Noether Networks: Meta-Learning Useful Conserved Quantities

Anonymous Author(s)
Affiliation
Address
email

Abstract

Progress in machine learning (ML) stems from a combination of data availability, computational resources, and an appropriate encoding of inductive biases. Useful biases often exploit symmetries in the prediction problem, such as convolutional networks relying on translation equivariance. Automatically discovering these useful symmetries holds the potential to greatly improve the performance of ML systems, but still remains a challenge. In this work, we focus on sequential prediction problems and take inspiration from Noether's theorem to reduce the problem of finding inductive biases to meta-learning useful conserved quantities. We propose Noether Networks: a new type of architecture where a meta-learned conservation loss is optimized inside the prediction function. We show, theoretically and experimentally, that Noether Networks improve prediction quality, providing a general framework for discovering inductive biases in sequential problems.

1 Introduction

The clever use of inductive biases exploiting symmetries has been at the heart of many landmark achievements in machine learning, such as translation invariance in CNN image classification [13], permutation invariance in Graph Neural Networks [18] for drug design [21], or roto-translational equivariance in SE3-transformers [11] for protein structure prediction. However, for data distributions of interest, there may be exploitable symmetries that are either unknown or difficult to describe with code. Progress has been made in automatically discovering symmetries, for instance for finite groups [33], but meta-learning and exploiting general continuous symmetries has presented a big challenge. This is because reasoning about symmetries often involves counter-factuals about perturbations to the data, and most ways of exploiting these have been through clever network design, which lives in discrete space.

In this work, we focus on sequential prediction problems and rely on the expressivity of conserved quantities to express symmetries in the prediction problem. This is formalized in Noether's theorem [16], which loosely states the following:

For every continuous symmetry property of a dynamical system,
there is a corresponding quantity whose value is conserved in time.

For example, consider a system of planets interacting via gravity: this system is translation invariant in all three cardinal directions (i.e. translating the entire system in the x,y, or z axis conserves the laws of motion). Noether's theorem asserts there must be a conserved quantity for each of these symmetries; in this case, linear momentum ($p_{x,y,z} = \sum_i m^{(i)} \cdot v^{(i)}_{x,y,z}$). Similarly, the system has a time-invariance (i.e. the laws of motion are the same today as they will be tomorrow). In this case, the corresponding conserved quantity is the total energy of the system.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

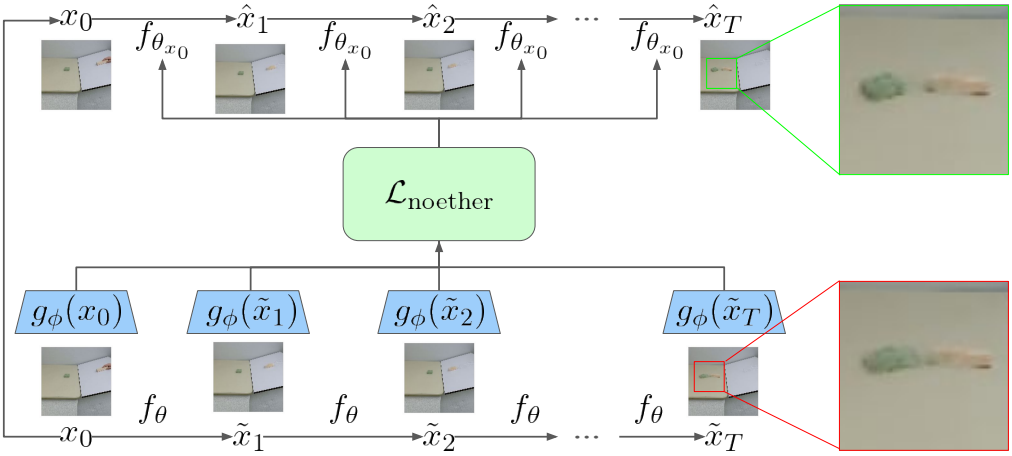

Figure 1: Noether Networks impose a conservation loss on a meta-learned embedding g_ϕ. This loss enforces different conserved quantities between predictions made by a sequential-prediction model f_θ which is then tailored to the specific query x_0. Noether Networks learn to impose their own inductive biases in order to improve final predictive performance.

Inspired by this equivalence, we propose that approximate conservation laws are a powerful paradigm for meta-learning useful inductive biases in sequential prediction problems. Whereas symmetries are global properties linked to counter-factuals about perturbing the actual data, conserved quantities can be evaluated using only the true data. This provides an immediate signal for machine learning algorithms to exploit.

Our approach involves meta-learning a parametric conservation loss function which is useful to the prediction task. We leverage the *tailoring* framework [2], which proposes to encode inductive biases by fine-tuning neural networks with hand-designed unsupervised losses inside the prediction function. In contrast to traditional auxiliary losses, which are added to the task loss during training, tailoring optimizes both during training and testing, customizing the model to each individual query to ensure there is no generalization gap for the self-supervised loss. We propose to meta-learn the self-supervised loss function and parameterize it in the form of a conservation loss; i.e. $f(x_0, \hat{x}_1, \ldots, \hat{x}_T) = \sum_{t=1}^{T} |g_\phi(x_0) - g_\phi(\hat{x}_t)|^2$. By doing so, we allow the network to discover its own inductive biases. The conservation form encodes a meta-inductive bias (inductive bias over inductive biases) which narrows the search space exponentially (in T) and simplifies the parametrisation. Inspired by Noether's theorem, we believe this formulation retains most of the expressivity for the biases we care about, while we aim to learn conservation laws not explicitly covered by the theorem, such as conservation of mass from pixels.

The main contribution of this paper is Noether Networks, an architecture class and algorithm that can automatically learn its own inductive biases in the form of meta-learned conservation loss functions. We theoretically characterize the advantages of such conservation laws as effective regularizers that constrain the learning to lower-dimensional output manifolds, lowering the generalization gap. Empirically, we first find that, when the meta-learned conservation loss takes the form of a synthesized program, Noether Networks recover known conservation laws from raw physical data. Second, we find that Noether Networks can learn conservation losses on raw videos, modestly improve the generalization of a video prediction model, especially on longer-term predictions.

2 Theoretical advantages of enforcing conservation laws

In this section, we demonstrate principled advantages of enforcing conservation laws of the form $\phi(f_\theta(x)) = \phi(x)$ by considering a special case where preimages under ϕ form affine subspaces. Our analysis shows that the dimensionality of preimages under ϕ controls the advantages of enforcing conservation laws in Noether Networks.

Let $d \geq 1$ be the dimensionality of x and y as $x, y \in \mathbb{R}^d$. We write $\phi : \mathbb{R}^d \to \mathcal{P}$ where $\mathcal{P} = \{\phi(x) : x \in \mathbb{R}^d\}$. For any $p \in \mathcal{P}$, we define the preimage of ϕ by $\phi^{-1}[\{p\}] = \{x \in \mathbb{R}^d : \phi(x) = p\}$, where ϕ is not required to be invertible. We denote by $\theta \in \Theta$ a learnable parameter vector. In this section, we consider a special case where $f_\theta(x) = x + v_\theta$ for some $v_\theta \in \mathbb{R}^d$ and there exists $A \in \mathbb{R}^{d \times m}$ such that for all $x \in \mathbb{R}^d$, $\phi^{-1}[\{\phi(x)\}] = \{x + Az : z \in \mathbb{R}^m\}$. Here, $m \in \{0, 1, \ldots, d\}$ is the dimensionality of preimages under ϕ . We denote by C the smallest upper bound on the loss value as $\mathcal{L}(f_\theta(x), y) \leq C$ (for all x, y and v). We define $\mathcal{V} = \{v_\theta \in \mathbb{R}^d : \theta \in \Theta\}$, $R = \sup_{v \in \mathcal{V}} \|v\|_2$, and $\psi(v) = \mathbb{E}_{x,y}[\mathcal{L}(f(x, v), y)] - \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f(x_i, v), y_i)$, where $f(x, v) = x + v$. The Lipschitz constant of a function ψ is the smallest real number $\zeta \geq 0$ such that, for all v and v' in $\mathcal{V}$, $|\psi(v) - \psi(v')| \leq \zeta \|v - v'\|_2$. Finally, we define the generalization gap by

$$G(\theta) = \mathbb{E}_{x,y}[\mathcal{L}(f_\theta(x), y)] - \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f_\theta(x_i), y_i).$$

Theorems 1–2 show that enforcing conservation laws of $\phi(f_\theta(x)) = \phi(x)$ is advantageous when $m < d$, where d is the dimensionality of x, and m is the dimensionality of the preimages under ϕ:

Theorem 1. *(without conservation laws) Let $\rho \in \mathbb{N}^+$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ over an iid draw of n examples $((x_i, y_i))_{i=1}^n$, the following holds for all $\theta \in \Theta$:*

$$G(\theta) \leq C\sqrt{\frac{d\ln(\sqrt{d}) + d\ln(2R(\zeta^{1-1/\rho})\sqrt{n}) + \ln(1/\delta)}{2n}} + \sqrt{\frac{\zeta^{2/\rho}}{n}}. \tag{1}$$

Theorem 2. *(with conservation laws) Let $\rho \in \mathbb{N}^+$. If $\phi(f_\theta(x)) = \phi(x)$ for any $x \in \mathbb{R}^d$ and $\theta \in \Theta$, then for any $\delta > 0$, with probability at least $1 - \delta$ over an iid draw of n examples $((x_i, y_i))_{i=1}^n$, the following holds for all $\theta \in \Theta$:*

$$G(\theta) \leq C\sqrt{\frac{m\ln(\max(\sqrt{m}, 1)) + m\ln(2R(\zeta^{1-1/\rho})\sqrt{n}) + \ln(1/\delta)}{2n}} + \mathbb{1}\{m \geq 1\}\sqrt{\frac{\zeta^{2/\rho}}{n}}. \tag{2}$$

The proofs of Theorems 1–2 are presented in Appendix A. As Theorems 1–2 attribute the advantage of enforcing conservation laws to the dimensionality m of preimages under ϕ, we now discuss various cases for the values of m:

- (Case of $m = d$) Let us consider an extreme scenario where the function ϕ maps all $x \in \mathbb{R}^d$ to one single point. In this scenario, the dimensionality of preimages under ϕ is maximized as $m = d$. Accordingly, the bound in Theorem 2 becomes the same as that in Theorem 1. Indeed, in this scenario, the conservation laws of $\phi(f_\theta(x)) = \phi(x)$ gives us no information, because it always holds for all $x \in \mathbb{R}^d$, even without imposing conservation laws.
- (Case of $m = 0$) Let us consider another extreme scenario where the function ϕ is invertible. In this scenario, the dimensionality of preimages under ϕ is zero as $m = 0$. Thus, imposing the condition of $\phi(f_\theta(x)) = \phi(x)$ makes the bound in Theorem 2 to be very small. Indeed, in this scenario, the condition of $\phi(f_\theta(x)) = \phi(x)$ implies that $f_\theta(x) = x$: i.e., x is not moving, and thus it is easy to generalize.
- (Case of $0 < m < d$) From the above two cases, we can see that the benefit of enforcing conservation laws comes from more practical cases in-between these, with $0 < m < d$.

In Theorem 2, the function ϕ can differ from the true function ϕ^* underling the system. This is because we analyze a standard generalization gap: i.e., the difference between the expected loss and the training loss. The cost of not using the true ϕ^* is captured in the training loss; i.e., the training loss can be large with the function ϕ that differs significantly from the true ϕ^*. Even in this case, the generalization gap can be small. For example, in the case of $m = 0$, the generalization bound is small, whereas the training loss will be large unless $x_{t+1} = x_t$. Therefore, our analysis gives us the insight on the tradeoff between the training loss and the dimensionality of preimages under ϕ.

3 Noether Networks

3.1 Tailoring and meta-tailoring to encode inductive biases

We perform a prediction-time optimization to encourage outputs to follow conservation laws using the tailoring and meta-tailoring frameworks [2]. Tailoring proposes to encode inductive biases in the

form of unsupervised losses optimized inside the prediction function. In doing so, tailoring fine-tunes the model to each query to ensure satisfies the bias for that query. For example, we may optimize for energy conservation (if we knew the formula for it) in a physics prediction problem. In meta-tailoring, we train the model to do well after the tailoring step has fine-tuned its parameters. This allows us to impose inductive biases on the test queries without the query inputs being known at training time, as in the transductive learning setting [26]. An important aspect of (meta-)tailoring is that the tailoring loss must be user-specified. This is acceptable in domains where the desired inductive bias is both known and easily encoded, but problematic when the appropriate inductive bias is more difficult to encode in the output space. Our approach can be seen as a generalization of tailoring where the unsupervised loss is meta-learned and takes the form of a conservation loss.

Note that, even though meta-tailoring and our search for a loss use the two-layer optimization typical of gradient-based meta-learning, we are still in the classical single-task single-distribution setting. In our framework, the model learns to impose its own inductive biases via a learned loss with parameters ϕ, and to leverage this loss via an adaptation of its parameters θ.

3.2 Discovering *useful* conservation laws

We build on the paradigm of conservation laws; i.e. quantities that are conserved over time. This comes with the following two problems:

1. Real data often has noise, breaking any exact conservation law. Moreover, many conservation laws are only approximately satisfied when we only have partial information of the system. For example, conservation of energy is not fully satisfied in real dissipative systems and estimating conservation of mass from pixels is bound to be inexact in the case of occlusions.
2. Optimizing for conservation alone can lead to trivial quantities, such as predicting a constant value $g_\phi(x_t) = C$.

We take the theoretical intuition from section 2 and search for *useful* conservation laws, whose (approximate) enforcing brings us closer to the true data manifold for the current sequence. Note that a useful conserved quantity doesn't need to be exactly conserved in the training data to improve prediction performance. We only need its conservation to be informative of the data manifold. This allows us to discover conservation laws even in imperfect, noisy environments. Moreover, by definition, tailoring with a trivial conserved quantity is unlikely to improve the final predictions. Finally, viewing conserved quantities as useful inductive biases matches will with machine learning, where we typically care about biases only if they help reduce the final task loss.

The search for such useful conserved quantities can take many forms. In this work, we focus on two: in section 4.1 we use a combination of program synthesis and gradient descent to generate a large, but finite, number of candidate parametric formulas for phsyical system prediction. We then try meta-tailoring with each formula on the training set and pick the conservation formula with the best loss. Formulas can be a great and interpretable description in scientific domains given the appropriate state-space. However, for general descriptions from raw data, we would like to describe the loss with a neural network as we described in the next section.

3.3 Meta-learning a neural loss function

We consider the case where conserved quantities are represented as a neural embedding. This allows us to take in raw predictions as input, such as images in the case of video prediction. For a single tailoring step the Noether loss enforcing conservation is:

$$\mathcal{L}_{\text{noether}}(x_0, \theta, T, \phi) = \sum_{t=1}^{T} \left| g_\phi(x_0) - g_\phi\left(\tilde{x}_t\right) \right|^2 = \sum_{t=1}^{T} \left| g_\phi(x_0) - g_\phi\left(f_\theta(x_0, t)\right) \right|^2 \tag{3}$$

Sometimes it is more convenient to compare the prediction at each time step with the previous one, because we often have $f_\theta(x_0, t) = f_\theta(\hat{x}_{t-1}, 1)$, with the previous time-step being the input to the network. We denote this as

$$\mathcal{L}_{\text{noether}}(x_0, \theta, T, \phi) = \sum_{t=1}^{T} \left| g_\phi(\tilde{x}_{t-1}) - g_\phi\left(\tilde{x}_t\right) \right|^2 = \sum_{t=1}^{T} \left| g_\phi(f_\theta(x_0, t-1)) - g_\phi\left(f_\theta(x_0, t)\right) \right|^2 \tag{4}$$

where $\hat{x}_0 = x_0$. These two cases are equivalent if we fully enforce the conservation law. Otherwise, the degree with which we enforce the conservation loss leads to different trade-offs. On one hand, comparing all predictions to x_0 has the advantage of always comparing to ground-truth, avoiding biases in f_θ, whereas if we compared adjacent predictions it would take $t-1$ steps for the optimization to propagate information from ground truth x_0 to $\hat{x}_t$. On the other hand, for imperfectly-conserved quantities like energy in dissipative systems, it may be better to encourage the quantity to be Lipschitz but not fully constant — here, comparing against the previous time-step may help.

In both cases, the optimization takes the same form if we tailor θ with a single gradient step:

$$\theta_{x_0}(\phi) = \theta - \nabla_\theta \left(\mathcal{L}_{\text{noether}}(x_0, \theta, T, \phi)\right); \; \hat{x}_t = f_{\theta_{x_0}(\phi)}(x_0) \tag{5}$$

We can now backpropagate the task loss $\mathcal{L}_{\text{task}}(x_{1:T}, \hat{x}_{1:T})$ back to ϕ, which will be optimized so that the unsupervised tailoring adaptation $\theta \mapsto \theta_{x_0}(\phi)$ helps to lower $\mathcal{L}_{\text{task}}$ when the sequence starts at x_0. The optimization requires second-order gradients to reason about how modifying ϕ affects $\mathcal{L}_{\text{task}}$ through $\theta_{x_0}(\phi)$. This is similar to MAML [9] and works on meta-learning loss functions for few-shot learning [3] or group distribution shift [31]. Algorithm 1 provides pseudo-code.

Algorithm 1 Training of Noether Networks with neural conservation loss

procedure METATRAINEMBEDDING($f_\theta, \mathcal{D}, N, T$)
  Sample batch $x^{(0)}_{0:T}, \ldots, x^{(N)}_{0:T} \sim \mathcal{D}$
  $\phi \leftarrow$ randomly initialized weights ▷ Random conserved embedding
  loop until convergence
    for $0 \leq n \leq N$ **do**
      $\tilde{x}^{(n)}_0 \leftarrow x^{(n)}_0$
      for $1 \leq t \leq T$ **do** ▷ unroll the time-series prediction
        $\tilde{x}^{(n)}_t \leftarrow f_\theta(\tilde{x}^{(n)}_{0:t-1})$ ▷ Use previous step prediction
      end for
      $\ell \leftarrow \sum_{t=1}^{T} \text{MSE}(g_\phi(\hat{x}^{(n)}_t), g_\phi(x^{(n)}_0))$ ▷ Cumulative inner loss for sequence
      $\theta_{x^{(n)}_0} \leftarrow \theta - \lambda_{\text{in}} \nabla_\theta \ell$
      $\hat{x}^{(n)}_0 \leftarrow x^{(n)}_0$
      for $1 \leq t \leq T$ **do**
        $\hat{x}^{(n)}_t \leftarrow f_{\theta_{x^{(n)}_0}}(\hat{x}^{(n)}_{0:t-1})$ ▷ Final prediction with tailored weights
      end for
      $\phi \leftarrow \phi - \lambda_{\text{out}} \nabla_\phi \mathcal{L}_{\text{task}}(\hat{x}^{(n)}_{1:T}, x^{(n)}_{1:T})$
    end for
  end loop
  return f_θ
end procedure

4 Experiments

Our experiments are designed to answer the following questions:

1. Can Noether Networks recover known conservation laws in scientific data?
2. Can Noether Networks parameterize useful conserved quantities from raw pixel information?
3. Are Noether Networks useful for settings with controlled dynamics?

4.1 Experimental domains

To answer the above questions, we consider a number of different experimental domains, which we overview in here, before moving on to the evaluation and results.

Ideal spring and ideal pendulum. Greydanus et al. [12] propose the setting of an ideal spring and ideal pendulum, which will allow us to understand the behavior of Noether Networks for scientific data where we know a useful conserved quantity: the energy. We build on the vanilla MLP baseline

Method	Description	RMSE
Vanilla MLP	N/A	0.0563
Noether Nets	$p^2 - 2.99\cos(q)$	0.0423
True $\mathcal{H}$ [oracle]	$p^2 - 3.00\cos(q)$	0.0422

Table 1: Recovering $\mathcal{H}$ for the ideal pendulum.

Method	Description	RMSE
Vanilla MLP	N/A	.0174
Noether Nets	$q^2 + 1.002\,p^2$	.0165
True $\mathcal{H}$ [oracle]	$q^2 + 1.000\,p^2$	.0166

Table 2: Recovering $\mathcal{H}$ for the ideal spring.

in their paper and discover conservation laws that, when used for meta-tailoring, can improve prediction performance. Since the baseline does not predict x_{t+1} but $\frac{dx}{dt}(x_t)$, we apply the loss to a finite-difference approximation, i.e. $\mathcal{L}_{\text{noether}}\left(x_t, x_t + \frac{dx}{dt}\Delta_t\right) = \mathcal{L}_{\text{noether}}\left(x_t, x_t + f_\theta(x_t)\Delta_t\right)$.

For the pendulum, $x = (p, q) \in \mathbb{R}^2$ has the angle of the pendulum q and its momentum p. The formula for the energy is equal to: $\mathcal{H} = 2mgl(1-\cos q) + \frac{l^2p^2}{2m}$. Greydanus et al. [12] set $m = \frac{1}{2}, l = 1, g = 3$, resulting in $\mathcal{H} = 3(1 - \cos q) + p^2$. Notice that a simpler conserved quantity is $p^2 - 3 \cdot \cos q$.

For the spring, $x = (p, q) \in \mathbb{R}^2$ has the displacement q as well as its momentum p, with an energy equal to: $\mathcal{H} = \frac{1}{2}kq^2 + \frac{p^2}{2m}$. Greydanus et al. [12] set $k = q = m = 1$, resulting in $\mathcal{H} = \frac{1}{2}(q^2 + p^2)$, where units are omitted. Thus, $q^2 + 1 \cdot p^2$ is a conserved quantity, with 1 having the appropriate units.

Domain specific language for scientific formulas. We program a simple domain specific language (DSL) for physical formulas. Since these formulas have physical meaning, each sub-formula carries its own associated physical units. When applying an operation, there is a check for validity (we can only input degrees to a sinusoid, we can only add quantities of the same units, etc). This allows us to significantly prune the exponential space, as done in AI-Feynman [24]. The vocabulary of the DSL is the following: <u>Input(i):</u> returns the i-th input, <u>Operation:</u> one of $\{+, -, \cdot, /, \sin, \cos, x^2\}$, <u>Parameter(u):</u> trainable scalar with units $[u]$.

Notice that, in contrast to most program synthesis approaches, our setting has continuous parameters which can be trained via gradient descent. The system first searches all valid symbolic formulas up to a certain depth and then trains their parameters on the true data to be approximately conserved.

Video prediction with real-world data. To characterize the benefits provided by Noether Networks in a real-world video prediction setting, we compare a Noether Network version of the SVG video prediction model — a variational autoencoder-based method [7] — to a standard SVG baseline on the ramp scenario of the Physics 101 dataset [29]. The dataset contains videos of various real-world objects sliding down an incline and colliding with a second object. Each model is conditioned on the first two frames of a given sequence and must predict the subsequent 20 frames. We restrict our data to the 20 degree incline (as opposed to the 10 degree incline, where several objects do not slide at all) and to sequences with a length of at least two seconds.

Pixel pendulum with controls. In settings such as robotics we may be interested in *action-conditioned* video prediction [17], i.e. predicting future video frames given the history of frames and an agent's actions. We recorded videos of episodes in OpenAI Gym's pendulum swing up environment, using a scripted policy to control the pendulum. Each model receives four history frames and a sequence of 26 policy actions starting from the current timestep, and must predict the next 26 future frames. As this is a visually simple environment, and we wish to test generalization, the training dataset has 5 episodes of 200 frames each. We hold out 195 episodes for testing.

4.2 Evaluation

Can Noether Networks recover known conservation laws in scientific data? The number of formulas increases exponentially with the number of terms involved, leading to vast search spaces. To save computation, we perform a pipe-lined approach that first generates all valid formulas up to depth 7. This removes formulas whose operations take in incompatible physical units and prunes equivalences of commutativity for the relevant operations. We obtain 41,460 and 72,372 formulas for the pendulum and spring, respectively. The difference arises due to the the different units of each

problem. Note that we do *not* enforce the conserved quantity to have units of energy, as in general not all conserved quantities are energies, so it would be too strong a bias.

The resulting formulas are purely symbolic, with some trainable parameters still to be defined. We check that they are approximately conserved by optimizing their parameters via gradient descent to have low variance within sequences of true data. We then do the same for random sequences; if conservation is two orders of magnitude smaller for the true data, we accept it as an approximate conservation. This measure is similar to others used before for conserved quantities [14], but has problems when the data is noisy, the minimisation is sub-optimal, or there are numerical issues. As a result, we obtain 210 and 219 approximately conserved quantities for the pendulum and spring, respectively.

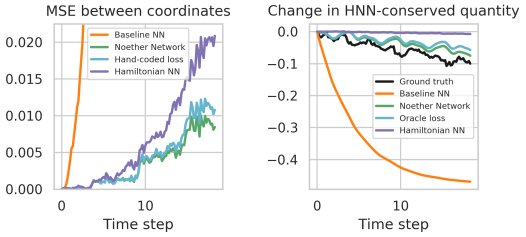

Figure 2: For a real pendulum where energy is not fully conserved, Noether Networks can still recover it. This is because they only look for quantities whose conservation helps improve predictions. Moreover, by only softly encouraging conservation, it better encodes imperfect conservations.

Finally, we evaluate candidates expression for their usefulness as Noether losses. For each candidate we try one inner step with a range of inner learning rates $[10^{-3:1:0.5}]$. To save compute, we first obtain the top 20 candidate equations by the train performance when tailoring the vanilla MLP with each loss. For those 20 candidates, we try them as meta-tailoring losses: starting from the pre-trained vanilla MLP, we fine-tune it for 100 epochs using meta-tailoring with each potential Noether loss. We then evaluate the MSE using the fine-tuned model for long-term predictions, keep the expression with the best loss in the training data, and report the long-term prediction on the test data.

We observe that, out of all conserved quantities, this process correctly singles out equations of the same form as the true Hamiltonian $\mathcal{H}$, with almost the exact parameters. It also finds equivalent losses that the initial pruning did not detect, such using a sum with a negative parameter: $p^2+(-2.99)\cdot\cos(q)$ instead of $p^2 - 2.99 \cdot \cos(q)$. Using these as losses reaches equivalent performance with that of the oracle, which fine-tunes using the true formula. Results are shown in Tables 1 and 2.

Finally, we run the same process for data coming from a real pendulum [19], which makes the Hamiltonian non-conserved. We use largely the same pipeline as for the ideal pendulum, the differences are explained in appendix B. Noether Networks end up discovering $p^2 - 2.39\cos(q)$, compared to the (potentially sub-optimal) $\mathcal{H} = p^2 - 2.4\cos q$ described in [12]. Using the discovered loss improves the baseline MLP by more than one order of magnitude, matches the performance of hand-coding the conservation loss, and improves over Hamiltonian Neural Networks, which fully impose the conservation of energy. Results can be seen in figure 2.

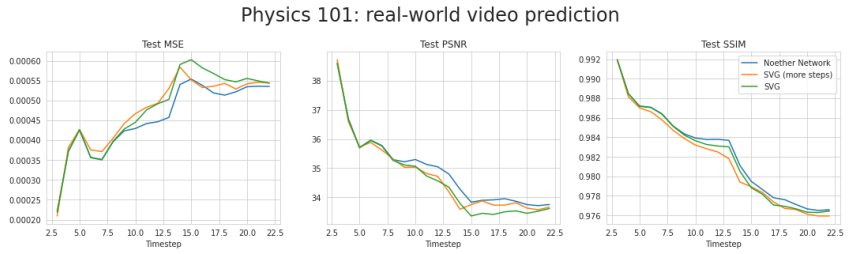

Figure 3: The Noether Network outperforms the SVG baseline by a small margin on 20-step prediction in all three metrics.

Can Noether Networks parameterize useful conserved quantities from raw pixel information? For our experiments with the Physics 101 ramp data, we train on only 311 training examples, which makes generalization to the test set difficult. In this low-data setting, the baseline SVG model struggles with overfitting — predicted frames often morph objects in the test set into objects seen at train-time, and the object in motion often morphs with the target object (see the prediction $\tilde{x}_T$ in Fig. 1, for example). For the Noether Network inner loss, we use a simple embedding parameterized

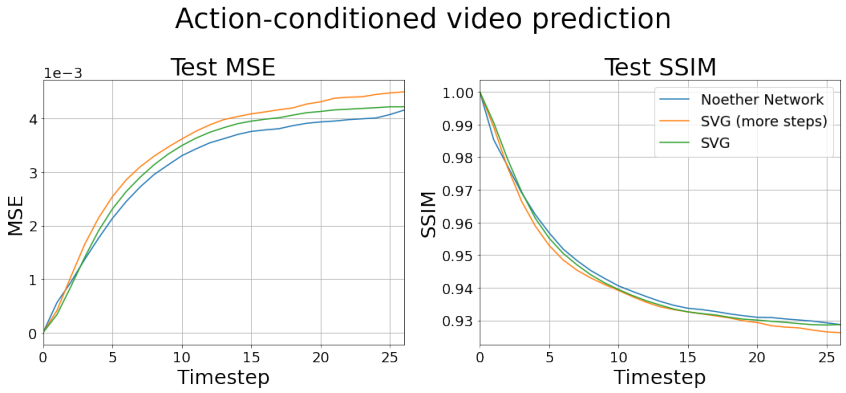

Figure 4: Test metrics in the controlled pendulum environment. The Noether Network has lower mean squared error (left) and slightly better structural similarity (right). Metrics are computed and displayed by output timestep, where timestep 0 is a given history frame. Note that simply training SVG for more steps does not increase performance.

as a shallow (two-layer) convolutional neural network followed by a fully connected layer, which projects frames onto an 8-dimensional embedding space. The inner loss for a sequence is the sum of the MSE of the embedding of each frame and the embedding of the initial frame. We train the embedding in the tailoring regime, holding the weights of the base SVG model, which trained for 400 epochs, constant. As seen in Fig. 3, taking a single tailoring step according to the learned inner loss improves performance marginally over the baseline SVG model (which has the same SVG weights as the Noether Network) on a held-out test set for MSE, PSNR, and SSIM. Additionally, the Noether Network outperforms the SVG baseline which trained for 50 more epochs. These results provide some evidence that Noether Networks are capable of learning useful inductive biases from raw video data.

Are Noether Networks useful in settings with controlled dynamics? As in the Physics 101 experiments, we begin with an SVG model but modify it to 1) remove the prior sampling component since the environment is deterministic given actions and 2) concatenate each action to the corresponding frame encoder output and feed the result to the LSTM. After training the SVG model for 50 epochs and fixing its weights, we run Alg. 1 for 20 epochs to learn an embedding network for conserving useful quantities. We found that Noether Networks perform better in this setting by directly adapting the latent LSTM activations against the conservation loss, rather than adapting any network parameters. Since Noether Networks take additional training steps starting from a pre-trained SVG model, we also continued training the SVG model for 20 further epochs, which we call "SVG (more steps)." Fig. 4 shows the results of each method on held out sequences. Noether Networks improve the overall mean squared error (MSE) and structural similarity (SSIM), more so than simply training SVG further.

5 Related Work

Unsupervised adaptation and meta-learning loss functions. Hand-designed unsupervised losses have been used to adapt to distribution shifts with rotation-prediction [23] or entropy-minimization [27], as well as to encode inductive biases [2]. Unsupervised losses have also been meta-learned for learning to encode human demonstrations [30], few-shot learning to exploit test-data as unsupervised information [3], and learning to adapt to group distribution shifts [31]. In contrast to these works, our unsupervised loss only takes the single query we care about, thus imposing no additional assumptions on top of standard prediction problems, and takes the form of a conservation law for predicted sequences.

Discovery of symmetries and conservation laws. There are several previous works which aim to learn conserved quantities in dynamical systems from data. The approach of Schmidt and Lipson [19] focuses on candidate equations that correctly predict dynamic relationships between parts of the

system by measuring the agreement between the ratios of the predicted and observed partial (time) derivatives of various components of a system. A set of analytical expressions is maintained and updated as new candidates achieve sufficient accuracy. More recently, Liu and Tegmark [14] discover conservation laws of Hamiltonian systems by performing local Monte Carlo sampling followed by linear dimensionality estimation to predict the number of conserved quantities from an explained ratio diagram. Wetzel et al. [28] learn a Siamese Network that learns to differentiate between trajectories. Contrary to both, Noether Networks do not need segmentations into trajectories with different conserved quantities and can deal with raw pixel inputs and outputs. Other approaches to learning symmetries from data include neuroevolution for learning connectivity patterns [20], regularizers for learning data augmentation [4], and meta-learning equivariance-inducing weight sharing [33]. However, these approaches don't aim to learn conserved quantities in dynamical systems.

Neural networks to discover symbolic physical laws. There has been a growing interest in leveraging the functional search power of neural networks for better scientific models. Multiple works train a neural network and then analyze it to get information, such as inferring missing objects via back-propagation descent [1], inspecting its gradients to guide a program synthesis approach [24], learning a blue-print with a GNN [6], or leveraging overparametrized-but-regularized auto-encoders [25]. Others, such as DreamCoder [8], take the explicit approach with a neural guided synthesis over the space of formulas. Unlike these works, our method can readily be applied both to symbolic conservation laws and to neural network conservation losses applied to raw videos.

Encoding physics-based inductive biases as architectural constraints. Recent works have encoded Hamiltonian and Lagrangian mechanics into neural models [12, 15, 5, 10, 32], with gains in data-efficiency in physical and robotics systems. In contrast to these works, we propose to encode biases via prediction-time fine-tuning instead of architectural constraints. This allows us to leverage the generality of loss functions to encode a class of inductive biases extending beyond Lagrangian mechanics, and handle raw video data. Outside mechanics, Suh and Tedrake [22] highlight the difficulty of learning physical inductive biases with deep models from raw data and instead propose to encode mass conservation in a video prediction model with a constrained linear model.

6 Discussion

Limitations. A disadvantage of the proposed approach over architectural constraints (in the cases when the latter can represent the same bias) is computation time. Noether Networks optimize the conservation loss inside the prediction function, which results in an overhead. This computation time can be reduced by only fine-tuning the higher-level layers, avoiding back-propagating back to the input. It is also sometimes the case that, by encoding the inductive bias during training the trained but un-tailored Noether Network performs better than a comparable architecture without the bias during training.

It is also worth noting that the theoretical guarantees apply to fully enforced conservation constraints, while in practice we only fine-tune our model to better satisfy them. However, we believe the theoretical result still conveys the potential benefits of Noether networks.

Broader impact. We proposed a new framework for meta-learning inductive biases in sequential settings, including from raw data. On the positive side, our framework can be used to discover conserved quantities in scientific applications, even when these are not fully conserved. By improving video prediction, we are also empowering predictive models for entities such as robots and self-driving cars, helping elder care and mobility. However, at the same time, it could be used for better predicting the movement and tracking of people from CCTV cameras, affecting privacy. It could also be used to improve the quality of deepfake video generation, which can be used for malicious purposes.

Overview and future work. We propose Noether Networks: a new framework for meta-learning inductive biases through conserved quantities imposed at prediction time. Originally motivated by applications in physics, our results indicate that Noether networks can recover known conservation laws in scientific data and also provide modest gains when applied to video prediction from raw pixel data. The generality of arbitrary unsupervised losses opens the door to broader applications. Finally, this work points at the usefulness of designing meta-learned inductive biases by putting priors on the biases instead of hard-coding biases directly.

References

[1] F. Alet, E. Weng, T. Lozano-Perez, and L. Kaelbling. Neural relational inference with fast modular meta-learning. In *Advances in Neural Information Processing Systems (NeurIPS) 32*, 2019.

[2] F. Alet, M. Bauza, K. Kawaguchi, N. G. Kuru, T. Lozano-Perez, and L. P. Kaelbling. Tailoring: encoding inductive biases by optimizing unsupervised objectives at prediction time. *arXiv preprint arXiv:2009.10623*, 2020.

[3] A. Antoniou and A. J. Storkey. Learning to learn by self-critique. In *Advances in Neural Information Processing Systems*, pages 9940–9950, 2019.

[4] G. Benton, M. Finzi, P. Izmailov, and A. G. Wilson. Learning invariances in neural networks. *arXiv preprint arXiv:2010.11882*, 2020.

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

[6] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho. Discovering symbolic models from deep learning with inductive biases. *arXiv preprint arXiv:2006.11287*, 2020.

[7] E. Denton and R. Fergus. Stochastic video generation with a learned prior. In *International Conference on Machine Learning*, pages 1174–1183. PMLR, 2018.

[8] K. Ellis, C. Wong, M. Nye, M. Sable-Meyer, L. Cary, L. Morales, L. Hewitt, A. Solar-Lezama, and J. B. Tenenbaum. Dreamcoder: Growing generalizable, interpretable knowledge with wake-sleep bayesian program learning. *arXiv preprint arXiv:2006.08381*, 2020.

[9] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. *arXiv preprint arXiv:1703.03400*, 2017.

[10] M. Finzi, K. A. Wang, and A. G. Wilson. Simplifying hamiltonian and lagrangian neural networks via explicit constraints. *arXiv preprint arXiv:2010.13581*, 2020.

[11] F. B. Fuchs, D. E. Worrall, V. Fischer, and M. Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *arXiv preprint arXiv:2006.10503*, 2020.

[12] S. Greydanus, M. Dzamba, and J. Yosinski. Hamiltonian neural networks. In *Advances in Neural Information Processing Systems*, pages 15353–15363, 2019.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[14] Z. Liu and M. Tegmark. Ai poincaré: Machine learning conservation laws from trajectories. *arXiv preprint arXiv:2011.04698*, 2020.

[15] M. Lutter, C. Ritter, and J. Peters. Deep lagrangian networks: Using physics as model prior for deep learning. *arXiv preprint arXiv:1907.04490*, 2019.

[16] E. Noether. Invariante variationsprobleme. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1918:235–257, 1918. URL http://eudml.org/doc/59024.

[17] J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh. Action-conditional video prediction using deep networks in atari games. *arXiv preprint arXiv:1507.08750*, 2015.

[18] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[19] M. Schmidt and H. Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

[20] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial life*, 15(2):185–212, 2009.

[21] J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, C. R. MacNair, S. French, L. A. Carfrae, Z. Bloom-Ackermann, et al. A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702, 2020.

[22] H. Suh and R. Tedrake. The surprising effectiveness of linear models for visual foresight in object pile manipulation. *arXiv preprint arXiv:2002.09093*, 2020.

[23] Y. Sun, X. Wang, Z. Liu, J. Miller, A. A. Efros, and M. Hardt. Test-time training for out-of-distribution generalization. *arXiv preprint arXiv:1909.13231*, 2019.

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

[25] S.-M. Udrescu and M. Tegmark. Symbolic pregression: Discovering physical laws from distorted video. *Physical Review E*, 103(4):043307, 2021.

[26] V. Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1995.

[27] D. Wang, E. Shelhamer, S. Liu, B. Olshausen, and T. Darrell. Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[28] S. J. Wetzel, R. G. Melko, J. Scott, M. Panju, and V. Ganesh. Discovering symmetry invariants and conserved quantities by interpreting siamese neural networks. *Physical Review Research*, 2 (3):033499, 2020.

[29] J. Wu, J. J. Lim, H. Zhang, J. B. Tenenbaum, and W. T. Freeman. Physics 101: Learning physical object properties from unlabeled videos. In *BMVC*, volume 2, page 7, 2016.

[30] T. Yu, C. Finn, A. Xie, S. Dasari, T. Zhang, P. Abbeel, and S. Levine. One-shot imitation from observing humans via domain-adaptive meta-learning. *arXiv preprint arXiv:1802.01557*, 2018.

[31] M. Zhang, H. Marklund, N. Dhawan, A. Gupta, S. Levine, and C. Finn. Adaptive risk minimization: A meta-learning approach for tackling group distribution shift. *arXiv preprint arXiv:2007.02931*, 2020.

[32] Y. D. Zhong, B. Dey, and A. Chakraborty. Symplectic ode-net: Learning hamiltonian dynamics with control. *arXiv preprint arXiv:1909.12077*, 2019.

[33] A. Zhou, T. Knowles, and C. Finn. Meta-learning symmetries by reparameterization. *arXiv preprint arXiv:2007.02933*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] In section 6
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] In section 6
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] In appendix A
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We plan, however, to open-source our code-based once cleaned.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [No] We will do it for the supplementary material deadline

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Some tables don't have them for space reasons, we will add them in the appendix.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We will add them in the appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We do so in the appendix.

(b) Did you mention the license of the assets? [Yes] We do so in the appendix.

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]