Sharpness-Aware Hybrid Model Learning for Architecture-Agnostic Parameter Estimation

Naoya Takeishi
School of Engineering
The University of Tokyo
ntake@g.ecc.u-tokyo.ac.jp

Abstract

Hybrid modeling, the combination of machine learning models and scientific mathematical models, enables flexible and robust data-driven prediction with partial interpretability. However, the unknown parameters of the scientific model cannot necessarily be estimated properly, since the flexibility of the machine learning model might make the scientific model part effectively ignored in prediction. We may avoid it by applying some regularization, but the formulation of such regularizers typically depends on model architectures and domain knowledge. In this paper, we propose an architecture-agnostic method to learn hybrid models while properly estimating the scientific parameters. The idea is to use the flatness of loss minima to achieve model simplicity, based upon the Occam's razor principle. We employ the idea of sharpness-aware minimization and adapt it to the hybrid modeling setting. Numerical experiments demonstrate the effectiveness of the SAM-based hybrid model learning for scientific parameter estimation.

1 Introduction

Combining scientific mathematical models and machine learning models can be useful for improving generalization, robustness, and partial interpretability of prediction. The combination of these two contrasting regimes has been actively studied in various fields, including weather forecasting [1, 2], robotics [3–6], and healthcare [7–9]. However, such *hybrid modeling* is, while being conceptually simple and intuitive, sometimes not technically straightforward. When a machine learning model is too flexible compared to a scientific model to be combined with, hybrid modeling may become pointless, with the machine learning part alone fitting to all the data variation and the scientific model left unused [10–14]. In other words, the unknown parameters of a scientific model, if any, may become unidentifiable regardless of their original identifiability. Such a failure mode is indeed problematic because identifying the scientific parameters correctly, in some sense, is a key factor in ensuring the partial interpretation of hybrid models.

The identifiability issue in hybrid modeling can be mitigated by constraining the flexibility of the machine learning part so that the scientific model is adequately utilized. For example, suppose an additive hybrid model $y = f(x) + g(x)$, where f and g are a scientific model and a neural net, respectively, and that both have unknown parameters. In learning such a hybrid model, minimizing a loss along with $\|g\|^2$ sounds natural to suppress the excess flexibility of g, as practiced by Yin et al. [10], because the two models are combined additively, and thus suppressing g's energy would directly result in maximizing the use of f. Now, how should we design a similar regularizer for a hybrid model with general composition, $y = g(x, f(x))$? What about $y = g_3(x, f_1(x, g_2(x)) \cdot g_1(f_2(x)))$, and whatever more complicated models? For hybrid architectures with various functional forms and nested function compositions, designing a good regularizer becomes far from trivial.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In this paper, we aim to develop a versatile method applicable to general hybrid models to partially address the unidentifiability of scientific parameters. We hypothesize that *seeking flat minima of a loss can help estimate scientific parameters*, at least to some extent. Based on the hypothesis we propose to use a variant of sharpness-aware minimization (SAM) [15] for learning deep hybrid models. The proposed method does not depend on specific model architectures and thus should be useful broadly for different hybrid models. We demonstrate the validity of the hypothesis through experiments with data from various physical systems and hybrid models with different architectures.

Our contribution lies in connecting two different lines of research that have never crossed. On the one hand, loss flatness and its algorithmic implementations have been discussed in connection with generalization capability [e.g., 16–19, 15]. Generalization is not equivalent to parameter identification, as a "correct" scientific parameter does not necessarily lead to the best generalization in hybrid models. On the other hand, regularization for hybrid models has been mostly developed in accordance with specific model architectures [10–14], and the property of a loss surface has not been considered relevant. In this paper, we shed light on the utility of loss flatness for hybrid model learning, opening up a new pathway toward architecture-agnostic parameter estimation in hybrid models.

2 Background: Hybrid Modeling

Suppose a prediction task to predict $y \in \mathcal{Y} \subset \mathbb{R}^{d_y}$ from $x \in \mathcal{X} \subset \mathbb{R}^{d_x}$. This restriction is only for notational simplicity, and the discussion below can be easily extended to more general settings. Let $f_\theta : \mathcal{X} \to \mathcal{Z}$ and $g_\phi : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ be a scientific mathematical model and a machine learning model (e.g., a deep neural net) parametrized by $\theta \in \Theta \subset \mathbb{R}^{d_\theta}$ and $\phi \in \Phi \subset \mathbb{R}^{d_\phi}$, respectively. We denote the output space of the scientific model by $\mathcal{Z}$. Importantly, we suppose that the scientific model, f_θ, is incomplete in some sense to describe data precisely, which makes it meaningful to combine the scientific model with the data-driven, machine learning model.

Unless stated otherwise, we consider the combination of the two as a general function composition:

$$h_{\theta,\phi}(x) = g_\phi(x, f_\theta(x)). \tag{1}$$

While more complex, nested compositions can be considered as well, to avoid clutter we stick to this simple form, which already may cause the issue discussed below. Note also that hybrid models in practice may contain further operations such as numerical integration over time; for example, it is typical to use hybrid neural ODEs, $y = \mathrm{ODESolve}(h_{\theta,\phi}(x; t) = 0)$, in modeling dynamical systems [10]. We omit such operations for notational simplicity.

We informally assume the following: First, g_ϕ is sufficiently flexible to approximate arbitrary functions from (compact subsets of) $\mathcal{X} \times \mathcal{Z}$ to $\mathcal{Y}$; this is usually the case when g_ϕ is a deep neural net. Second, we have data of $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ only, and the values of $z \in \mathcal{Z}$ are not observable as data; if they were, learning hybrid models would become trivial. Third, f_θ's parameters, θ, would be identifiable *if* we had sufficient data of x and z; in other words, the scientific model itself originally possessed structural and practical identifiability [see, e.g., 20–22].

Due to the flexibility of g_ϕ and the unobservability of z, we face a risk that the hybrid model, $h_{\theta,\phi}$, fits all the data variation by adjusting only the neural net's parameters ϕ, with arbitrary values of the scientific model's parameters θ. Let $L(\theta, \phi)$ denote the (population) loss function. In terms of estimating θ, merely minimizing the profile loss,

$$\min_\theta J(\theta) \quad \text{where} \quad J(\theta) \coloneqq \min_\phi L(\theta, \phi),$$

does not identify θ when the profile loss uniformly reaches a small level ε, i.e., $J(\theta) \approx \varepsilon$ for all θ, and thus the outer problem does not have clear minima (or has too many insignificant ones) in the space of θ. In the meantime the model can always fit the data due to the flexibility of g_ϕ. Consequently, the scientific model, f_θ, would be left unused, and the merits of hybrid modeling would be lost [10–12].

A natural circumvention to break the tie in the profile loss is to incorporate a constraint or regularizer. Such a regularizer should suppress the excess flexibility of the machine learning part and encourage the use of the scientific model in prediction. For instance, consider an additive hybrid model $h_{\theta,\phi}(x) = f_\theta(x) + g_\phi(x)$. Yin et al. [10] suggested suppressing the flexibility of g_ϕ by penalizing its functional norm, $\|g_\phi\|^2$. Valid formulation of such a regularizer heavily depends on task definition and model architectures; unlike the additive form that allows an intuitive design of the regularizer, it is unclear how we should regularize more complicated hybrid architectures.

3 Learning Hybrid Models with Simplicity Principle

3.1 Idea

A typical strategy in learning hybrid models is to add a regularizer, i.e., to minimize

$$J_R(\theta) \coloneqq \min_{\phi} \Big(L(\theta, \phi) + R(\theta, \phi) \Big), \tag{2}$$

where $R(\theta, \phi)$ is some regularization term for suppressing the excess flexibility of g_ϕ. For example, Yin et al. [10] suggested using $R \propto \|g_\phi\|^2$ for additive hybrid models. Takeishi and Kalousis [11] proposed to define R as the discrepancy between the full hybrid model and a reduced version where g_ϕ was somehow disabled, e.g., by replacing it with a simpler function. Standard regularizers, such as parameter norm $R \propto \|\phi\|^2$, are known to be not effective particularly for the estimation of θ [10, 11]. Good design of R heavily depends on the architecture of hybrid models, which hinders efficient development of various hybrid model architectures with complicated function compositions.

Toward an architecture-agnostic learning strategy, we step back to the Occam's razor principle that simpler models are preferred when multiple models explain the data equally well. We apply it to hybrid model learning, assuming the following intuitive property:

> good scientific parameter θ $\Leftrightarrow$ the residual task left to g_ϕ is simple and short to describe.

We show the condition for identification more formally in Appendix C. The idea can be implemented by using the free energy:

$$\bar{J}(\theta) \coloneqq \min_{q} \Big(\mathbb{E}_{\phi \sim q} L(\theta, \phi) + \mathrm{KL}(q(\phi) \parallel p(\phi)) \Big), \tag{3}$$

where p and q are prior and posterior distributions of ϕ, respectively. In (3), we consider distributions on ϕ, unlike the point estimation in (2). The KL term is necessary to prevent point-mass q. The minimizer of the $\min_q$ in (3) is known to be $q = q^*_\theta(\phi) \propto p(\phi) \exp(-L(\theta, \phi))$ [see, e.g., 23].

Why does using (3) as a loss realize simplicity-based learning? We can understand it based on the well-known argument about its relation to description length [e.g., 16, 24–26]. Suppose L is the negative log likelihood, i.e., $L(\theta, \phi) = -\log \pi(\mathcal{D} \mid \theta, \phi)$, where $\mathcal{D}$ and π denote data and a likelihood function. Substituting the minimizer q^*_θ back to (3), we have

$$\bar{J}(\theta) = -\mathbb{E}_{\phi \sim q^*_\theta} \log \pi(\mathcal{D} \mid \theta, \phi) + \mathrm{KL}(q^*_\theta(\phi) \parallel p(\phi)). \tag{4}$$

The first term of (4) is the expected cost of coding the data given ϕ (and θ), and from the "bits back" argument [16], the second term is the expected extra cost of coding ϕ itself using p as the prior code. Minimizing $\bar{J}$ therefore achieves a simple model in the sense of the minimum description length.

It has also been known that the loss in (3) or (4) is related to the flatness of loss minima [e.g., 16, 17, 19, 18, 27]. Suppose that a wrong estimation of θ can only be made good (in terms of small L) by a very specially tuned value of ϕ, namely ϕ_θ, corresponding to a non-flat, sharp minimum. The original profile loss, J, may exploit such ϕ_θ, as the argument of $\min_\phi$ can indeed point to ϕ_θ. On the other hand, $\bar{J}$ penalizes such ϕ_θ because for $\phi \neq \phi_\theta$ nearby, the value of $L(\theta, \phi)$ is not small, and thus $\mathbb{E}_{\phi \sim q} L(\theta, \phi)$ becomes large. We thus use an optimization method that takes the flatness into account, as presented in the next section.

The hypothetical benefit of $\bar{J}$ is based on the Occam's razor principle and cannot be automatically guaranteed for all problem settings, without additional assumptions about the data distribution and the model architecture. We thus examine the empirical performance of the proposed method in Section 5.

3.2 SAM for Hybrid Models

Although we can solve the inner problem of $\min_\theta \min_q (\cdots)$ exactly like (4), solving the outer problem directly is challenging because the computation and optimization of the normalizing constant of q^*_θ, which depends on θ, is usually not tractable. Thus in practice, we think of minimizing

$$\mathbb{E}_{\phi \sim q} L(\theta, \phi) + \mathrm{KL}(q(\phi) \parallel p(\phi)), \tag{5}$$

simultaneously for θ and (sufficient statistics of) $q(\phi)$. However, even if we restrict q to a tractable family, taking the expectation about high-dimensional ϕ is inefficient and unstable.

Algorithm 1 SAM for hybrid models

Require: Number of epochs M, data S, loss $\ell(\cdot,\cdot)$, learning rates η_θ, η_ϕ, perturbation radius ρ_ϕ
Ensure: Learned parameters θ, ϕ

Initialize θ, ϕ
for epoch $= 1$ to M **do**
  for each minibatch $B \subset S$ **do**
    $L_B(\theta,\phi) \leftarrow \frac{1}{|B|}\sum_{(x,y)\in B} \ell(h_{\theta,\phi}(x), y)$
    $\epsilon_\phi^* \leftarrow$ ascend direction by (7), (8), or (9) (or other variants)
    $\tilde{d}_\theta \leftarrow \nabla_\theta L_B(\theta, \phi+\epsilon_\phi^*), \quad \tilde{d}_\phi \leftarrow \nabla_\phi L_B(\theta, \phi+\epsilon_\phi^*)$
    $\theta \leftarrow \theta - \eta_\theta \tilde{d}_\theta, \quad \phi \leftarrow \phi - \eta_\phi \tilde{d}_\phi$
  end for
end for

We thus suggest relaxing the problem and using SAM [15] to seek flat (non-sharp) minima of the loss in learning hybrid models. See Appendix A for a general explanation of SAM and its variants. The overall procedures are shown in Algorithm 1. At each step we first compute the direction ϵ_ϕ^* to ascend the loss surface. We then compute the gradient at the perturbed point, go back to the original point storing the gradient, and finally do the update with the gradient.

Algorithm 1 is slightly modified from ordinary SAMs [15] in that the parameter is perturbed only in the ϕ-direction, as depicted in Figure 1. Because it is the machine learning part g_ϕ that should be simple for the maximum use of the scientific part f_θ, SAM should reach a value of ϕ around which the loss is flat along the ϕ-axis. On the other hand, nothing requires the flatness along the θ-axis; on the contrary, being flat along θ means that different values of θ achieve similarly small loss values, implying θ is hard to identify. In practice, however, the effect of the modification is not necessarily dominant in performance because a scientific model usually has a small number of parameters.

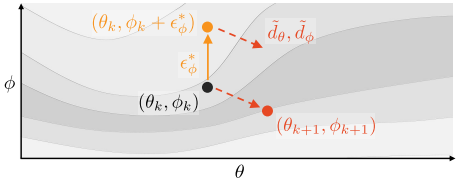


Figure 1: Update step of Algorithm 1.

Analysis Let us see how SAM is relevant to the original minimization objective in (5). Suppose the prior $p(\phi)$ and posterior $q(\phi)$ are Gaussian. Then the KL term in (5) can be computed analytically. We thus want to approximate the remaining, expected population loss $\mathbb{E}_{\phi\sim q} L(\theta,\phi)$. We can show that our SAM variant in Algorithm 1 minimizes an upper bound of it. Let $S \coloneqq \{(x_i, y_i) \mid i = 1, \dots, n\}$ be a training dataset with i.i.d. samples. The objective of SAM for hybrid models in Algorithm 1 is

$$L_S^{\text{sam}}(\theta,\phi) \coloneqq \max_{\|\epsilon\|\le\rho} L_S(\theta, \phi+\epsilon), \quad \text{where} \quad L_S(\theta,\phi) \coloneqq \frac{1}{n}\sum_{(x,y)\in S} \ell(h_{\theta,\phi}(x), y)).$$

The expected population loss is bounded by the SAM objective as follows:

Theorem 1 (informal)**.** *Suppose that ℓ is bounded and Lipschitz in θ, and that Θ is a bounded set. Let $q = q_\varphi \coloneqq \mathcal{N}(\phi \mid \varphi, \tau^2 I)$ for $\varphi \in \Phi$ and $\tau^2 > 0$. Then, with high probability over the choice of S,*

$$\mathbb{E}_{\phi\sim q_\varphi} L(\theta,\phi) \le L_S^{sam}(\theta,\varphi) + \mathcal{C}_n(\varphi,\Theta) + o(1),$$

for all $\theta \in \Theta$ and $\varphi \in \Phi$, where $\mathcal{C}_n(\varphi,\Theta)$ is a complexity term that is increasing in both $\|\varphi\|^2$ and the metric entropy of Θ, and decreases as $\widetilde{O}(1/\sqrt{n})$. The $o(1)$ term comes from a Gaussian tail and decays exponentially in d_ϕ.

Proof. See Appendix B. The proof is based on the PAC-Bayes argument of Foret et al. [15] with a uniform-convergence argument based on covering numbers of Θ. □

Moreover, Möllenhoff and Khan [27] analyzed the relation between the variational formulation of Bayesian inference, like (5), and SAM. They show the Bayes objective can be lower bounded by a soft version of SAM, where the perturbation ϵ is chosen with a soft penalty term $\propto \|\epsilon\|^2$, instead of the hard constraint $\|\epsilon\| \le \rho$. From this fact they point out (a variant of) SAM solves a relaxation of the Bayes problem, which further motivates us to use SAM to tackle the minimization of (5).

4 Related Work

4.1 Hybrid Modeling

Hybrid models (also called grey-box models, residual physics, etc.) combining scientific mathematical models (also called mechanistic models, physics models, expert models, etc.) and machine learning models have been studied for decades [e.g., 28–32]. Recently researchers have been interested in using deep neural nets in hybrid modeling [e.g. 10–12, 33–35, 14, 36–45]. The application of deep hybrid modeling has been actively practiced in various fields such as traffic prediction [46], weather forecasting [1, 2], robotics [3–6], disease prediction [47], and healthcare [7–9].

Several researchers have been working on methods to learn deep hybrid models appropriately to ensure the maximum use of scientific mathematical models. Yin et al. [10] discuss hybrid neural ODEs and propose a method to learn the model by minimizing the functional norm of the neural network part. It is a natural formulation for the additive combination of the scientific model and the neural net, but is not straightforwardly applicable to other forms of hybrid models. Takeishi and Kalousis [11] propose a regularizer that minimizes the discrepancy between the full hybrid model and a scientific-model-only reduction and apply it to variational autoencoders whose decoder has a hybrid structure. It is in principle applicable to general architectures as long as the reduced model can be defined, but the reduced model needs to be designed for each architecture. We will examine the performance of these methods in the experiments in Section 5.

Some methods utilize more specific model architectures and additional information. Zou et al. [13] present a method to use domain knowledge of the effects of interventions to maintain meaningful causal implications. Claes et al. [42] propose a principled method to properly learn hybrid models when the two parts work on distinct feature sets. Cohrs et al. [41] suggest using the method of double machine learning for specific types of hybrid models. In the context of simulation-based inference with misspecified simulators, which is very close to the idea of hybrid modeling, Wehenkel et al. [48] and Senouf et al. [49] utilize a small amount of calibration data, i.e., data of the output of the scientific model. These methods would certainly work well when the assumptions are met, but in this paper, we do not assume such specific architectures or the presence of calibration data.

4.2 Flatness and Generalization

Flatness of loss minima has been discussed in connection with generalization capability [e.g., 16, 17, 50, 19, 18, 51, 52]. Flat minima are intuitively understood as wide valleys in the loss landscape, where the loss remains low even if the parameters are perturbed. Researchers have studied the advantage of flat minima in generalization, both theoretically and empirically, often with concrete evidence of performance improvement in well-known tasks such as image classification.

One of the rationales of flat minima for better generalization is the correspondence between flatness and model simplicity. For example, Hinton and Zemel [24] and Hochreiter and Schmidhuber [17] connected flat minima with model complexity in terms of minimum description length [53]. At a flat minimum, relatively low precision is needed to describe the model's parameters maintaining the performance, thus the model is simple in the sense that the description length is short. Hochreiter and Schmidhuber [17] also argued that mere flatness is not sufficient to ensure a short description length and that a minimum should be equally flat for each parameter. It is also notable that, as discussed by Dinh et al. [54], the flatness of a loss function may be arbitrarily scaled without altering a function's outcomes. Such discussions imply the importance of careful definition of flatness [see, e.g., 55].

5 Experiment

5.1 Scope

The aim of the experiment is to examine the effectiveness of SAM-based learning of hybrid models. Importantly, we are interested in *how the method helps the estimation of scientific parameters*, especially for architectures to which the existing regularization methods [10, 11] are not straightforward to apply. Among the tasks introduced in Section 5.2, the first two tasks have already been studied in previous studies [e.g. 10], and thus the existing methods are applicable. Applying the existing methods to the other tasks is not straightforward or requires careful design of the regularizer.

Obviously, the performance of θ estimation can be measured only when some true or reference values of θ are available. We prepared the synthetic data and referred to the domain knowledge for the real-world data. Several methods of hybrid model learning are out of the scope of the current experiments or are not applicable to our tasks. The hybrid models to be tested in the experiments have unknown parameters in both of the scientific and machine learning parts; thus the learning methods used in some hybrid modeling studies [e.g., 34], where the scientific model parameters are fixed, are not of our main interest here. We do not assume at all the access to calibration data, i.e., data of the output of the scientific model, $z = f_\theta(x)$; thus the methods based on partial access to calibration data [48, 49] are out of the scope. Moreover, none of the methods assuming specific hybrid model architectures or additional information [13, 41, 42] are applicable to our tasks.

5.2 Tasks

5.2.1 Synthetic Data Tasks

PENDULUM TIME-SERIES As practiced in previous studies [e.g., 10, 11, 14], we generated data from the damped pendulum system $\ddot{v}(t) + \gamma\dot{v}(t) + (2\pi\omega)^2 \sin v(t) = 0$, where $v(t)$ is a pendulum's angle, and $\ddot{v}$ and $\dot{v}$ are the second and first derivatives with regard to t, respectively. After generating data, we assume no access to the data-generating values of γ and ω. We formulate the task as a sequence prediction from $x = \mathbf{v}(0) \in \mathbb{R}^2$ to $y = [\mathbf{v}(\Delta t), \ldots, \mathbf{v}(m_y \Delta t)] \in \mathbb{R}^{m_y \times 2}$, where $\mathbf{v} = [v, \dot{v}]$, Δt is the time step, and m_y is the output sequence length.

To formulate a hybrid model, we suppose we only know a part of the equation as an incomplete scientific model $f_\theta(v) := \ddot{v}(t) + (2\pi\tilde{\omega})^2 \sin v(t)$, where $\theta = \{\tilde{\omega}\}$ is the unknown parameter to be identified. Then a hybrid neural ODE is formulated as

$$y = \mathrm{ODESolve}(f_\theta(v) + g_\phi(v) = 0; v(0) = x), \tag{6}$$

where g_ϕ is a feed-forward neural net with two hidden layers and the ReLU activation function.

REACTION-DIFFUSION We generated data from a reaction-diffusion system: $u_t(t, \xi) = a\nabla^2 u + u - u^3 - \kappa - v$, $v_t(t, \xi) = b\nabla^2 v + u - v$, where $u(t, \xi)$ and $v(t, \xi)$ are the concentrations of two chemical species at time t and spatial location $\xi \in [-1, 1]^2$, u_t and v_t are the temporal derivatives of u and v, respectively, and ∇^2 is the Laplacian operator with regard to the space ξ. The task is again to predict a sequence of the states $y \in \mathbb{R}^{m_y \times 2 \times d \times d}$ given an initial condition $x \in \mathbb{R}^{2 \times d \times d}$, where $d = 32$ is the number of spatial discretization points along each axis. We build a hybrid neural PDE model with an incomplete scientific model as the diffusion terms: $f_\theta(u, v) := [u_t - \tilde{a}\nabla^2 u, \;\; v_t - \tilde{b}\nabla^2 v]^\top$, where $\theta = \{\tilde{a}, \tilde{b}\}$ is the unknown parameter to be identified. The machine learning part g_ϕ is a convolutional neural net. The two models are combined like in (6).

DUFFING OSCILLATOR We generated data from a nonlinear oscillator system: $\ddot{v}(t) + \gamma\dot{v}(t) + \alpha v(t) - \beta v(t)^3 = 0$. We formulate the task similarly to the preceding tasks, i.e., the prediction from an initial condition $x \in \mathbb{R}^2$ to a subsequent states $y \in \mathbb{R}^{m_y \times 2}$. The hybrid model to be used is again a hybrid neural ODE like (6) with an incomplete scientific model: $f_\theta(v) \coloneqq \ddot{v}(t) + \tilde{\alpha} v(t)$, where $\theta = \{\tilde{\alpha}\}$ is the unknown parameter to be estimated. We used a feed-forward neural net as g_ϕ.

PENDULUM IMAGES We generated trajectories of a pendulum as in the PENDULUM TIME-SERIES task and then created gray-scale images of size $d \times d$ with $d = 48$ pixels based on the simulated pendulum angles. The task is to predict future image sequences $y \in [0, 1]^{m_y \times d \times d}$ given an input sequence $x \in [0, 1]^{m_x \times d \times d}$. The hybrid model is built by composing the hybrid neural ODE in (6) and convolutional neural nets for decoding and encoding. The decoder maps the pendulum angle to an image, and the encoder maps the input images to a latent representation including the initial condition of the pendulum. Thus the overall architecture of the hybrid model is more complicated than the mere neural ODE with the additive combination. Note that the pendulum's initial condition is inherently unidentifiable; we are only interested in estimating the pendulum's frequency, i.e., $\theta = \{\tilde{\omega}\}$.

5.2.2 Real-World Data Tasks

WIND TUNNEL We used a dataset comprising measurements from a real-world wind tunnel [56]. We extracted a subset of the data to formulate a task to predict $y \in \mathbb{R}^{m_y}$ (time-seires of the pressure

inside the tunnel) from $x \in \mathbb{R}^{m_x \times 3}$ (time-series of the loads of the intake and exhaust fans and the position of the hatch). The hatch controls an additional opening in the middle of the tunnel. As a "scientific" model, we use the models "A1" and "C2" presented in Gamella et al. [56], which give a rough relation between the pressure and the fan loads. These models are incomplete in terms of the effect of the hatch position and the transient dynamics. The hybrid model transforms the output of these models with a feed-forward neural net, thus the overall architecture is more complicated than additive ones. We set a scalar parameter of the model C2 as the unknown parameter to be estimated, $\theta \in \mathbb{R}$. As the data-generating value of θ is unknown, we used a value suggested in the paper and the codes [56] as a reference to compute the estimation error of θ.

LIGHT TUNNEL We used a dataset comprising real-world measurements from a light tunnel made of light sources, linear polarizers, and a light sensor [56]. We use the data to formulate a task to predict RGB images taken by the sensor, $y \in [0, 1]^{3\times100\times100}$, from the light source configuration and the polarizer angles, $x \in \mathbb{R}^5$. As an incomplete scientific model, we use the model "F3" presented in Gamella et al. [56], which models the effect of the polarizers in a frequency-dependent way. We treat nine parameters in the model F3 as unknown parameters to be estimated, i.e., $\theta \in \mathbb{R}^9$. Our hybrid model transforms the output of the scientific model with a convolutional neural net; so its architecture depends on function compositions more complicated than mere additive combinations.

5.3 Training Methods

Empirical risk minimization (`erm`) We consider the empirical risk minimization without any particular hybrid model regularization on the machine learning part's parameters ϕ, i.e., to minimize the prediction error. We applied a slight weight decay just to stabilize the optimization.

Parameter regularization (`p-reg`) As a straightforward regularizer, we use the parameter ℓ_2 norm

$$R_{\mathrm{p}}(\phi) = \lambda_{\mathrm{p}} \|\phi\|_2^2.$$

The regularized risk is then $L_S(\theta, \phi) + R_{\mathrm{p}}(\phi)$, and $\lambda_{\mathrm{p}} > 0$ is a tunable hyperparameter.

Functional regularization (`f-reg`) We employ the idea to suppress the machine learning part's flexibility as a function by penalizing its functional norm [10] or the discrepancy between the overall model and the scientific-model-only reduction [11]. The exact form of the regularizer depends on model architectures and must be designed accordingly. For the PENDULUM TIME-SERIES, DUFFING OSCILLATOR, and REACTION-DIFFUSION tasks, where f_θ and g_ϕ are additively combined, we define the regularizer as the minimization of the function norm [10]:

$$R_{\mathrm{f}}(\phi) = \frac{\lambda_{\mathrm{f}}}{|X|} \sum_{x \in X} \|g_\phi(x)\|_2^2,$$

where X is a set of x drawn from data or some distribution. For the PENDULUM IMAGES, WIND TUNNEL, and LIGHT TUNNEL tasks, as the models are not additive, we follow the idea of Takeishi and Kalousis [11]; we define the regularizer as the minimization of

$$R_{\mathrm{f}}(\phi) = \frac{\lambda_{\mathrm{f}}}{|X|} \sum_{x \in X} \|h_{\theta,\phi}(x) - h'_{\theta,\phi'}(x)\|_2^2,$$

where $h'_{\theta,\phi'}$ is a scientific-model-only reduction of the hybrid model; for instance, for a hybrid model of the form $h_{\theta,\phi}(x) = g_\phi(x, f_\theta(x))$, we define $h'_{\theta,\phi'}(x) = Af_\theta(x)$ with a linear map A, where $\phi' = \{A\}$. Such a linear map is required when the dimensionalities of z and y are different; when they are in the same ambient space, we simply fix A as the identity map. For both definitions, the coefficient $\lambda_{\mathrm{f}} > 0$ is a hyperparameter.

SAM family (`sam`, `asam`, `fsam`) We apply SAM to the learning of hybrid models as motivated in Section 3. Instead of fixing the learning rates as in Algorithm 1, we use adaptive learning rate by Adam. We try three different ways to compute the ascend direction ϵ^*_ϕ: the standard SAM [15], adaptive SAM [57], and Fisher SAM [58]. We will refer to these three variants as `sam`, `asam`, and `fsam`, respectively. The tunable hyperparameter is the perturbation radius ρ_ϕ.

Table 1: Estimation errors of θ and prediction errors of y. For LIGHT TUNNEL, we report the cosine similarity between the estimated and reference θ's because the scale of θ is structurally unidentifiable; otherwise the root mean squared errors are reported. The best and the second best results are **bolded** and underlined, respectively. Mean and SD over 5 runs with different random seeds are reported.

		`erm`	`p-reg`	`f-reg`	`sam`	`asam`	`fsam`
PENDULUM TIME-SERIES	θ-error ($\times 10^{-3}$)	19.9 ±16.9	2.55 ±0.10	**0.37** ±0.09	4.00 ±0.54	3.35 ±1.08	3.74 ±0.46
	y-error ($\times 10^{-2}$)	9.11 ±0.13	8.00 ±0.02	43.1 ±0.42	8.21 ±0.06	8.11 ±0.07	8.20 ±0.06
REACTION -DIFFUSION	θ-error ($\times 10^{-4}$)	9.27 ±0.69	**0.72** ±0.31	1.26 ±0.66	2.79 ±1.97	3.66 ±1.67	2.80 ±1.98
	y-error ($\times 10^{0}$)	1.11 ±<0.01	1.11 ±<0.01	1.11 ±<0.01	1.19 ±0.01	1.25 ±0.02	1.19 ±0.01
DUFFING OSCILLATOR	θ-error ($\times 10^{-2}$)	30.8 ±2.25	13.4 ±2.41	18.7 ±3.67	**1.32** ±0.65	3.22 ±1.75	**1.32** ±0.91
	y-error ($\times 10^{-2}$)	6.74 ±0.05	6.45 ±0.01	6.75 ±0.04	6.43 ±0.02	6.46 ±0.02	6.43 ±0.03
PENDULUM IMAGES	θ-error ($\times 10^{-2}$)	12.2 ±5.77	12.9 ±2.79	11.7 ±5.66	5.83 ±3.86	5.48 ±3.39	**5.04** ±4.25
	y-error ($\times 10^{-0}$)	9.07 ±0.14	9.13 ±0.27	8.98 ±0.12	9.75 ±0.30	9.57 ±0.15	9.59 ±0.29
WIND TUNNEL	θ-error ($\times 10^{-2}$)	69.6 ±0.18	69.5 ±0.29	4.39 ±0.09	2.15 ±0.98	2.39 ±0.77	**1.90** ±0.67
	y-error ($\times 10^{1}$)	5.31 ±0.03	5.30 ±0.02	5.28 ±0.03	5.11 ±0.01	5.09 ±0.02	5.09 ±0.01
LIGHT TUNNEL	θ cosine sim. ↑	0.76 ±0.27	0.87 ±0.23	0.96 ±0.01	**0.98** ±0.01	**0.98** ±0.01	**0.98** ±0.01
	y-error ($\times 10^{0}$)	4.16 ±0.06	4.29 ±0.07	4.21 ±0.05	5.03 ±0.14	8.22 ±0.30	4.61 ±0.02

5.4 Results

Prediction and estimation errors (Table 1) Table 1 reports the test prediction error (y-error) and the estimation error of θ (θ-error). The magnitude of the values can be guessed in comparison to the values in the `erm` column. The SAM-based methods achieved better or comparable performance in all the tasks, although they do not assume specific hybrid model architectures. SAM family shows slightly worse y-errors in some cases, but the degradation is mostly qualitatively insignificant, and if needed we can re-train ϕ after fixing θ to recover the y-error without changing the θ-error (see Appendix F.3). More specifically:

- For PENDULUM TIME-SERIES and REACTION-DIFFUSION, the method of Yin et al. [10] was implemented as `f-reg`. It should work well because the hybrid models are in the additive form, as already confirmed in [10]. In these tasks, the performance of the SAM-based methods is slightly worse than `p-reg` and `f-reg` but is significantly improved from `erm`. Interestingly, for the REACTION-DIFFUSION task, the mere ℓ_2 regularization (`p-reg`) worked better than `f-reg`.
- For PENDULUM IMAGES, WIND TUNNEL, and LIGHT TUNNEL, where the models are non-additive, the baseline methods did not work well. It is not surprising because the baseline regularizers need special design for each architecture, which requires careful tuning to achieve good θ estimation.
- The result for DUFFING OSCILLATOR is interesting. Although the model was additive, the function norm-based `f-reg` did not estimate θ well, while the SAM-based methods were successful. This is because the function norm of g_ϕ for this task is actually minimized not at the data-generating value of $\tilde{\alpha}$ but at a larger value (see Appendix F.6).

Model behaviors (Figure 2) Figure 2 exemplifies the predictions in the LIGHT TUNNEL task, where for each method, the upper row is the prediction by the scientific model part, $\tilde{z} := f_{\tilde{\theta}}(x)$, whereas the lower row is the full prediction by the hybrid model, $\tilde{y} = h_{\tilde{\theta},\tilde{\phi}}(x)$. The light colors predicted as $\tilde{z}$ by the `erm` model sometimes deviate from the truth, implying the inadequate use of the scientific model due to the misestimation of θ. The colors in $\tilde{z}$ of the `fsam` model align well with the truth.

Hyperparameter sensitivity (Figure 3) We investigate the sensitivity of the performance to the configuration of the hyperparameters: λ_{p}, λ_{f}, and ρ_ϕ for the `p-reg`, `f-reg`, and {`sam`, `asam`, `fsam`}, respectively. We show a part of the results in Figure 3; the remaining results are in Appendix F. As a reference we also show the performance by the `erm` model with dashed lines. In terms of the θ-estimation error, the performance is not very sensitive to the hyperparameter value within the standard range of ρ [15, 57, 58]. On the other hand, the y-prediction error gets significantly worse than the reference by `erm` when the regularization effect becomes too strong.

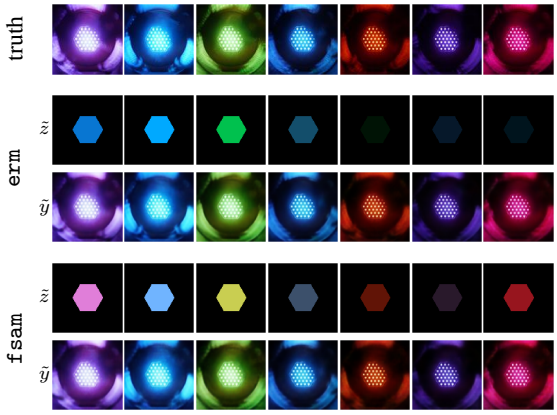
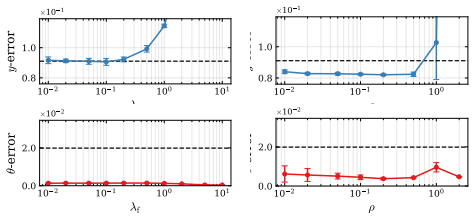
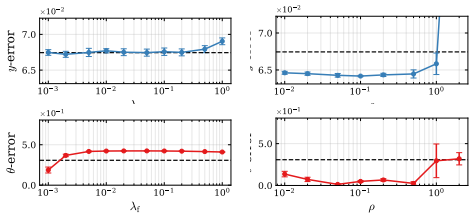


Figure 2: Data and predictions in the LIGHT TUNNEL task: (*top row*) ground truth images; (*middle two rows*) predictions from the `erm` model; and (*bottom two rows*) predictions from the `fsam` model. For each model, the upper row shows the predictions $\tilde{z}$ from the scientific model part (model F2), and the lower row shows the full predictions $\tilde{y}$.



(a) PENDULUM TIME-SERIES: (*left*) `f-reg` and (*right*) `fsam`



(b) DUFFING OSCILLATOR: (*left*) `f-reg` and (*right*) `fsam`

Figure 3: Effects of the configuration of hyperparameters: λ_f for `f-reg` and ρ for `fsam`. The dashed lines represent the average performance of `erm`. Full results are presented in the appendix.

Effect of SAM modification (Table 2) In Algorithm 1, we perturb only ϕ and leave θ unperturbed, whereas ordinary SAM methods perturb all the parameters. We examined the practical effect of such a modification. Table 2 summarizes the θ-estimation error by our application of Fisher SAM (perturbing only ϕ; `fsam`) and the ordinary Fisher SAM (perturbing ϕ and θ). For the PENDULUM TIME-SERIES and WIND TUNNEL tasks, the modified method, `fsam`, achieves clearly smaller error than the ordinary case. We found that the loss around the learned solution was particularly sharp along the θ-direction in these two tasks, which supports the role of the modification as a safeguard against flattening the loss along θ (see Appendix F.5). The effect of limiting the perturbation to ϕ appears marginal in some cases, as we anticipated earlier. For the PENDULUM IMAGES task, the ordinary Fisher SAM performs slightly better, but the difference is not significant even with more random seeds (see Appendix F.4).

Table 2: θ-error by ordinary Fisher SAM

		`fsam`	ordinary
PEND T-S	($\times 10^{-3}$)	**3.7** $\pm$0.5	37 $\pm$7.1
REACT-DIFF	($\times 10^{-4}$)	**2.8** $\pm$2.0	**2.8** $\pm$1.1
DUFFING	($\times 10^{-2}$)	**1.3** $\pm$0.9	1.4 $\pm$1.0
PEND IMAGE	($\times 10^{-2}$)	5.0 $\pm$4.3	**3.3** $\pm$2.6
WIND	($\times 10^{-2}$)	**1.9** $\pm$0.7	5.4 $\pm$1.5
LIGHT	(cos-sim $\uparrow$)	**.98** $\pm$.01	.97 $\pm$.02

6 Conclusion

In this paper, we suggest focusing on loss flatness in hybrid modeling, the combination of scientific mathematical models and machine learning models. The underlying idea is that the machine learning part of a hybrid model should be kept as simple as possible for the maximum use of the scientific model part. We have hypothesized that SAM is useful to achieve such a simple model. We have empirically validated the effectiveness of the idea on different tasks using synthetic and real-world data. An overall observation from the experiments is that SAM is indeed useful for identifying unknown parameters of the scientific model part, despite not requiring specific hybrid model architectures. The SAM-based learning method for hybrid models works in an architecture-agnostic manner, unlike the existing regularization methods that require specific design depending on the model architecture.

An important limitation is that unless one has specific assumptions about the data-generating process, the proposed learning method does not necessarily identify *the correct* value of the unknown scientific parameters, as is the case with the existing methods for hybrid modeling. It is something essential as there is often no such thing like the correct scientific parameter; the "correctness" solely depends on one's beliefs about the data-generating process and the principles of model selection. To overcome the fundamental limitation, one has to develop a specific method and theory for particular hybrid model architectures. Establishing a rigorous and universal theory of identification for general hybrid models is an interesting yet challenging open problem.

Acknowledgements

This work was supported by JST PRESTO JPMJPR24T6, JSPS KAKENHI 25K24911, and JSPS KAKENHI JP26K02968.

References

[1] Wanghan Xu, Fenghua Ling, Tao Han, Hao Chen, Wanli Ouyang, and Lei Bai. Generalizing weather forecast to fine-grained temporal scales via physics-AI hybrid modeling. In *Advances in Neural Information Processing Systems*, volume 37, pages 23325–23351, 2024.

[2] Yogesh Verma, Markus Heinonen, and Vikas Garg. ClimODE: Climate and weather forecasting with physics-informed neural ODEs. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

[3] Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauza, Leslie P. Kaelbling, Joshua B. Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3066–3073, 2018.

[4] Tim Salzmann, Elia Kaufmann, Jon Arrizabalaga, Marco Pavone, Davide Scaramuzza, and Markus Ryll. Real-time neural MPC: Deep learning model predictive control for quadrotors and agile robotic platforms. *IEEE Robotics and Automation Letters*, 8(4):2397–2404, 2023.

[5] Eric Heiden, David Millard, Erwin Coumans, Yizhou Sheng, and Gaurav S. Sukhatme. NeuralSim: Augmenting differentiable simulators with neural networks. In *Proceedings of 2021 IEEE International Conference on Robotics and Automation*, pages 9474–9481, 2021.

[6] Junpeng Gao, Mike Y. Michelis, Andrew Spielberg, and Robert K. Katzschmann. Sim-to-real of soft robots with learned residual physics. *IEEE Robotics and Automation Letters*, 9(10): 8523–8530, 2024.

[7] Andrew C. Miller, Nicholas J. Foti, and Emily Fox. Learning insulin-glucose dynamics in the wild. In *Proceedings of the 5th Machine Learning for Healthcare Conference*, pages 172–197, 2020.

[8] Victoriya Kashtanova, Ibrahim Ayed, Andony Arrieula, Mark Potse, Patrick Gallinari, and Maxime Sermesant. Deep learning for model correction in cardiac electrophysiological imaging. In *Proceedings of the 5th International Conference on Medical Imaging with Deep Learning*, pages 665–675, 2022.

[9] Emanuele Palumbo, Sorawit Saengkyongam, Maria R. Cervera, Jens Behrmann, Andrew C. Miller, Guillermo Sapiro, Christina Heinze-Deml, and Antoine Wehenkel. Hybrid modeling of photoplethysmography for non-invasive monitoring of cardiovascular parameters. arXiv:2511.14452, 2025.

[10] Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. Augmenting physical models with deep networks for complex dynamics forecasting. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[11] Naoya Takeishi and Alexandros Kalousis. Physics-integrated variational autoencoders for robust and interpretable generative modeling. In *Advances in Neural Information Processing Systems 34*, pages 14809–14821, 2021.

[12] Naoya Takeishi and Alexandros Kalousis. Deep grey-box modeling with adaptive data-driven models toward trustworthy estimation of theory-driven models. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, pages 4089–4100, 2023.

[13] Bob Junyi Zou, Matthew E. Levine, Dessi P. Zaharieva, Ramesh Johari, and Emily Fox. Hybrid^2 neural ODE causal modeling and an application to glycemic response. In *Proceedings of the 41st International Conference on Machine Learning*, pages 62934–62963, 2024.

[14] Antoine Wehenkel, Jens Behrmann, Hsiang Hsu, Guillermo Sapiro, Gilles Louppe, and Jörn-Henrik Jacobsen. Robust hybrid learning with expert augmentation. *Transactions of Machine Learning Research*, 2023.

[15] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Proceedings of the 9th International Conference on Learning Representations*, 2021.

[16] Geoffrey E. Hinton and Drew van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the 6th Annual Conference on Computational Learning Theory*, pages 5–13, 1993.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

[18] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biasing gradient descent into wide valleys. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[19] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Conference on Uncertainty in Artificial Intelligence*, 2017.

[20] Joseph H.A. Guillaume, John D. Jakeman, Stefano Marsili-Libelli, Michael Asher, Philip Brunner, Barry Croke, Mary C. Hill, Anthony J. Jakeman, Karel J. Keesman, Saman Razavi, and Johannes D. Stigter. Introductory overview of identifiability analysis: A guide to evaluating whether you have the right type of data for your modeling purpose. *Environmental Modelling & Software*, 119:418–432, 2019.

[21] F. Anstett-Collin, L. Denis-Vidal, and G. Millérioux. A priori identifiability: An overview on definitions and approaches. *Annual Reviews in Control*, 50:139–149, 2020.

[22] Franz-Georg Wieland, Adrian L. Hauber, Marcus Rosenblatt, Christian Tönsing, and Jens Timmer. On structural and practical identifiability. *Current Opinion in Systems Biology*, 25: 60–69, 2021.

[23] Paul Dupuis and Richard S. Ellis. *A Weak Convergence Approach to the Theory of Large Deviations*. Wiley, 1997.

[24] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and Helmholtz free energy. In *Advances in Neural Information Processing Systems 6*, pages 3–10, 1993.

[25] Antti Honkela and Harri Valpola. Variational learning and bits-back coding: An information-theoretic view to Bayesian learning. *IEEE Transactions on Neural Networks*, 15(4):800–810, July 2004. ISSN 1941-0093. doi: 10.1109/TNN.2004.828762.

[26] Alex Graves. Practical variational inference for neural networks. In *Advances in Neural Information Processing Systems 24*, pages 2348–2356, 2011.

[27] Thomas Möllenhoff and Mohammad Emtiyaz Khan. SAM as an optimal relaxation of Bayes. In *Proceedings of the 11th International Conference on Learning Representations*, 2022.

[28] Dimitris C. Psichogios and Lyle H. Ungar. A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, 38(10):1499–1511, 1992.

[29] R. Rico-Martínez, J. S. Anderson, and I. G. Kevrekidis. Continuous-time nonlinear signal processing: A neural network based approach for gray box identification. In *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, pages 596–605, 1994.

[30] Michael L. Thompson and Mark A. Kramer. Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, 40(8):1328–1340, 1994.

[31] U. Forssell and P. Lindskog. Combining semi-physical and neural network modeling: An example of its usefulness. *IFAC Proceedings Volumes*, 30(11):767–770, 1997.

[32] Artur M. Schweidtmann, Dongda Zhang, and Moritz Von Stosch. A review and perspective on hybrid modeling methodologies. *Digital Chemical Engineering*, 10:100136, 2024.

[33] Zhaozhi Qian, William R. Zame, Lucas M. Fleuren, Paul Elbers, and Mihaela van der Schaar. Integrating expert ODEs into neural ODEs: Pharmacology and disease progression. In *Advances in Neural Information Processing Systems 34*, pages 11364–11383, 2021.

[34] Manuel Haußmann, Sebastian Gerwinn, Andreas Look, Barbara Rakitsch, and Melih Kandemir. Learning partially known stochastic dynamics with empirical PAC Bayes. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pages 478–486, 2021.

[35] Viraj Mehta, Ian Char, Willie Neiswanger, Youngseog Chung, Andrew Oakleigh Nelson, Mark D. Boyer, Egemen Kolemen, and Jeff Schneider. Neural dynamical systems: Balancing structure and flexibility in physical prediction. In *Proceedings of the 2021 IEEE Conference on Decision and Control*, pages 3735–3742, 2021.

[36] Tushar and Souvik Chakraborty. Deep physics corrector: A physics enhanced deep learning architecture for solving stochastic differential equations. *Journal of Computational Physics*, 479:112004, 2023.

[37] Maja Rudolph, Stefan Kurz, and Barbara Rakitsch. Hybrid modeling design patterns. *Journal of Mathematics in Industry*, 14(1):3, 2024.

[38] Samuel Holt, Tennison Liu, and Mihaela van der Schaar. Automatically learning hybrid digital twins of dynamical systems. In *Advances in Neural Information Processing Systems 37*, pages 72170–72218, 2024.

[39] Yubo Ye, Maryam Tolou, Sumeet Vadhavkar, Xiajun Jiang, Huafeng Liu, and Linwei Wang. On the identifiability of hybrid deep generative models: Meta-learning as a solution. In *Advances in Neural Information Processing Systems 37*, pages 7714–7735, 2024.

[40] Stefano Giampiccolo, Federico Reali, Anna Fochesato, Giovanni Iacca, and Luca Marchetti. Robust parameter estimation and identifiability analysis with hybrid neural ordinary differential equations in computational biology. *npj Systems Biology and Applications*, 10(1):139, 2024.

[41] Kai-Hendrik Cohrs, Gherardo Varando, Nuno Carvalhais, Markus Reichstein, and Gustau Camps-Valls. Causal hybrid modeling with double machine learning—applications in carbon flux modeling. *Machine Learning: Science and Technology*, 5(3):035021, 2024.

[42] Yann Claes, Vân Anh Huynh-Thu, and Pierre Geurts. Hybrid additive modeling with partial dependence for supervised regression and dynamical systems forecasting. *Machine Learning*, 114(3):72, 2025.

[43] Romain Thoreau, Laurent Risser, Véronique Achard, Béatrice Berthelot, and Xavier Briottet. Physics-informed variational autoencoders for improved robustness to environmental factors of variation. *Machine Learning*, 114:198, 2025.

[44] Gurjeet Sangra Singh, Frantzeska Lavda, Giangiacomo Mercatali, and Alexandros Kalousis. Variational grey-box dynamics matching. In *Proceedings of the 29th International Conference on Artificial Intelligence and Statistics*, 2026.

[45] Till Richter and Niki Kilbertus. OrthoReg: Orthogonal regularization for hybrid symbolic-neural dynamical systems. arXiv:2606.19145, 2026.

[46] Ryu Shirakami, Toshiya Kitahara, Koh Takeuchi, and Hisashi Kashima. QTNet: Theory-based queue length prediction for urban traffic. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4832–4841, 2023.

[47] Sercan Ö. Arık, Chun-Liang Li, Jinsung Yoon, Rajarishi Sinha, Arkady Epshteyn, Long T. Le, Vikas Menon, Shashank Singh, Leyou Zhang, Nate Yoder, Martin Nikoltchev, Yash Sonthalia, Hootan Nakhost, Elli Kanal, and Tomas Pfister. Interpretable sequence learning for COVID-19 forecasting. In *Advances in Neural Information Processing Systems 33*, pages 18807–18818, 2020.

[48] Antoine Wehenkel, Juan L. Gamella, Ozan Sener, Jens Behrmann, Guillermo Sapiro, Marco Cuturi, and Jörn-Henrik Jacobsen. Addressing misspecification in simulation-based inference through data-driven calibration. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 65949–65980, 2025.

[49] Ortal Senouf, Antoine Wehenkel, Cédric Vincent-Cuaz, Emmanuel Abbé, and Pascal Frossard. Inductive domain transfer in misspecified simulation-based inference. In *Neural Information Processing Systems 38*, 2025.

[50] Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Alessandro Ingrosso, Carlo Lucibello, Luca Saglietti, and Riccardo Zecchina. Unreasonable effectiveness of learning neural networks: From accessible states and robust ensembles to basic algorithmic schemes. *Proceedings of the National Academy of Sciences*, 113(48):E7655–E7662, 2016.

[51] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *Proceedings of the 5th International Conference on Learning Representations*, 2017.

[52] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, 2018.

[53] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[54] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1019–1028, 2017.

[55] Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Normalized flat minima: Exploring scale invariant definition of flat minima for neural networks using PAC-bayesian analysis. In *Proceedings of the 37th International Conference on Machine Learning*, pages 9636–9647, 2020.

[56] Juan L. Gamella, Jonas Peters, and Peter Bühlmann. Causal chambers as a real-world physical testbed for AI methodology. *Nature Machine Intelligence*, 7:107–118, 2025.

[57] Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5905–5914, 2021.

[58] Minyoung Kim, Da Li, Shell X. Hu, and Timothy Hospedales. Fisher SAM: Information geometry and sharpness aware minimisation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 11148–11161, 2022.

[59] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha Dvornek, Sekhar Tatikonda, James Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Appendix Contents

A Sharpness-Aware Minimization **14**

B Loss Bound **15**

C Identifiability Condition **17**

D Task Description **18**

D.1 Pendulum Time-Series 18

D.2 Reaction-Diffusion 19

D.3 Duffing Oscillator 19

D.4 Pendulum Images 20

D.5 Wind Tunnel 20

D.6 Light Tunnel 20

E Learning and Model Selection **21**

F More Results **21**

F.1 Learning with Varying Scientific Parameters 21

F.2 Sensitivity to Hyperparameters 22

F.3 Re-training ϕ with Fixed θ 22

F.4 Results with More Random Seeds 24

F.5 Sharpness along θ and the Effect of Perturbing Only ϕ 24

F.6 Why `f-reg` Fails in the DUFFING OSCILLATOR Task 24

F.7 Computational Cost 25

F.8 Learning with High-Dimensional Scientific Parameters 25

F.9 Stress Test with a Genuinely Complex Residual 26

A Sharpness-Aware Minimization

Sharpness-aware minimization (SAM) [15] is a method that aims to learn deep neural networks with the flatness (or sharpness) of the loss taken into account. SAM and its variants [e.g., 57–59, 27] have been shown to improve the generalization capability of deep neural nets in typical tasks such as image classification.

The idea of SAM is to minimize the worst-case loss within a certain neighborhood around the current parameters. Let $L_S(w)$ be a loss function on a training dataset S for parameters w of a model. SAM tries to solve the following minimax problem:

$$\min_{w} \max_{\|\epsilon\| \leq \rho} L_S(w + \epsilon),$$

where $\rho > 0$ is a hyperparameter that defines the size of the neighborhood. The inner maximization can be approximated by a single step of gradient ascent, yielding the perturbation:

$$\epsilon^*_{\text{SAM}} = \rho \frac{\nabla L_S(w)}{\|\nabla L_S(w)\|}. \tag{7}$$

The update rule of SAM is then given as

$$w \leftarrow w - \eta \nabla L_S(w + \epsilon^*_{\text{SAM}}),$$

where $\eta > 0$ is the learning rate. The update step is performed by 1) computing the perturbation at the current parameter, 2) evaluating the loss gradient at the perturbed point, and 3) updating the parameter with the computed gradient. In practice the higher-order derivative through ϵ^* is usually ignored [15].

To compute and minimize the sharpness in a meaningful manner, the geometry of the parameter manifold should be taken into account. It has been addressed by multiple researchers based on different ideas. For example, Kwon et al. [57] proposed adaptive SAM to scale the perturbation by the magnitude of each parameter, i.e.,

$$\epsilon^*_{\text{ASAM}} = \rho \frac{w^2 \nabla L_S(w)}{\|w \nabla L_S(w)\|} \quad \text{(elementwise)}. \tag{8}$$

Kim et al. [58] proposed to use the loss curvature information for scaling. Let $F(w)$ be the expected Hessian matrix of the loss (i.e., the Fisher information matrix) at w. Then, Fisher SAM defines the perturbation as

$$\epsilon^*_{\text{FSAM}} = \rho \frac{F(w)^{-1} \nabla L_S(w)}{\sqrt{\nabla L_S(w)^\top F(w)^{-1} \nabla L_S(w)}}, \tag{9}$$

meaning that the perturbation is scaled by the local curvature of the loss landscape.

B Loss Bound

The purpose of the analysis here is to show that the empirical SAM objective,

$$L_S^{\text{sam}}(\theta, \phi) = \max_{\|\epsilon\| \le \rho} L_S(\theta, \phi + \epsilon),$$

where $L_S(\theta, \phi) = \frac{1}{n} \sum_{(x,y) \in S} \ell(h_{\theta,\phi}(x), y)$, and S is an i.i.d. training dataset of size n, gives an upper bound on the expected population loss,

$$\mathbb{E}_{\phi \sim q} L(\theta, \phi).$$

For the sake of discussion, without losing much practicality, we restrict the posterior q to be a Gaussian distribution

$$q = q_\varphi(\phi) \coloneqq \mathcal{N}(\phi \mid \varphi, \tau^2 I),$$

with some mean $\varphi \in \Phi$ and variance $\tau^2 > 0$.

In upper bounding $\mathbb{E}_{\phi \sim q_\varphi} L(\theta, \phi)$, we need to take care of the two arguments, θ and φ. We follow the PAC-Bayes argument given in the SAM paper by Foret et al. [15] for bounding the φ-part and then expand the discussion for all θ using the uniform-convergence argument based on covering numbers.

Assumption B.1. The value of ℓ is bounded in $[0, 1]$.

Assumption B.2. For every $\phi \in \Phi$ and (x, y), the map $\theta \mapsto \ell(h_{\theta,\phi}(x), y)$ is λ-Lipschitz on $\Theta \subset \mathbb{R}^{d_\theta}$ in the ℓ_2 norm.

Assumption B.3. Θ is a bounded set.

Theorem B.1. *Suppose Assumptions B.1, B.2, and B.3 hold. Let $\tau = \frac{\rho}{\sqrt{d_\varphi} + s}$ with some $s \ge 0$. Let $N(r, \Theta)$ with some $r > 0$ denote the r-covering number of Θ in the ℓ_2 norm. Then, with probability at least $1 - \delta$ over the choice of S,*

$$\mathbb{E}_{\phi \sim q_\varphi} L(\theta, \phi) \le L_S^{\text{sam}}(\theta, \varphi) + 2\lambda r + \exp\left(\frac{-(\frac{\rho}{\tau} - \sqrt{d_\varphi})^2}{2} \right) + \sqrt{\frac{\frac{1}{4} d_\varphi \log\left(1 + \frac{\|\varphi\|_2^2}{d_\varphi \tau^2}\right) + \frac{1}{4} + \log \frac{nN(r,\Theta)}{\delta} + 2\log(6n + 3d_\varphi)}{n - 1}}, \tag{10}$$

for all $\theta \in \Theta$ and $\varphi \in \Phi$.

Proof. Let $\Theta_r \subset \Theta$ be a minimal r-cover of Θ in the ℓ_2 norm; by definition, $|\Theta_r| = N(r, \Theta)$.

From the definition of q_φ, the target quantity can be written in the reparametrized form:

$$\mathbb{E}_{\phi \sim q_\varphi} L(\theta, \phi) = \mathbb{E}_{\epsilon \sim \mu_\tau} L(\theta, \varphi + \epsilon), \tag{11}$$

where $\mu_\tau := \mathcal{N}(0, \tau^2 I)$. For each fixed $\theta_k \in \Theta_r$, the discussion in Foret et al. [15] gives the following PAC-Bayes bound: with probability at least $1 - \delta'$ over the choice of S, for all $\varphi \in \Phi$,

$$\mathbb{E}_{\epsilon \sim \mu_\tau} L(\theta_k, \varphi + \epsilon) \le \mathbb{E}_{\epsilon \sim \mu_\tau} L_S(\theta_k, \varphi + \epsilon) + \sqrt{\frac{\frac{1}{4} d_\varphi \log\left(1 + \frac{\|\varphi\|_2^2}{d_\varphi \tau^2}\right) + \frac{1}{4} + \log \frac{n}{\delta'} + 2\log(6n + 3d_\varphi)}{n - 1}}. \tag{12}$$

Note that this bound holds uniformly in φ on a single $1 - \delta'$ event over S, since the PAC-Bayes inequality is uniform in the posterior mean. Applying the union bound over the $N(r, \Theta)$ points of Θ_r and setting $\delta' = \delta / N(r, \Theta)$, we obtain that with probability at least $1 - \delta$ over S, (12) holds simultaneously for all $\theta_k \in \Theta_r$ and $\varphi \in \Phi$, with $\log \frac{n}{\delta'}$ replaced by $\log \frac{nN(r,\Theta)}{\delta}$.

For any $\theta \in \Theta$, by definition of the covering, there exists $\theta_k \in \Theta_r$ with $\|\theta - \theta_k\|_2 \le r$. By assumption, $\theta \mapsto \ell(h_{\theta,\phi}(x), y)$ is λ-Lipschitz uniformly in ϕ, x, and y. Since L_S is the empirical average and L is the population expectation of ℓ, both inherit λ-Lipschitzness in θ for every ϕ':

$$|L(\theta, \phi') - L(\theta_k, \phi')| \le \lambda r, \quad |L_S(\theta, \phi') - L_S(\theta_k, \phi')| \le \lambda r.$$

Taking expectation over $\phi' \sim q_\varphi$ (equivalently, $\phi' = \varphi + \epsilon$ with $\epsilon \sim \mu_\tau$) preserves the inequalities, so

$$\mathbb{E}_{\epsilon \sim \mu_\tau} L(\theta, \varphi + \epsilon) \le \mathbb{E}_{\epsilon \sim \mu_\tau} L(\theta_k, \varphi + \epsilon) + \lambda r,$$
$$\mathbb{E}_{\epsilon \sim \mu_\tau} L_S(\theta_k, \varphi + \epsilon) \le \mathbb{E}_{\epsilon \sim \mu_\tau} L_S(\theta, \varphi + \epsilon) + \lambda r.$$

Chaining these with (12) after the union bound, with probability at least $1 - \delta$, for all $\theta \in \Theta$ and $\varphi \in \Phi$,

$$\mathbb{E}_{\epsilon \sim \mu_\tau} L(\theta, \varphi + \epsilon) \le \mathbb{E}_{\epsilon \sim \mu_\tau} L_S(\theta, \varphi + \epsilon) + 2\lambda r + \sqrt{\frac{\frac{1}{4} d_\varphi \log\left(1 + \frac{\|\varphi\|_2^2}{d_\varphi \tau^2}\right) + \frac{1}{4} + \log \frac{nN(r,\Theta)}{\delta} + 2\log(6n + 3d_\varphi)}{n - 1}}. \tag{13}$$

Splitting the expectation to the inside and outside of a ball of radius ρ, the first term of the rhs of (13) is

$$\mathbb{E}_{\epsilon \sim \mu_\tau} L_S(\theta, \varphi + \epsilon) = \int_{\|\epsilon\| \le \rho} L_S(\theta, \varphi + \epsilon) d\mu_\tau(\epsilon) + \int_{\|\epsilon\| > \rho} L_S(\theta, \varphi + \epsilon) d\mu_\tau(\epsilon).$$

Since the loss is bounded in $[0, 1]$,

$$\mathbb{E}_{\epsilon \sim \mu_\tau} L_S(\theta, \varphi + \epsilon) \le L_S^{\text{sam}}(\theta, \varphi) \mu_\tau(B_\rho) + \mu_\tau(B_\rho^{\text{c}}) \le L_S^{\text{sam}}(\theta, \varphi) + \mu_\tau(B_\rho^{\text{c}}), \tag{14}$$

where B_ρ is the ball of radius ρ around 0, and B_ρ^{c} denotes the complement of B_ρ.

For $\epsilon \sim \mu_\tau$, write $\epsilon = \tau z$ with $z \sim \mathcal{N}(0, I)$. The map $z \mapsto \|z\|_2$ is 1-Lipschitz, and Jensen's inequality tells $\mathbb{E}\|z\| \le \sqrt{\mathbb{E}\|z\|^2} = \sqrt{d_\varphi}$. The Borell–TIS inequality gives, for every $t \ge 0$,

$$\Pr\left(\|z\| \ge \mathbb{E}\|z\| + t\right) \le e^{-\frac{t^2}{2}},$$

and consequently,

$$\Pr\left(\|\epsilon\| \ge \tau(\sqrt{d_\varphi} + t)\right) \le e^{-\frac{t^2}{2}}.$$

Setting $t = \frac{\rho}{\tau} - \sqrt{d_\varphi} = s \ge 0$,

$$\mu_\tau(B_\rho^{\text{c}}) = \Pr\left(\|\epsilon\| > \rho\right) \le e^{-\frac{(\frac{\rho}{\tau} - \sqrt{d_\varphi})^2}{2}}. \tag{15}$$

Substituting (14) and (15) into (13), and using the reparametrization (11), with probability at least $1-\delta$, for all $\theta \in \Theta$ and $\varphi \in \Phi$,

$$\mathbb{E}_{\phi\sim q_\varphi} L(\theta,\phi) \le L_S^{\mathrm{sam}}(\theta,\varphi) + 2\lambda r + \exp\left(\frac{-(\frac{\rho}{\tau}-\sqrt{d_\varphi})^2}{2}\right) + \sqrt{\frac{\frac{1}{4}d_\varphi \log\left(1+\frac{\|\varphi\|_2^2}{d_\varphi \tau^2}\right) + \frac{1}{4} + \log\frac{nN(r,\Theta)}{\delta} + 2\log(6n+3d_\varphi)}{n-1}}.$$

□

C Identifiability Condition

We try to formulate a condition for the identification of θ more formally than in the main text and show that such a condition does not depend on specific model architectures. The target of the analysis here is the free energy in Equation (3):

$$\bar{J}(\theta) \coloneqq \min_q \Big(\mathbb{E}_{\phi\sim q} L(\theta,\phi) + \mathrm{KL}(q(\phi) \parallel p(\phi))\Big).$$

We aim to state the conditions under which this $\bar{J}$ is helpful for identifying the scientific parameter, θ.

Let $\theta_0 \in \Theta$ be the "correct" value of the scientific parameter. What a "correct" scientific parameter means depends on context, and we do not limit the scope of discussion here in this regard. Let $\phi_0 \in \Phi$ be the corresponding value of the machine learning parameter, that is, $\phi_0 \in \arg\min_\phi L(\theta_0,\phi)$. We assume that θ_0 is identifiable if we fix the machine learning parameter at ϕ_0. More formally:

Assumption C.1. Let $\delta_{\phi_0}(\theta) \coloneqq L(\theta,\phi_0) - L(\theta_0,\phi_0)$. Under the fixed ϕ_0, θ_0 is identifiable in the sense that $\delta_{\phi_0}(\theta) > 0$ for all $\theta \neq \theta_0$.

The core of our motivation in the hybrid modeling study is the observation that, regardless of the original identifiability assumption under the fixed, oracle ϕ_0, the profile loss:

$$J(\theta) = \min_\phi L(\theta,\phi)$$

loses the identifiability, at least practically, because when the machine learning part of a hybrid model is highly flexible, the $\min_\phi$ operation yields $J(\theta) \approx J(\theta_0)$ for many (or even all) $\theta \neq \theta_0$.

The previous studies on hybrid model learning [10–12] used and analyzed some regularizers to break the ties in the profile loss. They are certainly helpful when the model architecture follows the design idea of the regularizers. However, the need to design regularizers specifically for each architecture hinders the broad application of hybrid modeling with various architecture choices.

Now let us see under what conditions $\bar{J}$ may help the identification. It is well known that the minimizer of the $\min_q$ operation in the definition of $\bar{J}$ is

$$q_\theta^*(\phi) \coloneqq \frac{p(\phi)\exp(-L(\theta,\phi))}{Z(\theta)},$$

where the partition function is $Z(\theta) \coloneqq \int p(\phi)\exp(-L(\theta,\phi))d\phi$. A natural consequence is

$$\bar{J}(\theta) = -\log Z(\theta),$$

and thus

$$\bar{J}(\theta) = J(\theta) - \log V(\theta) \quad \text{where} \quad V(\theta) = \int \exp\Big(-(L(\theta,\phi)-J(\theta))\Big)p(\phi)d\phi. \tag{16}$$

This view allows us to think that $\bar{J}(\theta)$ has the $-\log V(\theta)$ term as a regularizer. From the definition of $V(\theta)$, it measures the mass of prior $p(\phi)$ corresponding to the region where $L(\theta,\phi) - J(\theta) \ge 0$ is small, that is, for a fixed θ, the region of good ϕ in terms of the small excess loss.

Let

$$G_{\phi_0}(\theta) \coloneqq L(\theta,\phi_0) - J(\theta) \ge 0$$

be the loss improvement, under fixed θ, given when ϕ is moved from ϕ_0 to the profile minimizer, $\arg\min_\phi L(\theta,\phi)$. This value reflects the extent of the compensation by the machine learning part for a wrong θ. We can formulate an identification condition as follows.

Proposition C.1. *Let Assumption C.1 hold. If*

$$-\log V(\theta) > -\log V(\theta_0) + G_{\phi_0}(\theta) - \delta_{\phi_0}(\theta) \quad \textit{for} \quad \theta \neq \theta_0, \tag{17}$$

then $\bar{J}$ can identify θ_0 in the sense that $\bar{J}(\theta) - \bar{J}(\theta_0) > 0$ for $\theta \neq \theta_0$.

Proof. Since $J(\theta) = L(\theta, \phi_0) - G_{\phi_0}(\theta)$ by the definition of G_{ϕ_0}, we have

$$J(\theta) - J(\theta_0) = L(\theta, \phi_0) - L(\theta_0, \phi_0) - G_{\phi_0}(\theta) + G_{\phi_0}(\theta_0) = \delta_{\phi_0}(\theta) - G_{\phi_0}(\theta),$$

because $G_{\phi_0}(\theta_0) = 0$ as we assumed $\phi_0 \in \arg\min_\phi L(\theta_0, \phi)$. Consequently, from (16),

$$\bar{J}(\theta) - \bar{J}(\theta_0) = \delta_{\phi_0}(\theta) - G_{\phi_0}(\theta) + \log \frac{V(\theta_0)}{V(\theta)}.$$

Therefore, when (17) holds, we have $\bar{J}(\theta) - \bar{J}(\theta_0) > 0$ for $\theta \neq \theta_0$. □

An important note is that we here never say that the use of $\bar{J}$ admits a free-hand identification of θ. On the contrary, the identification is ensured only when the condition (17) is met, and we have not discussed the exact situations when it is met. Rather, the point of the discussion above is that the identification condition (17) does not presume specific model architectures. Moreover, the global identification discussed above is probably overly strong, and arguments for weaker local identification would be useful for analyzing practical deep hybrid modeling.

D Task Description

D.1 Pendulum Time-Series

Data We simulated the damped pendulum system:

$$\ddot{v}(t) + \gamma \dot{v}(t) + (2\pi\omega)^2 \sin v(t) = 0,$$

where $v(t) : \mathbb{R} \to \mathbb{R}$ is the pendulum's angle, and $\ddot{v}$ and $\dot{v}$ denote the second and first derivatives with regard to time $t \in \mathbb{R}$, respectively. For the simulation we used the explicit Runge-Kutta method of order 5(4) for the numerical integration with a time step of 0.02. We then subsampled the sequence by a factor of 10, so the effective time step of the data is $\Delta t = 0.2$. We set $\gamma = 0.5$ and $\omega = 2/3$ for data generation.

We formulate the task as the prediction from an initial state x to a subsequent sequence of states y:

$$\begin{aligned} x &= \mathbf{v}(0) \in \mathbb{R}^2 \quad \text{and} \\ y &= [\mathbf{v}(\Delta t), \ldots, \mathbf{v}(m_y \Delta t)] \in \mathbb{R}^{m_y \times 2}, \end{aligned}$$

where $\mathbf{v} = [v, \dot{v}]$, and $m_y = 10$ is the output sequence length. The initial angle $v(0)$ and angular velocity $\dot{v}(0)$ were randomly sampled from the uniform distribution $U(-\pi/2, \pi/2)$. We generated 25, 25, and 25 input-output pairs for the training, validation, and test data, respectively. We added zero-mean Gaussian noise with standard deviation 0.01 to the data. Similar data were also used in Yin et al. [10].

Model The hybrid model we used is a hybrid neural ODE as practiced in Yin et al. [10]. We suppose that we know a part of the data-generating equation as an incomplete scientific model:

$$f_\theta(v) \coloneqq \ddot{v}(t) + (2\pi\tilde{\omega})^2 \sin v(t),$$

where $\theta = \{\tilde{\omega}\}$ is the unknown parameter to be identified. Then the hybrid neural ODE is formulated as

$$y = \mathrm{ODESolve}(f_\theta(v) + g_\phi(v) = 0; v(0) = x)$$

where g_ϕ is a feed-forward neural net with two hidden layers of size 128 and the ReLU activation function.

D.2 Reaction-Diffusion

Data We simulated the FitzHugh-Nagumo type reaction-diffusion system:

$$u_t(t,\xi) = a\nabla^2 u + u - u^3 - \kappa - v,$$
$$v_t(t,\xi) = b\nabla^2 v + u - v,$$

where $u(t,\xi), v(t,\xi) : \mathbb{R} \times [-1,1]^2 \to \mathbb{R}$ are the concentrations of two chemical species at time $t \in \mathbb{R}$ and spatial location $\xi \in [-1,1]^2$. u_t and v_t denote the temporal derivatives of u and v, respectively. ∇^2 is the Laplacian operator with regard to the space. For the simulation we used the explicit Runge-Kutta method of order 5(4) for the numerical integration with a time step of 0.001. We then subsampled the sequence by a factor of 100, so the effective time step of the data is $\Delta t = 0.1$. We set $a = 0.001$, $b = 0.005$, and $\kappa = 0.005$ for data generation.

The task is to predict from an initial state $x \in \mathbb{R}^{2\times d\times d}$ to the subsequent sequence of states $y \in \mathbb{R}^{m_y\times 2\times d\times d}$, where $d = 32$ is the number of spatial discretization points along each axis, and $m_y = 5$ is the output sequence length. We used the five-point stencil method for approximating the Laplacian operator. The initial concentrations $u(0,\xi)$ and $v(0,\xi)$ were generated by sampling from the uniform distribution $U(0,1)$. We generated 100, 100, and 100 input-output pairs for the training, validation, and test data, respectively. We added zero-mean Gaussian noise with standard deviation 0.01 to the data. Similar data were also used in Yin et al. [10].

Model We set an incomplete scientific model as the diffusion terms:

$$f_\theta(u,v) \coloneqq \begin{bmatrix} u_t - \tilde{a}\nabla^2 u \\ v_t - \tilde{b}\nabla^2 v \end{bmatrix},$$

where $\theta = \{\tilde{a}, \tilde{b}\}$ is the unknown parameters to be identified. It is combined with a machine learning part g_ϕ implemented as a convolutional neural net (kernel size 3) with two hidden layers of size (i.e., number of channels) 16, the ReLU activation function, and batch normalization. The combination is formulated as

$$y = \mathrm{ODESolve}(f_\theta(u,v) + g_\phi(u,v) = 0; v(0) = x).$$

D.3 Duffing Oscillator

Data We simulated a nonlinear oscillator system:

$$\ddot{v}(t) + \gamma\dot{v}(t) + \alpha v(t) - \beta v(t)^3 = 0.$$

For the simulation we used the explicit Runge-Kutta method of order 5(4) for the numerical integration with a time step of 0.005. We then subsampled the sequence by a factor of 20, so the effective time step of the data is $\Delta t = 0.1$. We set $\alpha = 1$, $\beta = -1$, and $\gamma = 0.05$ for data generation.

We formulate the task as the prediction from an initial condition x to a subsequent states y of length m_y:

$$x = \mathbf{v}(0) \in \mathbb{R}^2 \quad \text{and}$$
$$y = [\mathbf{v}(\Delta t), \ldots, \mathbf{v}(m_y \Delta t)] \in \mathbb{R}^{m_y\times 2},$$

where $\mathbf{v} = [v, \dot{v}]$, and $m_y = 10$ is the output sequence length. We generated 100, 100, and 100 input-output pairs for the training, validation, and test data, respectively. We added zero-mean Gaussian noise with standard deviation 0.01 to the data.

Model We set an incomplete scientific model:

$$f_\theta(v) \coloneqq \ddot{v}(t) + \tilde{\alpha} v(t),$$

where $\theta = \{\tilde{\alpha}\}$ is the unknown parameter to be identified. The hybrid model is a hybrid neural ODE:

$$y = \mathrm{ODESolve}(f_\theta(v) + g_\phi(v) = 0; v(0) = x)$$

where g_ϕ is a feed-forward neural net with two hidden layers of size 128 and the ReLU activation function.

D.4 Pendulum Images

Data We simulated trajectories of a pendulum as in the PENDULUM TIME-SERIES task and then rendered gray-scale images of size $d \times d$ with $d = 48$ pixels based on the simulated pendulum angles. The task is to predict future image sequences, $y \in [0, 1]^{m_y \times d \times d}$, given a preceding sequence of images, $x \in [0, 1]^{m_x \times d \times d}$, where $m_x = 10$ and $m_y = 10$ are the input and output sequence length, respectively. We generated 200, 200, and 200 input-output pairs for the training, validation, and test data, respectively.

Model The hybrid model is built by combining the hybrid neural ODE used in the PENDULUM TIME-SERIES task and additional neural nets for encoding and decoding. The model has two encoders. The first encoder receives the input image sequence and gives the initial condition of the pendulum for the hybrid neural ODE. The second encoder receives the input image sequence and gives a latent representation, which is then concatenated with the output of the hybrid neural ODE. The decoder maps the output of the hybrid neural ODE (pendulum angles), concatenated with the latent representation, to images. It comprises 1) a network with fully-connected layers with two hidden layers of size 1152 with ReLU and 2) a convolutional network (kernel size 3) with two hidden layers of size (i.e., number of channels) 576 and 288 with ReLU and batch normalization.

Note that the absolute initial condition of the pendulum is in principle unidentifiable because we do not give any correspondence between the images and the pendulum angles. However it is not problematic here because we focus on the estimation of the pendulum's frequency parameter, $\theta = \{\tilde{\omega}\}$, and the interest is not in the phase of the oscillation. The encoder should learn the map from images to the initial condition only up to the translational invariance.

D.5 Wind Tunnel

Data We used a part of the dataset by Gamella et al. [56][1]. It is a dataset comprising measurements from a real-world wind tunnel. We extracted a subset of the data ("loads_hatch_mix_fast_run" of "wt_walks_v1" dataset) to formulate a task to predict a sequence of the pressure inside the tunnel $y \in \mathbb{R}^m$ from a sequence of the loads of the intake and exhaust fans and the position of the hatch, $x \in \mathbb{R}^{m \times 3}$, where m is the length of the input and output sequences. The hatch controls an additional opening at the middle of the tunnel. We created the subsequences $\{(x, y)\}$ by a sliding window of length $m = 200$ with a stride of 50 from the original time series. Finally we have 600, 150, and 235 input-output pairs for the training, validation, and test data, respectively.

Model As an incomplete scientific model f_θ, we use the models "A1" and "C2" presented in Gamella et al. [56], which give the relation between the pressure and the fan loads. They are incomplete in terms of the effect of the hatch position and the transient dynamics of the pressure. The hybrid model transforms the output of these models with a feed-forward neural net. We set a scalar parameter of the model C2, namely the ratio $r \in [0, 1]$ of the maximum airflow that the fans produce at full speed, as the unknown parameter to be estimated, i.e., $\theta = \{r\}$. As the truly data-generating value of the parameter is never known, we used a value suggested in the paper and the codes [56], $r = 0.7$, as a reference value to compute the estimation error of θ. The full hybrid model is

$$y = g_\phi(z, x),$$

where $z = f_\theta(x) \in \mathbb{R}^m$ is the output of the scientific model, and g_ϕ is a feed-forward neural net with two hidden layers of size $2m$ and the ReLU activation function.

D.6 Light Tunnel

Data We used another part of the dataset by Gamella et al. [56][1]. It contains real-world measurements from a light tunnel made of light sources, linear polarizers, and a light sensor. We extracted a subset of the data ("uniform_ap_1.8_iso_500.0_ss_0.005" of "lt_camera_v1" dataset) to formulate a task to predict the RGB image taken by the sensor, $y \in [0, 1]^{3 \times 100 \times 100}$, from the light source configuration and the polarizer angles, $x = [R \quad G \quad B \quad \alpha_1 \quad \alpha_2] \in \mathbb{R}^5$. R, G, B are the intensities of the red, green, and blue light sources, respectively, and α_1, α_2 are the angles of the two polarizers. Finally we have 6000, 1000, and 2500 input-output pairs for the training, validation, and test data, respectively.

[1] `https://github.com/juangamella/causal-chamber`

Model As an incomplete scientific model, f_θ, we use the model "F3" presented in Gamella et al. [56], which models the effect of the polarizers in a frequency-dependent way:

$$\begin{bmatrix}\tilde{R}\\ \tilde{G}\\ \tilde{B}\end{bmatrix} = \min\left(1, e\,\mathrm{diag}(\boldsymbol{w})\boldsymbol{S}\,\mathrm{diag}\left((\boldsymbol{t}^{\mathrm{p}} - \boldsymbol{t}^{\mathrm{c}})\cos^2(\alpha_1 - \alpha_2) + \boldsymbol{t}^{\mathrm{c}}\right)\begin{bmatrix}R\\ G\\ B\end{bmatrix}.\right) \tag{18}$$

We used the value of e and $\boldsymbol{S}$ suggested in the paper and the codes [56]. The remaining, $\boldsymbol{w} \in \mathbb{R}^3$, $\boldsymbol{t}^{\mathrm{p}} \in \mathbb{R}^3$, and $\boldsymbol{t}^{\mathrm{c}} \in \mathbb{R}^3$, are treated as unknown parameters to be estimated, i.e., $\theta \in \mathbb{R}^9$. As evident from Equation (18), the magnitude of these parameters is inherently unidentifiable. We thus compared the estimation results and the reference values suggested in the paper and the codes [56] with the cosine similarity. Given $\tilde{R}$, $\tilde{G}$, and $\tilde{B}$ computed by Equation (18), then f_θ outputs an image by replicating the RGB values spatially in a hexagonal shape resembling the light placement pattern. The full hybrid model further transforms such an image with a convolutional neural net with U-Net architecture.

E Learning and Model Selection

For all tasks and learning strategies, we used the Adam optimizer for setting adaptive learning rates. The overall learning rate of Adam was set to 10^{-4} for PENDULUM TIME-SERIES, REACTION-DIFFUSION, and DUFFING OSCILLATOR; 3×10^{-4} for WIND TUNNEL; and 10^{-3} for PENDULUM IMAGES and LIGHT TUNNEL. We run the optimization for 20000 iterations and evaluate the final iterate. We did not use early stopping, and the number of iterations was fixed in advance for all the methods based on pilot runs in which we visually confirmed convergence. We did full-batch training for PENDULUM TIME-SERIES, REACTION-DIFFUSION, DUFFING OSCILLATOR, and WIND TUNNEL; and mini-batch training with a batch size of 50 for PENDULUM IMAGES and LIGHT TUNNEL. We applied the cosine annealing schedule for the learning rate with a single period over the entire training iterations, i.e., the learning rate is gradually decreased from the initial value to 10^{-6}.

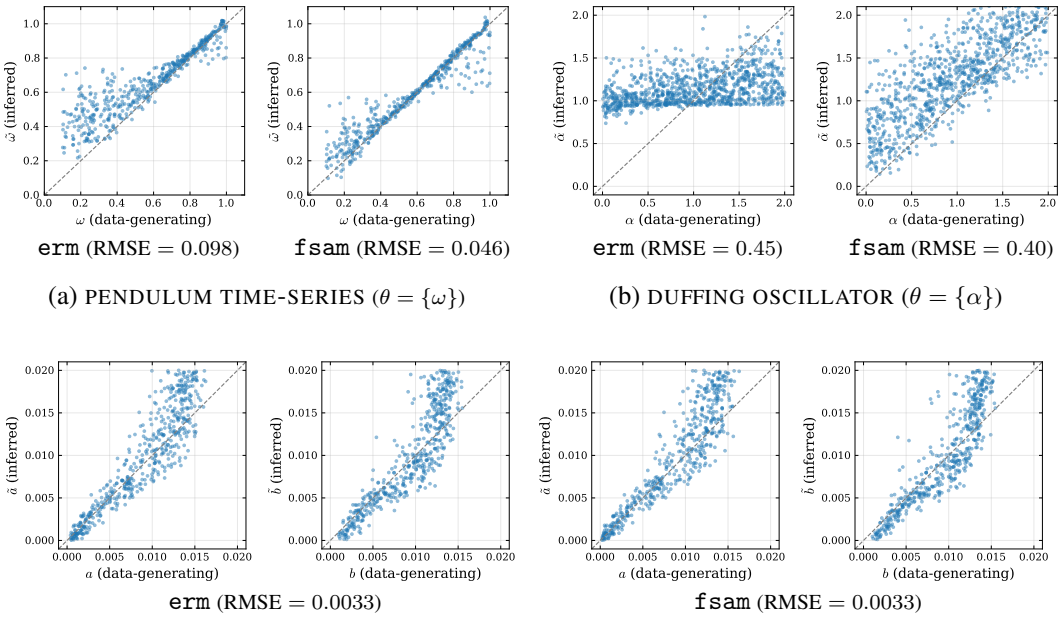
For each method, we report the performance with the hyperparameter (λ_{p}, λ_{f}, or ρ_ϕ) that achieved the best θ-error averaged over the random seeds. Although it is the normal practice to select models by validation prediction errors, we did not do so because the y-error and θ-error are usually not consistent with finite training and validation datasets; it is, in the first place, the cause of our headache in hybrid modeling! If the validation/test prediction error was a good proxy of the parameter estimation error, we (and other researchers) would not have done such a lot of research on this topic. To maintain the comparison fair, we applied this selection strategy not only to the proposed methods but also to all the baseline methods. We stress, however, that this selection uses the ground-truth (or reference) values of θ, which are not available in practice; the values in Table 1 should thus be read as a controlled comparison of the methods rather than as the performance attainable without any knowledge of θ.

F More Results

F.1 Learning with Varying Scientific Parameters

We addressed a more challenging problem setting, where the data are generated by varying values of θ. For the PENDULUM TIME-SERIES task, we randomly sampled ω from the uniform distribution $U(0.1, 1.0)$ for each input-output pair. The input x is now a sequence $x = [\mathbf{v}(-(m_x - 1)\Delta t), \ldots, \mathbf{v}(0)] \in \mathbb{R}^{m_x \times 2}$, where $m_x = 10$ is the input sequence length, instead of the mere initial condition $\mathbf{v}(0)$ in the original task. We added an encoder network to the model, which receives the input sequence and outputs an estimate of ω. The encoder network is a feed-forward neural net with two hidden layers of size 256 and the ReLU activation function. x is first flattened to a vector before feeding it to the encoder.

We altered the REACTION-DIFFUSION task similarly to the above. We randomly sampled a and b from the uniform distributions $U(0, 0.02)$ and $U(0, 0.02)$, respectively, for each input-output pair. The input x is now a sequence of 2-channel 2-D fields of size $m_x \times 2 \times d \times d$ with length $m_x = 5$. The encoder network comprises a convolutional network and a feed-forward network with fully-connected layers. The convolutional net has two hidden layers of size 288 and 144, the ReLU activation function, and batch normalization. We feed the input x to the convolutional net by first flattening the temporal



`erm` (RMSE $= 0.098$) `fsam` (RMSE $= 0.046$)

(a) PENDULUM TIME-SERIES ($\theta = \{\omega\}$)

`erm` (RMSE $= 0.45$) `fsam` (RMSE $= 0.40$)

(b) DUFFING OSCILLATOR ($\theta = \{\alpha\}$)

`erm` (RMSE $= 0.0033$) `fsam` (RMSE $= 0.0033$)

(c) REACTION-DIFFUSION ($\theta = \{a, b\}$)

Figure 4: Learning with varying scientific parameters: Inferred values of θ.

dimension into the channel dimension. Its output is then fed to the feed-forward net, which has one hidden layer of size 256 and the ReLU activation function.

For the DUFFING OSCILLATOR task, we randomly sampled α from the uniform distribution $U(0, 2)$ for each input-output pair. The input x is now a sequence of length $m_x = 10$. The encoder network is a feed-forward neural net with two hidden layers of size 256 and the ReLU activation function.

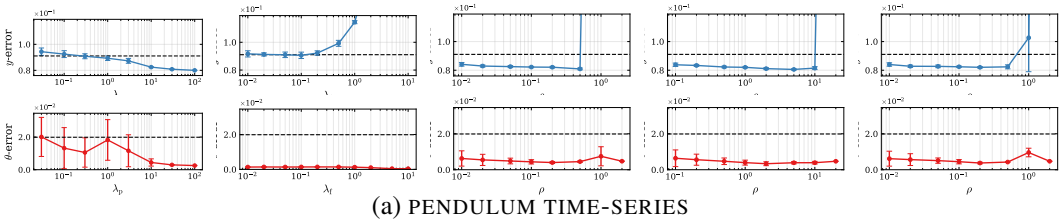
In Figure 4, we show the outputs of the encoder networks. Most notably, in the DUFFING OSCILLATOR task, the encoder learned with `fsam` infers θ with better correlation to the data-generating values, compared to the `erm` case. Meanwhile there remain non-negligible variances in the inferred values, which suggests the difficulty of the problem. On the other hand, the results of the REACTION-DIFFUSION task show almost no difference between the two methods, which implies that the task is relatively easy.

F.2 Sensitivity to Hyperparameters

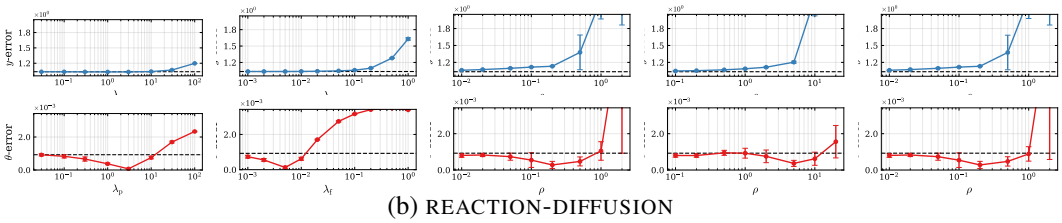
In Figure 5, we report the full result of the sensitivity of parameters with regard to the hyperparameter configurations.

F.3 Re-training ϕ with Fixed θ

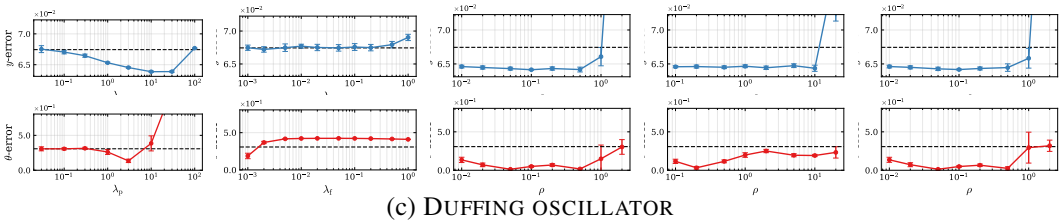
In Table 1, the SAM-based methods show slightly worse y-errors than `erm` in the REACTION-DIFFUSION, PENDULUM IMAGES, and LIGHT TUNNEL tasks. We examined whether the y-error can be recovered by re-training ϕ after fixing θ. For the `fsam` models in Table 1, we fixed θ at the estimated values, re-initialized ϕ, and trained ϕ again with the same configuration as the original training (optimizer, learning rate, batch size, prior, and the number of iterations). Table 3 shows the results. The re-training recovered y-errors comparable to those of `erm` in all the three tasks, while the θ-errors remain exactly at the values in Table 1 because θ is fixed. The extra optimization cost is a limitation, but the original degradation of the y-error was mostly too small to notice by visually inspecting predictions, so we consider it reasonable to run the re-training only when needed.



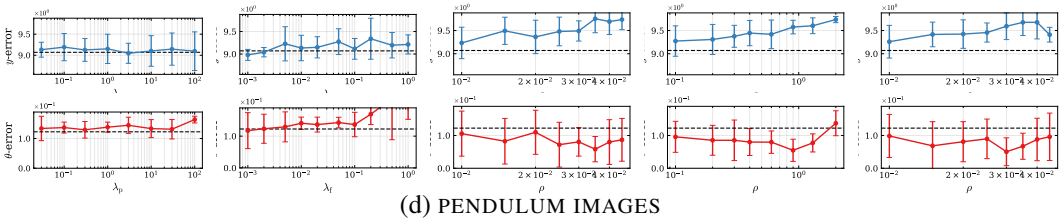
(a) PENDULUM TIME-SERIES



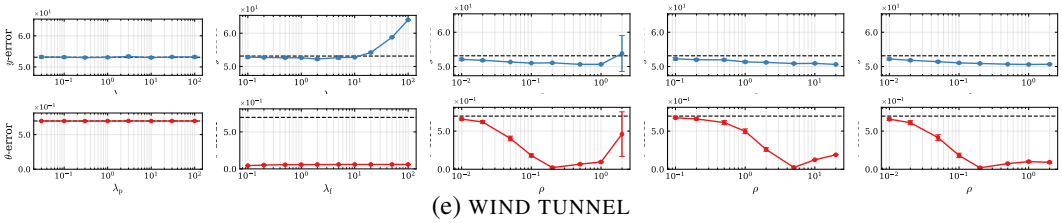
(b) REACTION-DIFFUSION



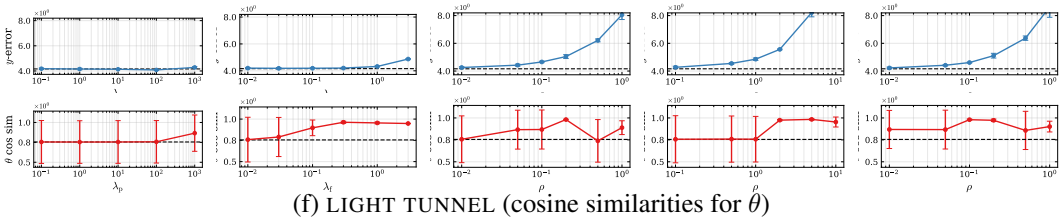
(c) DUFFING OSCILLATOR



(d) PENDULUM IMAGES



(e) WIND TUNNEL



(f) LIGHT TUNNEL (cosine similarities for θ)

Figure 5: Sensitivity of performance with regard to the hyperparameters. From left to right: `p-reg`, `f-reg`, `sam`, `asam`, and `fsam`.

Table 3: y-error before and after re-training ϕ with fixed θ.

		`erm` (Table 1)	`fsam` (Table 1)	`fsam` (ϕ re-trained)
REACTION-DIFFUSION	y-error ($\times 10^0$)	1.11 $\pm$<0.01	1.19 $\pm$0.01	1.11 $\pm$<0.01
PENDULUM IMAGES	y-error ($\times 10^0$)	9.07 $\pm$0.14	9.59 $\pm$0.29	9.13 $\pm$0.34
LIGHT TUNNEL	y-error ($\times 10^0$)	4.16 $\pm$0.06	4.61 $\pm$0.02	4.13 $\pm$0.03

Table 4: θ-error with 10 runs with different random seeds.

		`erm`	`p-reg`	`f-reg`	`fsam`
REACTION-DIFFUSION	θ-error ($\times 10^{-4}$)	8.94 $\pm$0.77	**0.93** $\pm$0.38	<u>1.42</u> $\pm$0.53	2.98 $\pm$1.90
PENDULUM IMAGES	θ-error ($\times 10^{-2}$)	14.9 $\pm$5.08	<u>14.5</u> $\pm$3.17	14.6 $\pm$5.69	**6.85** $\pm$4.90

F.4 Results with More Random Seeds

For the REACTION-DIFFUSION and PENDULUM IMAGES tasks, where the variances of the θ-error over the random seeds are large in Table 1, we ran five more random seeds (ten in total) with the same hyperparameters. Table 4 reports the θ-errors with the ten seeds; the trends remain the same as in Table 1, and so do the trends in the y-error. We ran Wilcoxon signed-rank tests (exact, two-sided, paired over the seeds) on the ten-seed θ-errors. For REACTION-DIFFUSION, `fsam` is significantly better than `erm` ($p \approx 0.002$) and significantly worse than `p-reg` ($p \approx 0.0039$) and `f-reg` ($p \approx 0.049$). For PENDULUM IMAGES, `fsam` (the best) is significantly better than `p-reg` (the second best; $p \approx 0.0039$). It is not surprising that `f-reg` outperforms `fsam` in REACTION-DIFFUSION because the hybrid model here has the simple additive structure, for which penalizing $\|g_\phi\|_2^2$ is known to work well [10]; the even better performance of `p-reg` can also be understood as an indirect way to make $\|g_\phi\|_2^2$ small. Since `fsam` is architecture-agnostic and does not directly exploit the additivity, it is natural that it underperforms `f-reg` there, but it is still clearly better than `erm`.

We also ran five more random seeds for the comparison between `fsam` and the ordinary Fisher SAM in Table 2 for the four tasks where the difference was unclear. Table 5 reports the results. The trends remain the same as in Table 2, and none of the differences is significant (the Wilcoxon signed-rank tests give $p \approx 0.63$, 0.56, 0.23, and 0.32 from top to bottom).

F.5 Sharpness along θ and the Effect of Perturbing Only ϕ

In Tables 2 and 5, perturbing only ϕ (the left column, `fsam`) is clearly better than perturbing both θ and ϕ (the right column, ordinary Fisher SAM) in the PENDULUM TIME-SERIES and WIND TUNNEL tasks. We hypothesize that in these two tasks, having a sharp loss minimum along the θ-direction is particularly important for identifying θ; consequently, perturbing θ as well by the ordinary SAM flattens the loss along θ and harms the identification.

To examine this hypothesis, we measured the sharpness ratio $r \coloneqq |\Delta_\theta L|/|\Delta_\phi L|$, where $\Delta_\theta L$ ($\Delta_\phi L$) is the change of the training loss L when only θ (only ϕ) is perturbed around a learned solution. We used relative perturbations of size $\delta = 0.01$: θ is multiplied by $1+\delta$, and each element of ϕ is multiplied by $1 + \delta s$ with a random sign $s \in \{-1, 1\}$, where $\Delta_\phi L$ is averaged over three draws of the signs. Both $\Delta_\theta L$ and $\Delta_\phi L$ are averaged over the five seeds before taking the ratio. A large r suggests that the loss around the solution is particularly sharp along θ. Table 6 shows r computed for the models trained with `fsam` in Table 2. The two largest values of r are observed exactly for the PENDULUM TIME-SERIES and WIND TUNNEL tasks, i.e., `fsam` retains the sharpness along θ in these tasks. Moreover, we confirmed that the ordinary Fisher SAM makes these values of r substantially smaller. This result supports our hypothesis that perturbing only ϕ is particularly important for these two tasks. In summary, while restricting the perturbation to ϕ is not always necessary, it can be useful as a safeguard for tasks where a flat minimum along the θ-direction harms the identification.

F.6 Why `f-reg` Fails in the DUFFING OSCILLATOR Task

In the DUFFING OSCILLATOR task, the hybrid model is additive, so `f-reg` penalizes $R_{\mathrm{f}}(\phi) \propto \|g_\phi\|_2^2$. Since g_ϕ fits the residual that f_θ cannot express, `f-reg` succeeds if the residual is minimized when $\theta = \{\tilde{\alpha}\}$ takes the data-generating value. However, this does not hold in this task. When the hybrid model

Table 5: θ-error by `fsam` and by the ordinary Fisher SAM with 10 runs with different random seeds.

		`fsam`	ordinary
REACTION-DIFFUSION	($\times 10^{-4}$)	$\mathbf{3.0}$ ± 1.9	4.1 ± 3.7
DUFFING OSCILLATOR	($\times 10^{-2}$)	$\mathbf{2.3}$ ± 1.2	2.5 ± 1.4
PENDULUM IMAGES	($\times 10^{-2}$)	6.9 ± 4.9	$\mathbf{5.3}$ ± 3.4
LIGHT TUNNEL	(cos-sim $\uparrow$)	$\mathbf{.98}$ $\pm .01$	$.97$ $\pm .02$

Table 6: Sharpness ratio r of the models trained with `fsam`. A large r means that the loss is sharp along θ relative to ϕ.

	PEND T-S	REACT-DIFF	DUFFING	PEND IMAGE	WIND	LIGHT
r	2.0×10^{3}	1.9×10^{-1}	4.1×10^{0}	1.3×10^{0}	1.1×10^{1}	5.1×10^{-1}

(6) reproduces the data-generating dynamics exactly, g_ϕ must represent $(\alpha - \tilde{\alpha})v - \beta v^3 + \gamma \dot{v}$, whose mean squared value over the data distribution is minimized at $\tilde{\alpha} = \alpha + (-\beta\,\mathbb{E}[v^4] + \gamma\,\mathbb{E}[v\dot{v}])/\mathbb{E}[v^2]$. On the training data, this minimizer is $\tilde{\alpha} \approx 1.42$, whereas the data-generating value is $\alpha = 1$; the root mean square of the residual is about 0.22 at $\tilde{\alpha} = 1$, whereas it is about 0.11 at $\tilde{\alpha} \approx 1.42$. Consistently, with a moderately large λ_f, we confirmed that the estimates by `f-reg` quickly approached $\tilde{\alpha} \approx 1.42$. Such a mismatch happens because the term of f_θ, $\tilde{\alpha}v$, and the missing term, v^3, are positively correlated on the data distribution, so making $\tilde{\alpha}$ larger than the data-generating value happens to make the residual smaller.

F.7 Computational Cost

Algorithm 1 requires two evaluations of the loss gradient per iteration, one at the current point for computing ϵ_ϕ^* and one at the perturbed point, whereas `erm`, `p-reg`, and `f-reg` require one. The inverse Fisher information in `fsam` is computed from the first gradient and needs no additional evaluation. Table 7 reports the wall-clock time per iteration of `erm` and `fsam`, measured with the configurations used for Table 1. PENDULUM TIME-SERIES, DUFFING OSCILLATOR, and WIND TUNNEL were measured on a single CPU thread (Intel Xeon Gold 6530), and the others on a GPU (NVIDIA RTX PRO 6000 Blackwell). The actual overhead is about 1.5–1.6 times rather than 2 times because the computation other than the gradient evaluation, such as the gradient of the prior and the optimizer update, is common to the two methods.

F.8 Learning with High-Dimensional Scientific Parameters

To examine whether the method scales to scientific models with many unknown parameters, we created a variant of the REACTION-DIFFUSION task with a higher-dimensional θ. In the original task, the data are generated with the scalar diffusion coefficients $a, b \in \mathbb{R}$. In the new task, the diffusion coefficients are instead spatially varying fields, i.e., the data are generated from

$$u_t = \nabla \cdot (\mathbf{a} \nabla u) + u - u^3 - \kappa - v,$$
$$v_t = \nabla \cdot (\mathbf{b} \nabla v) + u - v$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{d \times d}$ with $d = 32$ are smooth random fields (sums of Fourier modes up to wavenumber 3) taking values in $[0.0002, 0.002]$ and $[0.002, 0.008]$, respectively. The hybrid model estimates the two fields directly on the grid, i.e., $\theta = \{\tilde{\mathbf{a}}, \tilde{\mathbf{b}}\}$ with $\dim \theta = 2048$, and g_ϕ is a convolutional neural net with four layers of 32 channels and periodic padding, which receives Fourier features of the spatial coordinates in addition to the state. As in the original task, the task is to predict from an initial state to the subsequent sequence of $m_y = 5$ states; we generated 1000, 100, and 100 input-output pairs for the training, validation, and test data, respectively. The settings follow the original task, and the hyperparameters λ_p, λ_f, and ρ_ϕ were re-tuned.

Table 8 reports the estimation error of θ and the correlation coefficients between the estimated and data-generating fields. `fsam` remains effective for this high-dimensional θ; in particular, it recovers the field $\mathbf{a}$ with a much higher correlation than the other methods. Note that $\dim \theta = 2048$ is still much smaller than the number of parameters of g_ϕ, so the computational cost remains practical; whether the method scales further, e.g., to millions of scientific parameters, is left for future work.

Table 7: Wall-clock time per iteration (ms) of `erm` and `fsam`, and their ratio.

	`erm`	`fsam`	ratio
PENDULUM TIME-SERIES	14.6 ±0.21	23.6 ±0.17	1.62
REACTION-DIFFUSION	47.3 ±1.34	74.1 ±3.09	1.57
DUFFING OSCILLATOR	15.0 ±0.12	23.8 ±0.27	1.58
PENDULUM IMAGES	35.3 ±0.32	57.7 ±1.27	1.63
WIND TUNNEL	31.0 ±0.03	45.9 ±0.04	1.48
LIGHT TUNNEL	84.3 ±0.76	130 ±0.25	1.55

Table 8: Results of the REACTION-DIFFUSION task with spatially varying diffusion coefficients.

	`erm`	`p-reg`	`f-reg`	`fsam`
θ-error ($\times 10^{-1}$)	1.46 ±0.04	1.43 ±0.04	<u>1.40</u> ±0.03	**1.14** ±0.03
$\tilde{\mathbf{a}}$ corr. coeff. ↑	.74 ±.05	.75 ±.05	<u>.81</u> ±.03	**.95** ±.01
$\tilde{\mathbf{b}}$ corr. coeff. ↑	.96 ±<.01	<u>.96</u> ±<.01	.96 ±.01	**.97** ±<.01

Table 9: Results of the DUFFING OSCILLATOR task with a genuinely complex residual.

	`erm`	`p-reg`	`f-reg`	`fsam`
θ-error ($\times 10^{-1}$)	2.99 ±0.13	3.55 ±0.72	<u>2.38</u> ±0.06	**2.33** ±0.64
y-error ($\times 10^{-2}$)	7.05 ±0.09	7.22 ±0.04	7.02 ±0.08	9.93 ±0.10

F.9 Stress Test with a Genuinely Complex Residual

The proposed method relies on the simplicity of the residual that g_ϕ must fit under a good θ. To examine what happens when the true residual is genuinely complex, we created a variant of the DUFFING OSCILLATOR task, in which the data are generated from

$$\ddot{v}(t)+\gamma\dot{v}(t)+\alpha v(t)-\beta v(t)^3 = A\, c(v(t),\dot{v}(t)), \quad c(v,\dot{v}) = \sqrt{\frac{2}{K}}\sum_{j=1}^{K} s_j \sin\left(\pi\left(\frac{m_j v}{S_v}+\frac{n_j \dot{v}}{S_{\dot{v}}}\right)\right),$$

where $K = 32$, $A = 0.3$, $(S_v, S_{\dot{v}}) = (1.25, 1.75)$, (m_j, n_j) runs over all the pairs with $m_j \in \{1,\dots,4\}$ and $n_j \in \{\pm 1,\dots,\pm 4\}$, and the signs $s_j \in \{-1, 1\}$ are randomly chosen and fixed. The additional term is unknown to the learner, and the hybrid model is the same as in the original task, so the residual that g_ϕ must fit is genuinely high-order, not a simple missing term. The other settings follow the original task, except for $\rho_\phi = 0.2$ for `fsam` and $\lambda_{\mathrm{f}} = 5 \times 10^{-3}$ for `f-reg`.

Table 9 shows the results. No method estimates θ well anymore, and the clear advantage of `fsam` in the original task disappears. At the same time, `fsam` is not drawn to a θ worse than that of `erm`; the benefit vanishes, but it does no particular harm. This is a concrete example of the limitation of the proposed method (and of the existing baselines): the effectiveness of the SAM-based method relies on the residual admitting a degree of simplicity, or at least not being more complex under the correct θ than under incorrect alternatives. In this sense, by "architecture-agnostic" we only mean that the regularizer need not be redesigned for each model architecture; the effectiveness of the method still depends on the structure of the residual. Designing a simplicity-seeking method that remains effective under genuinely complex residuals is an open challenge.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The point of applying SAM for hybrid models lies in the estimation of scientific parameters, and the claims in the abstract and introduction emphasize this regard.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The method is based on Occam's razor principle, and we made it clear that it is not applicable to all hybrid models without further assumptions.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We showed the necessary assumptions clearly and provided a complete proof.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We described all the details of the datasets, models, and learning methods used in the experiment.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We clarified the full details of the synthetic data, and we used public real-world data for full reproduction of the results. The codes will be published too upon final acceptance of the paper.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All the experimental settings are described.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The performance comparison is done with the random variation taken into account.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All the experimental settings are described.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research is conducted to conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: The paper presents a fundamental methodology for hybrid model learning and has no notable direct consequences in terms of societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper presents a fundamental methodology for hybrid model learning and poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have clear references to the real-world datasets used in the experiment.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not specifically release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: We used LLMs only for writing, editing, and formatting purposes.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.