = f_1(x_j) \implies f_{2,a}(x_i) = f_{2,a}(x_j), f_{2,b}(x_i) = f_{2,b}(x_j)$. Then we have $(f_{2,b} + f_{2,a})(x_i) = (f_{2,b} + f_{2,a})(x_j) \implies (f_{2,b} + f_{2,a})(\cdot) \in \mathcal{F}_{inj}(f_1)$, vector addition is closed.
   The other axioms are obvious. □

3. $\forall f_1(\cdot) \in \mathcal{F}, \mathcal{F}_{skewsym\text{-}inj}(f_1)$ is a vector space over the field $\mathbb{R}$.

   Proof.

   Pick two elements $f_{2,a}(\cdot), f_{2,b}(\cdot)$ from $\mathcal{F}_{skewsym\text{-}inj}(f_1)$, by Definition 4.4, $f_1(\cdot)$ if $\forall r \in \mathbb{R}, \mathbb{E}[f_{2,a}(x)|f_1(x) = r] = 0, \mathbb{E}[f_{2,b}(x)|f_1(x) = r] = 0$. Then we have $\mathbb{E}[f_{2,a}(x) + f_{2,b}(x)|f_1(x) = r] = 0 \implies (f_{2,b} + f_{2,a})(\cdot) \in \mathcal{F}_{skewsym\text{-}inj}(f_1)$, vector addition is closed.
   The other axioms are obvious. □

4. $\forall f_1(\cdot) \in \mathcal{F}, \mathcal{F}_{skewsym\text{-}inj}(f_1) \cap \mathcal{F}_{inj}(f_1) = \{\text{zero vector}\}$, where the zero vector here is $z(\cdot) \equiv 0$.

   Proof.
   We prove the claim by contradiction. Suppose there exists $x_0 \in \mathcal{X}$ and non-zero $f_2(\cdot) \in \mathcal{F}_{skewsym\text{-}inj}(f_1) \cap \mathcal{F}_{inj}(f_1), f_2(x_0) = s \neq 0$. Because $f_2(\cdot) \in \mathcal{F}_{inj}(f_1), \forall f_1(x) = f_1(x_0), f_2(x) = s$, then we have $\mathbb{E}[f_2(x)|f_1(x) = f_1(x_0)] = s\mathbb{E}[1|f_1(x) = f_1(x_0)] = s \neq 0$. This contradicts with the definition of $\mathcal{F}_{skewsym\text{-}inj}(f_1)$ and completes the proof. □

5. $\forall f_1(\cdot) \in \mathcal{F}, \mathcal{F}_{skewsym\text{-}inj}(f_1) \cup \mathcal{F}_{inj}(f_1) = \mathcal{F}$.

   Proof.
   For arbitrary $f_1(\cdot), f_2(\cdot) \in \mathcal{F}$, let $f_{2,inj}(x) = \mathbb{E}_{x'}[f_2(x')|f_1(x') = f_1(x)], f_{2,skewsym\text{-}inj}(x) = f_2(x) - f_{2,inj}(x)$.
   For $f_{2,inj}(x), \forall x_i, x_j \in \mathcal{X}$, if $f_1(x_i) = f_1(x_j)$,

$$
\begin{aligned}
f_{2,inj}(x_i) &= \mathbb{E}_{x'}[f_2(x')|f_1(x') = f_1(x_i)] \\
&= \mathbb{E}_{x'}[f_2(x')|f_1(x') = f_1(x_j)] \\
&= f_{2,inj}(x_j)
\end{aligned}
$$

Next, we need to prove the integrability of $f_{2,inj}(\cdot)$ and $f_{2,skewsym\text{-}inj}(\cdot)$. In Proposition 4.6, we assume $f_1(\cdot)$ and $f_2(\cdot)$ are square integrable on the c set $\mathcal{X}$. Square integrability on compact set indicates function boundedness (i.e. $\exists C > 0, \forall x \in \mathcal{X}, |f_1(x)|, |f_2(x)| < C$).

$f_{2,inj}(x) = \mathbb{E}_{x'}[f_2(x')|f_1(x') = f_1(x)]$ is the expectation of a square integrable function on a subset of X ($f_1^{-1}(f_1(x))$), calculate the square integral of $f_{2,inj}(\cdot)$ on this subset:

$$\begin{aligned}\int_{x\in f_1^{-1}(f_1(x))} f_{2,inj}^2(x)dx &= \int_{x\in f_1^{-1}(f_1(x))} \mathbb{E}_{x'}^2[f_2(x')|f_1(x') = f_1(x)]dx \\ &\leq \int_{x\in f_1^{-1}(f_1(x))} \mathbb{E}_{x'}[f_2^2(x')|f_1(x') = f_1(x)]dx \\ &= \int_{x\in f_1^{-1}(f_1(x))} f_2^2(x)dx\end{aligned}$$

where the inequality comes from variance-mean relationship (i.e. $\mathbb{E}^2[x] = \mathbb{E}[x^2] - \text{Var}(x)$).

For each subset of X, the square integral of $f_{2,inj}(\cdot)$ is less or equal to that of $f_2(\cdot)$. Then for the whole set X, we have

$$\int_{x\in\mathcal{X}} f_{2,inj}^2(x)dx \leq \int_{x\in\mathcal{X}} f_2^2(x)dx$$

With $f_2(\cdot) \in \mathcal{F}$ being square integrable on $\mathcal{X}$,$f_{2,inj}(\cdot)$ is also square integrable on $\mathcal{X}$ (so as $f_{2,skewsym\text{-}inj}(\cdot) = f_2(\cdot) - f_{2,inj}(\cdot)$ because the square integrability is preserved under subtraction/sum). Therefore, we have $f_{2,inj}(\cdot) \in \mathcal{F}_{inj}(f_1)$. For $f_{2,skewsym\text{-}inj}(x)$, $\forall r \in \mathbb{R}$,

$$\begin{aligned}&\mathbb{E}_{x'}[f_{2,skewsym\text{-}inj}(x')|f_1(x') = r] \\ &= \mathbb{E}_{x'}[f_2(x') - \mathbb{E}_{x''}[f_2(x'')|f_1(x'') = r]|f_1(x') = r] \\ &= \mathbb{E}_{x'}[f_2(x')|f_1(x') = r] - \mathbb{E}_{x''}[f_2(x'')|f_1(x'') = r] \\ &= 0\end{aligned}$$

Therefore, we have $f_{2,skewsym\text{-}inj} \in \mathcal{F}_{skewsym\text{-}inj}(f_1)$. $\forall f_1(\cdot), f_2(\cdot) \in \mathcal{F}$, it can be decomposed into $f_{2,skewsym\text{-}inj} \in \mathcal{F}_{skewsym\text{-}inj}(f_1)$ and $f_{2,inj} \in \mathcal{F}_{inj}(f_1)$. □

A.2. Justification of Assumption 4.12

In the Theorem 4.9, we assume the found $h(\cdot)$ is functional injective and relative continuous to $g(\cdot)$, the states within the preimage of $g(\cdot)$ will have similar values under the $h(\cdot)$ function mapping as well. Then we can also approximate $h(\cdot)$ with Taylor expansion in $g_{\mathcal{X}}^{-1}(g(x), \epsilon)$ and higher-order terms can be neglected when ϵ is small. The following approximation is for 1D scenario but can be extended to multi-dimension x. We also only consider single neighborhood cases, if there exist multiple neighborhoods, we can take the supreme as the upper bound of the LHS in Assumption 4.12.

When ϵ is small, the preimage including x can be estimated by $g_{\mathcal{X}}^{-1}(g(x), \epsilon) \approx [x - \epsilon/(g'(x)), x + \epsilon/(g'(x))]$, where $g'(x)$ denotes the derivative of $g(x)$.

$$\int_{x'\in g_{\mathcal{X}}^{-1}(g(x),\epsilon)} (h(x) - h(x'))dx' \approx \int_{x-\epsilon/g'(x)}^{x+\epsilon/g'(x)} (h(x) - h(x'))dx'$$

we need to consider 2nd order term because the first order integrates to 0

$$\begin{aligned}&\approx \int_{x-\epsilon/g'(x)}^{x+\epsilon/g'(x)} (h'(x)(x' - x) + \frac{1}{2}h''(x)(x' - x)^2)dx' \\ &= [h'(x)\xi^2 + \frac{1}{6}h''(x)\xi^3]\Big|_{\xi=-\epsilon/g'(x)}^{\xi=+\epsilon/g'(x)} = \frac{1}{3}\frac{h''(x)\epsilon^3}{g'(x)^3}\end{aligned}$$

$$\int_{x' \in g_{\mathcal{X}}^{-1}(g(x),\epsilon)} (h(x) - h(x'))^2 dx'$$

we can drop the 2nd order term

$$\approx \int_{x-\epsilon/g'(x)}^{x+\epsilon/g'(x)} (h'(x)(x'-x))^2 dx' = [h''(x)\xi^3]\Big|_{\xi=-\epsilon/g'(x)}^{\xi=+\epsilon/g'(x)}$$
$$= \frac{2}{3}\frac{(h'(x))^2\epsilon^3}{g'(x)^3}$$

$$\frac{\int_{x' \in g_{\mathcal{X}}^{-1}(g(x),\epsilon)} (h(x) - h(x'))dx'}{\int_{x' \in g_{\mathcal{X}}^{-1}(g(x),\epsilon)} (h(x) - h(x'))^2 dx'} \approx \frac{1}{3}\frac{h''(x)\epsilon^3}{g'(x)^3} \Big/ \frac{2}{3}\frac{(h'(x))^2\epsilon^3}{g'(x)^3}$$
$$= \frac{1}{2}\frac{h''(x)}{h'(x)^2} \leq \max_{x\in\mathcal{X}} \frac{1}{2}\frac{h''(x)}{h'(x)^2} = \text{Constant}$$

Then we have $LHS \approx C_1\epsilon \cdot$ Constant $\to 0$ when $\epsilon \to 0$, therefore, there always exists a small ϵ to satisfy this condition.

Notice the above approximation and constant upper bound implicitly lie on that $|g'(x)| \neq 0$ (thus $|h'(x)| \neq 0$ by relative Lipschitz continuity in Assumption 4.10). When these derivatives are 0, we can use high order derivatives to replace them (Higher order non-zero derivative always exists as $g(\cdot)$ is non-constant and smooth).

A.3. Proof of Theorem 4.9

Proof. Let the perturbed $h(x) = \lim_{\delta\to 0} h_1(x) + \delta h_2(x)$, where $h_1(\cdot) \in F_{inj}(g)$ and $h_2(\cdot) \in F_{skewsym\text{-}inj}(g)$. Let $\mathcal{L}$ denote the integral square ratio loss $\mathcal{L}_{ISQL}$ for notation simplicity. For arbitrary fixed $h_1(\cdot), h_2(\cdot)$ and small ϵ, consider $\mathcal{L}$ being a function of δ: $\mathcal{L}(h(\cdot),\epsilon) = \mathcal{L}(\delta)$, our proof goal is $\frac{dL}{d\delta} = 0$ and $\frac{d^2L}{d\delta^2} > 0$. Hence, we can claim if we found $h(\cdot) = h_1(\cdot) \in F_{inj}(g)$ and satisfies Assumption 4.10, Assumption 4.11 and 4.12, there exists $\overline{\delta} > 0, s.t. \forall \delta < \overline{\delta}, h_2(\cdot) \in \mathcal{F}_{skewsym\text{-}inj}, \mathcal{L}(h_1(\cdot) + \delta h_2(\cdot)) > \mathcal{L}(h_1(\cdot))$. Then $h(\cdot)$ is a directional local minimum point against any perturbation on $F_{skewsym\text{-}inj}(g)$. We prove the existence of $\overline{\delta}$ by taking $\delta \to 0$.

By Definition 4.4, if $h_2(\cdot) \in F_{skewsym\text{-}inj}(g), \forall r \in \mathbb{R}$, we have $\mathbb{E}_x[h_2(x)|g(x) = r] = 0$. Suppose we only perturb $h(\cdot)$ by δ at single interval where $g(x) = r$, this will violate the property of $h_2(\cdot)$, because the expectation of $h_2(\cdot)$ on $g^{-1}(r)$ will also be non-zero. To compensate the violation, we need to simultaneously perturb $h(\cdot)$ on other intervals within $g^{-1}(r)$, subject to $\mathbb{E}_x[h_2(x)|g(x) = r] = 0$. In optimal control theory (Liberzon, 2011), to include a larger family of control perturbations (e.g. discontinuous control input) and maintain the constraint of the right endpoint, a typical approach is to add two "needle" perturbations, where the control input is perturbed at two intervals. Inspired by this approach, we draw two non-overlapping intervals with the same volume from the preimage $g^{-1}(r)$, and perturb them by an inverse pair of magnitude. This symmetric perturbation function belongs to $F_{skewsym\text{-}inj}(g)$ as its expectation on $g^{-1}(r)$ is always zero. This "needle" pair can be viewed as a basic perturbation in $F_{skewsym\text{-}inj}(g)$ and any $h_2(\cdot) \in F_{skewsym\text{-}inj}(g)$ can be decomposed into these pairs.

We formulate this perturbation as following: for certain pre-image $g^{-1}(r)$, we choose two intervals $\mathcal{X}_i, \mathcal{X}_j \subset g^{-1}(r), \int_{x\in\mathcal{X}_i} 1dx = \int_{x\in\mathcal{X}_i} 1dx = \tau > 0, \mathcal{X}_i \cap \mathcal{X}_j = \emptyset$. Because we will append another infinitesimal coefficient δ ahead of $h_2(\cdot)$, the absolute magnitude of the perturbation function (or the "needle height") does not matter. Without loss of generality, let the needle function be:

$$h_2(x) = \begin{cases} +1 & \text{if } x \in \mathcal{X}_i \\ -1 & \text{if } x \in \mathcal{X}_j \\ 0 & \text{otherwise.} \end{cases}$$

To simplify notation, let $h_1(x) = s, \forall x \in g^{-1}(r)$, $g^{-1}(g(x_i),\epsilon) = \overline{g}$. Then the perturbed function is

$$h(x) = h_1(x) + \delta h_2(x) = \begin{cases} h_1(x) + \delta = s + \delta & \text{if } x \in \mathcal{X}_i \\ h_1(x) - \delta = s - \delta & \text{if } x \in \mathcal{X}_j \\ h_1(x) & \text{otherwise.} \end{cases}$$

$$
\begin{aligned}
\mathcal{L} = \mathcal{L}(h_1) &= \int_{x\in\mathcal{X}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx \\
&= \int_{x\in\mathcal{X}\setminus\overline{g}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx + \int_{\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx \\
&+ \int_{x\in\mathcal{X}_i} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx + \int_{x\in\mathcal{X}_j} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L} + \delta\mathcal{L} = \mathcal{L}(h) &= \mathcal{L}(h_1+\delta h_2) \\
&= \int_{x\in\mathcal{X}\setminus\overline{g}} \frac{\int_{x'\in\overline{g}}(h(x)-h(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h(x)-h(x''))^2dx''}dx + \int_{\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}} \frac{\int_{x'\in\overline{g}}(h(x)-h(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h(x)-h(x''))^2dx''}dx \\
&+ \int_{x\in\mathcal{X}_i} \frac{\int_{x'\in\overline{g}}(h(x)-h(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h(x)-h(x''))^2dx''}dx + \int_{x\in\mathcal{X}_j} \frac{\int_{x'\in\overline{g}}(h(x)-h(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h(x)-h(x''))^2dx''}dx
\end{aligned}
$$

Let $\overline{\mathcal{L}}_1, \overline{\mathcal{L}}_2, \overline{\mathcal{L}}_3, \overline{\mathcal{L}}_4$ denote the 4 terms in the above bracket, address each term:

$$
\begin{aligned}
&\overline{\mathcal{L}}_1 + \delta\overline{\mathcal{L}}_1 \\
&= \int_{x\in\mathcal{X}\setminus\overline{g}} \frac{\int_{x'\in\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x'))^2dx' + \int_{x'\in\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x'))^2dx'}{\int_{x''\in\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x''))^2dx'' + \int_{x''\in\mathcal{X}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x''))^2dx''}dx \\
&= \int_{x\in\mathcal{X}\setminus\overline{g}} \frac{2\delta^2\tau + \int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x''))^2dx'' - \int_{x''\in\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x''))^2dx'' + \int_{x''\in\mathcal{X}}(h(x)-h(x''))^2dx''}dx \\
&= \int_{x\in\mathcal{X}\setminus\overline{g}} \frac{2\delta^2\tau + \int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{[-2(h_1(x)-s)^2 + (h_1(x)-s-\delta)^2 + (h_1(x)-s+\delta)^2]\tau + \int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx \\
&= \int_{x\in\mathcal{X}\setminus\overline{g}} \frac{2\delta^2\tau + \int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{2\delta^2\tau + \int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx
\end{aligned}
$$

neglect $\mathcal{O}(\delta^2)$ terms

$$
\approx \int_{x\in\mathcal{X}\setminus\overline{g}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}[1 - \frac{2\delta^2\tau}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''} + \frac{2\delta^2\tau}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx''}]dx
$$

$$
\begin{aligned}
&\overline{\mathcal{L}}_2 + \delta\overline{\mathcal{L}}_2 \\
&= \int_{x\in\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}} \frac{\int_{x'\in\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x'))^2dx' + \int_{x'\in\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x'))^2dx'}{\int_{x''\in\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x''))^2dx'' + \int_{x''\in\mathcal{X}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}}(h(x)-h(x''))^2dx''}dx \\
&= \int_{x\in\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}} \frac{2\delta^2\tau + \int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{2\delta^2\tau + \int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx
\end{aligned}
$$

neglect $\mathcal{O}(\delta^2)$ terms

$$
= \int_{x\in\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}[1 - \frac{2\delta^2\tau}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''} + \frac{2\delta^2\tau}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx''}]dx
$$

$$
\overline{\mathcal{L}}_3 + \delta\overline{\mathcal{L}}_3 = \int_{x\in\mathcal{X}_i} \frac{\int_{x'\in\overline{g}}(h(x)-h(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h(x)-h(x''))^2dx''}dx
$$

$$= \int_{x\in\mathcal{X}_i} \frac{\int_{x'\in\overline{g}}(s+\delta-h(x'))^2dx'}{\int_{x''\in\mathcal{X}}(s+\delta-h(x''))^2dx''}dx$$

perturbation at $x \in \mathcal{X}_i$ will also affect $h(x')$ and $h(x'')$ terms, but this deviation will be canceled by the deviation of $h(x)$

$$= \int_{x\in\mathcal{X}_i} \frac{\int_{x'\in\overline{g}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}}[(s-h_1(x'))^2+2\delta(s-h_1(x'))+\delta^2]dx' + \int_{x'\in\mathcal{X}_j}[(s+\delta-s+\delta)^2]dx'}{\int_{x''\in\mathcal{X}\setminus\{\mathcal{X}_i,\mathcal{X}_j\}}[(s-h_1(x''))^2+2\delta(s-h_1(x''))+\delta^2]dx'' + \int_{x''\in\mathcal{X}_j}[(s+\delta-s+\delta)^2]dx''}dx$$

using $h_1(x) = s, \forall x \in \mathcal{X}_i \bigcup \mathcal{X}_j \subset g^{-1}(r)$ and neglect higher order terms with $\lim_{a\to 0}\frac{1}{1+a} \approx 1-a+a^2$, let $|\mathcal{X}| = \int_{x\in\mathcal{X}} 1dx, |\overline{g}| = \int_{x\in\overline{g}} 1dx$ be the measure of $\mathcal{X}$ and $\overline{g}$

$$\approx \int_{x\in\mathcal{X}_i} \frac{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''} * [1+\frac{2\delta\int_{x'\in\overline{g}}(s-h_1(x'))dx' + (|\overline{g}|+2\tau)\delta^2}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}]$$

$$* [1-\frac{2\delta\int_{x''\in\mathcal{X}}(s-h_1(x''))dx'' + (|\mathcal{X}|+2\tau)\delta^2}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''} + 4\delta^2\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}^2]dx$$

Apply the same treatment to $\overline{\mathcal{L}}_4 + \delta\overline{\mathcal{L}}_4$, notice adding two terms will cancel δ terms

$$\overline{\mathcal{L}}_3 + \delta\overline{\mathcal{L}}_3 + \overline{\mathcal{L}}_4 + \delta\overline{\mathcal{L}}_4$$

$$\approx 2\tau\frac{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''} * [1+\frac{(|\overline{g}|+2\tau)\delta^2}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'} - \frac{(|\mathcal{X}|+2\tau)\delta^2}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}$$

$$+ 4\delta^2\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}^2 - 4\delta^2\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}]$$

$$= 2\tau\frac{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''} * [1+\frac{|\overline{g}|\delta^2}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'} - \frac{|\mathcal{X}|\delta^2}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}$$

$$+ 4\delta^2\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}^2 - 4\delta^2\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}]$$

$$+ \int_{x\in\{\mathcal{X}_i,\mathcal{X}_j\}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}[-\frac{2\delta^2\tau}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''} + \frac{2\delta^2\tau}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx''}]$$

sum up all terms:

$$\sum_{k=1}^{4}[\overline{\mathcal{L}}_k + \delta\overline{\mathcal{L}}_k]$$

$$= \int_{x\in\mathcal{X}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx - 2\delta^2\tau\int_{x\in\mathcal{X}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''} * \frac{1}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2}dx$$

$$+ 2\delta^2\tau\int_{x\in\overline{g}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''} * \frac{1}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}dx$$

$$+ 2\frac{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''} * [\frac{|\overline{g}|\delta^2\tau}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'} - \frac{|\mathcal{X}|\delta^2\tau}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}$$

$$+ 4\delta^2\tau\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}^2 - 4\delta^2\tau\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}]$$

$$= \sum_{k=1}^{4}\overline{\mathcal{L}}_k$$

$$+ 2\delta^2\tau\Bigg[-\int_{x\in\mathcal{X}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''} * \frac{1}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx \cdots\cdots\cdots\cdots ①$$

$$+\int_{x\in\overline{g}}\frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}*\frac{1}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}dx\cdots\cdots\cdots\cdots\cdots ②$$

$$+\frac{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}*[\frac{|\overline{g}|}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}-\frac{|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\cdots\cdots ③$$

$$+4\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}(\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}-\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'})]\Bigg]\cdots\cdots ④$$

There is no first order δ term in the perturbed loss function ($\frac{d\mathcal{L}}{d\delta}=0$), then we proved $h(\cdot)$ is a stationary point to any perturbation from $F_{skewsym\text{-}inj}(g)$.

Before we evaluate the second order coefficients inside the big bracket, we first derive two useful inequalities regarding the generalized variance:

Recall we assume $h_1(\cdot)$ is C_1 relative Lipschitz continuous to $g(\cdot)$, $\forall x\in\mathcal{X}_i\cap\mathcal{X}_j, x'\in g_{\mathcal{X}}^{-1}(r,\epsilon)$, then

$$\begin{aligned}(h_1(x)-h_1(x'))^2&\leq C_1^2(g(x)-g(x'))^2\\&\leq C_1^2\epsilon^2\end{aligned}$$

where the second inequality comes from the definition of $g_{\mathcal{X}}^{-1}(r,\epsilon)$.

Then $\forall x''\in\mathcal{X}$, we have

$$\begin{aligned}\frac{\int_{\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{\overline{g}}1dx'}&\leq C_1^2\epsilon^2\\&<\frac{1}{5C_2}\frac{\int_{\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}{\int_{\mathcal{X}}1dx''}\\&<\frac{\int_{\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}{5\int_{\mathcal{X}}1dx''}\end{aligned}$$

where the first inequality comes from averaging $(h_1(x)-h_1(x'))^2$ term on $\overline{g}$, the second inequality comes from the Assumption 4.11.

We have the local generalized variance less than the global generalized variance $\forall x\in\mathcal{X}_i\cap\mathcal{X}_j, x'\in g_{\mathcal{X}}^{-1}(r,\epsilon), x''\in\mathcal{X}$:

$$\frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{|\overline{g}|}<\frac{1}{5C_2}\frac{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}{|\mathcal{X}|}<\frac{1}{C_2}\frac{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}{|\mathcal{X}|}\tag{7}$$

$$\frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{|\overline{g}|}<\frac{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}{5|\mathcal{X}|}<\frac{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}{|\mathcal{X}|}\tag{8}$$

Now we are ready to compare each pair of terms:

• term ①, ②:

$$\int_{x\in\mathcal{X}}\frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}*\frac{1}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx$$

by Equation (7)

$$<\int_{x\in\mathcal{X}}\frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}*\frac{1}{C_2}\frac{1}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}\frac{|\overline{g}|}{|\mathcal{X}|}dx$$

$$= \int_{x\in\mathcal{X}} \frac{1}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}\frac{1}{C_2}\frac{|\overline{g}|}{|\mathcal{X}|}dx$$

$$\leq |\mathcal{X}|\frac{1}{|\mathcal{X}|\min_{x\in\mathcal{X}}\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}\frac{1}{C_2}\frac{|\overline{g}|}{|\mathcal{X}|}$$

by Assumption 4.11

$$= |\mathcal{X}|\frac{1}{|\mathcal{X}|\max_{x\in\mathcal{X}}\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}\frac{|\overline{g}|}{|\mathcal{X}|}$$

$$< \int_{x\in\overline{g}} \frac{1}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''}dx$$

$$= \int_{x\in\overline{g}} \frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}{\int_{x''\in\mathcal{X}}(h_1(x)-h_1(x''))^2dx''} * \frac{1}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}dx$$

Then we have ① + ② > 0

• term ③: The following equation is directly from taking the inverse of Equation (8) when $h_1(x) = s$

$$\frac{|\overline{g}|}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'} > \frac{|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}$$

• term ④:

$$\begin{aligned}
&\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''} / \frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'} \\
&= \frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''/|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''/|\mathcal{X}|} / \frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'/|\overline{g}|}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'/|\overline{g}|} \\
&= \frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''/|\mathcal{X}|}{\int_{x'\in\overline{g}}(s-h_1(x'))dx'/|\overline{g}|} / \frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''/|\mathcal{X}|}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'/|\overline{g}|} \qquad (9)
\end{aligned}$$

by Equation (8)

$$> \frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''/|\mathcal{X}|}{\int_{x'\in\overline{g}}(s-h_1(x'))dx'/|\overline{g}|}$$

$\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''/|\mathcal{X}|$ can be seen as the difference between s and the mean of h_1 over $\mathcal{X}$.

Now consider two scenarios,

1. If $\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''/|\mathcal{X}| > C_1\epsilon$, then we have

$$\begin{aligned}
\text{Equation (9)} &> \frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''/|\mathcal{X}|}{\int_{x'\in\overline{g}}C_1|r-g(x')|/|\overline{g}|} \\
&> \frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))/|\mathcal{X}|}{C_1\epsilon} \\
&> 1
\end{aligned}$$

therefore,

$$4\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\left(\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''} - \frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}\right) > 0$$

With all the summations of each pair of terms in the big bracket being positive, the proof is complete.

2. If $\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''/|\mathcal{X}| \leq C_1\epsilon$, we neglect the positive term $(4\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''})$ in ④ and compare the negative term $(-4\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'})$ with the negative terms $(-\frac{|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''})$ in ③. We have

$$
\begin{aligned}
&4\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}\\
&\leq 4\frac{C_1\epsilon|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}\\
&\to 4\frac{C_1\epsilon|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))dx'}{\int_{x'\in\overline{g}}(h_1(x)-h_1(x'))^2dx'}
\end{aligned}
\tag{10}
$$

by Assumption 4.12

$$
\leq 4\frac{|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}
$$

Considering ③ + ④ together, we have

$$
\begin{aligned}
&\frac{|\overline{g}|}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}-\frac{|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\\
&+4\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}-4\frac{\int_{x''\in\mathcal{X}}(s-h_1(x''))dx''}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}\frac{\int_{x'\in\overline{g}}(s-h_1(x'))dx'}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}
\end{aligned}
$$

neglecting the third term (because it is positive) and by Equation (10)

$$
> \frac{|\overline{g}|}{\int_{x'\in\overline{g}}(s-h_1(x'))^2dx'}-\frac{5|\mathcal{X}|}{\int_{x''\in\mathcal{X}}(s-h_1(x''))^2dx''}
$$

by Equation (8)

$$
> 0
$$

Under this scenario, all the summations of each pair of terms in the big bracket are also positive, then we have $\frac{d^2\mathcal{L}}{d\delta^2} > 0$, the proof is complete.

□

B. Supplementary Experiments

B.1. Experiment Details

Table 4. Experimental details

	Ideal spring mass	Chemical kinematics	Kepler system	Heat equation
Dimension	2	2	4	101
Dynamics	$\dot{x}[1] = x[2]$ $\dot{x}[2] = -x[1]$	$\dot{x}[1] = -\kappa_1 x[1] + \kappa_2 x[2]$ $\dot{x}[2] = +\kappa_1 x[1] - \kappa_2 x[2]$	$\dot{x}[1] = x[3]$ $\dot{x}[2] = x[4]$ $r = \sqrt{x[1]^2 + x[2]^2}$ $\dot{x}[3] = -\frac{x[1]}{r^3}$ $\dot{x}[4] = -\frac{x[2]}{r^3}$	$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial y^2}$
Conservations	$x[1]^2 + x[2]^2 = C$	$x[1] + x[2] = C$	$\frac{x[3]^2+x[4]^2}{2} - \frac{1}{\sqrt{x[1]^2+x[2]^2}} = C_1$ $x[1]x[4] - x[2]x[3] = C_2$	$\int U dx = C$
Learning rate	1e-3	1e-3	1e-3	1e-3
# of training traj.	50	50	50	100
Training traj. length (s)	10	10	10	10
# of points in training traj.	100	100	100	100
Training data noise std.	0.01	0.01	0.01	0.01
# of eval traj.	10	10	10	10
Eval. traj. length (s)	50	10	5	1
Constrastive Learning Batch size	10	10	10	10
Constrastive Learning Epochs	1000	1000	1000	1000
Dynamics Learning Batch size	100	100	100	100
Dynamics Learning Epochs	1000	1000	1000	1000

All the dynamical modeling and contrastive learning neural networks mentioned in the paper are fully connected neural networks, with 1 hidden layer of 100 neurons and tanh activations. The autoencoder used for the heat equation is also fully connected, both encoder and decoder use 2 hidden layers (32,16 neurons) and tanh activations. We conduct all the experiments on a single 2080Ti GPU. The additional experiment details are listed in Table 4.

B.2. Comparison with Other Dynamics Model Baselines

Table 5. Simulation error comparison with classical (non-DNN based) method (SINDy (Brunton et al., 2016))

Task	Mean square error		Violation of conservation laws	
	ConCerNet	SINDy	ConCerNet	SINDy
Ideal spring mass	9.2e-3 ± 5.6e-3	**1.3e-4±7.6e-5**	**1.4e-4 ± 7.4e-5**	1.5e-4 ± 5.6e-5
Chemical reaction	5.9e-3± 3.7e-3	**1.4e-3 ± 3.6e-4**	5.6e-3 ± 5.4e-3	**6.6e-7 ± 3.2e-7**
Kepler system	**0.140 ± 0.067**	1.1 ± 0.47	**5.2e-3 ± 4.2e-3**	10.2 ± 6.8

We provide a comparison between our model and one of the most popular classical non-DNN based system identification methods called SINDy (Sparse Identification of Nonlinear Dynamics, Brunton et al. (2016)) on the performance of dynamics learning. We use the PySINDy package (Kaptanoglu et al., 2021) which automatically selects the sparse polynomial basis functions. As expected, SINDy outperforms our DNN based approach on the first two linear examples because they fit in the polynomial basis functions and SINDy is more compact than DNN models. However, SINDy does not perform well on the Kepler example requiring more complex basis functions. Besides, SINDy cannot directly handle the heat equation problem due to the large space dimension.

Table 6. Simulation error comparison with DNN-based prior work (HNN (Greydanus et al., 2019))

Task	Mean square error			Violation of conservation laws		
	Baseline NN	ConCerNet	HNN	Baseline NN	ConCerNet	HNN
Ideal spring mass	0.022 ± 0.023	**9.2e-3 ± 5.6e-3**	0.016±0.024	0.012 ± 0.032	1.4e-4 ± 7.4e-5	**2.7e-5 ± 1.9e-5**
Kepler system	0.794 ± 0.476	**0.140 ± 0.067**	0.174 ± 0.126	0.016 ± 0.011	**5.2e-3 ± 4.2e-3**	0.019 ± 0.014

Table 7. Average R^2 comparison with prior work (HNN (Greydanus et al., 2019)) in conservation property learning

Task	Conservation	ConCerNet	HNN
Ideal spring mass system	$x[1]^2 + x[2]^2 = C$	0.997 ± 0.001	0.9995 ± 0.0002
Kepler system	$\frac{x[3]^2+x[4]^2}{2} - \frac{1}{\sqrt{x[1]^2+x[2]^2}} = C_1$	0.858 ± 0.104	0.983 ± 0.008
	$x[1]x[4] - x[2]x[3] = C_2$	0.994 ± 0.001	0.906 ± 0.032

Table 6 and Table 7 show the result comparison between our proposed method and another DNN-based method HNN (Greydanus et al., 2019) on the two examples. HNN is not applicable to the other two experiments, because they are not Hamiltonian systems.

For dynamical system simulation, the two methods show similar performance, and their conservation and coordinate errors are much smaller than the vanilla neural network.

For contrastive learning with a single conservation value (e.g. ideal spring mass system), HNN performs slightly better than ConCerNet. For systems with multiple conservation laws, there is no universal metric for conservation because any combination of conservation equations is conserved. ConCerNet empirically learns the Angular momentum function and HNN learns the Hamiltonian value.

We do not provide comparison experiments with prior works that claim to find the exact solution for multiple conservation laws, as these methods rely on human knowledge of prior information such as fitting function classes. Using the Kepler system as an example, because any combination of angular momentum and energy conservation is also conserved, a natural question will be why the method could find the individual solution instead of a summation of the two. Due to this limitation, it is difficult to evaluate the practical performance of the method because the user/problem-dependent knowledge is unclear.

B.3. Model Performance with Different Noise Levels

One might question the ConCerNet performance under large observation noises. We extend the model comparison experiment of Table 3 into various noise settings and show the results in Table 8. As the noise level grows, the ConCerNet performance is compromised. However, the baseline NN performance decays much faster than ConCerNet. Especially for potentially unstable systems like the ideal spring mass and Kepler system, both coordinate error and conservation violations could grow exponentially to testing simulation time when training data is corrupted. ConCerNet provides a safety regulation for this issue. Regarding the Heat equation example where the system stabilizes during the simulation, we observe similarly bad performance between the two methods under high noises. This issue might come from the autoencoder learning, as the low representation space cannot interpret the high random noise observation. In the low noise setting, our method can still beat the baseline NN by a large margin.

Table 8. Simulation error comparison with different noise level

Task	Noise std.	Mean square error		Violation of conservation laws	
		Baseline NN	ConCerNet	Baseline NN	ConCerNet
Ideal spring mass	0.0	3.7e-3 ± 2.8e-3	**6.1e-4 ± 5.1e-4**	0.014 ± 0.020	**3.5e-5 ± 4.8e-5**
	0.01	0.022 ± 0.023	**9.2e-3 ± 5.6e-3**	0.012 ± 0.032	**1.4e-4 ± 7.4e-5**
	0.1	0.53 ± 0.180	**0.45 ± 0.12**	0.081 ± 0.033	**1.3e-3 ± 1.0e-3**
	1.0	21.2 ± 39.1	**0.71 ± 0.28**	7.5e4 ± 1.4e4	**1.45 ± 2.54**
Chemical reaction	0.0	0.01 ± 0.01	**7.3e-3 ± 9.7e-4**	0.032 ± 0.040	**5.5e-3 ± 4.1e-3**
	0.01	9.4e-3 ± 5.8e-3	**5.9e-3 ± 3.7e-3**	0.019 ± 0.026	**5.6e-3 ± 5.4e-3**
	0.1	0.195 ± 0.112	**0.038 ± 0.035**	0.63 ± 1.02	**0.04 ± 0.04**
	1.0	2.16 ± 0.47	**0.94 ± 0.72**	7.2 ± 2.6	**1.92 ± 1.67**
Kepler system	0.0	0.13 ± 0.08	**0.10 ± 0.10**	5.9e-3 ± 1.7e-3	**3.7e-4 ± 3.1e-4**
	0.01	0.794 ± 0.476	**0.14 ± 0.07**	0.016 ± 0.011	**5.2e-3 ± 4.2e-3**
	0.1	0.89 ± 0.22	**0.73 ± 0.27**	0.58 ± 0.77	**0.50 ± 0.52**
	1.0	5.27 ± 3.23	**1.21 ± 0.52**	144 ± 90.2	**1.77 ± 1.51**
Heat equation	0.0	1.2e-4 ± 2.2e-5	**7.2e-5 ± 1.3e-5**	0.43 ± 0.20	**0.027 ± 0.040**
	0.01	7.3e-5 ± 2.6e-5	**2.8e-5 ± 1.3e-5**	0.09 ± 0.07	**0.013 ± 0.008**
	0.1	2.5e-4 ± 3.9e-5	**2.4e-4 ± 9.7e-5**	**0.05 ± 0.03**	0.11 ± 0.09
	1.0	**0.013 ± 6.2e-4**	0.013 ± 2.7e-4	**2.95 ± 0.76**	3.01 ± 0.73

We also test the performance of contrastive conservation learning under varying noise levels and record the R^2 coefficient of

the linear regression to the exact conservation function in Table 9. As expected, the R^2 coefficient decreases with larger noise, but a strong linear association exists for noise standard deviation under 0.1. For the heat equation problem with a noise standard deviation of 1.0, we notice the autoencoder is incapable to capture the major modes in the system, this partially explains the bad performance of both conservation learning and dynamics learning under high noises.

Table 9. Averaged learned invariant fitting error vs noise level

Task/Conservation	Noise std.	R^2	Task/Conservation	Noise std.	R^2
Ideal spring mass Energy conservation	0.0	0.998	Kepler system Energy conservation	0.0	0.828
	0.01	0.997		0.01	0.858
	0.1	0.986		0.1	0.800
	1.0	0.061		1.0	0.154
Chemical reaction mass conservation	0.0	1-2e-5	Kepler system Angular momentum conservation	0.0	0.997
	0.01	1-2e-6		0.01	0.994
	0.1	1-9e-6		0.1	0.812
	1.0	0.984		1.0	0.175
Heat equation Energy conservation	0.0	1-7e-4			
	0.01	1-5e-4			
	0.1	0.875			
	1.0	0.095			

B.4. Data Efficiency of the Conservation Learning

We test the data efficiency of the contrastive conservation learning and record the R^2 coefficient of the linear regression to the exact conservation function in Table 10.

Table 10. Averaged learned invariant fitting error vs trajectory number and points per trajectory

Task/Conservation	traj. / points per traj.	5	10	20	40
Ideal spring mass Energy conservation	5	0.547	0.828	0.971	0.989
	10	0.971	0.995	0.997	0.997
	20	0.996	0.996	0.987	0.994
	40	0.997	0.997	0.998	0.997
	80	0.997	0.994	0.992	0.997
Chemical reaction mass conservation	5	1-4.8e-4	1-3.3e-4	1-1.5e-5	1-3.2e-5
	10	1-1.2e-4	1-1.2e-4	1-5.6e-5	1-6.3e-3
	20	1-7.1e-4	1-2.3e-4	1-1.3e-6	1-1.3e-6
	40	1-8.5e-5	1-1.1e-4	1-2.5e-5	1-1.9e-7
	80	1-4.1e-5	1-7.8e-6	1-4.9e-6	1-1.1e-6
Kepler system Energy conservation	5	0.007	0.300	0.170	0.477
	10	0.290	0.346	0.503	0.763
	20	0.369	0.430	0.768	0.873
	40	0.393	0.748	0.815	0.863
	80	0.492	0.698	0.817	0.911
Kepler system Angular momentum conservation	5	0.0005	0.367	0.194	0.615
	10	0.303	0.386	0.579	0.902
	20	0.380	0.530	0.879	0.922
	40	0.570	0.887	0.947	0.968
	80	0.515	0.816	0.956	0.985
Heat equation Energy conservation	5	0.188	0.293	0.148	0.207
	10	0.301	0.217	0.132	0.318
	20	0.280	0.312	0.409	0.438
	40	0.723	0.273	0.550	0.674
	80	0.994	0.998	0.999	0.999