Discovering Governing Laws with World Models

Michal Novomestsky[1]

Supervised by Mehrtash Harandi[2] and Riccardo Ali[3]

[1]Department of Mechanical and Aerospace Engineering, Monash University
[2]Department of Electrical and Computer Systems Engineering, Monash University
[3]Department of Computer Science, University of Cambridge

Submitted October 19, 2026.

***Abstract*—Discovering closed-form mathematical equations from observational data is a cornerstone of scientific discovery and engineering practice, yet most symbolic regression schemes largely rely on dated combinatorial search heuristics that scale poorly with system complexity. In this work, we propose decoding the latent space of large pretrained vision and multi-physics foundation models into closed-form equations with large language models. To tap into the latent space of decoder-only foundation models, we introduce *decoder compression*, a mechanism that bottlenecks intermediate residual stream activations into compact embedding tokens. Evaluated across both analytic and challenging real-world benchmarks spanning fluid dynamics, reaction-diffusion and magnetohydrodynamics, our approach is able to consistently outperform classical genetic algorithms and end-to-end neural baselines by up to an order of magnitude in fit accuracy. Furthermore, our framework is roughly $2\times$ to $48\times$ faster than most baselines, demonstrating a scalable paradigm for AI-driven scientific discovery.**

I. Introduction

THE link between prediction and compression is a well-studied phenomenon, with [26] referring to machine learning and information theory as "two sides of the same coin". In particular, it can be shown that learning compact representations (i.e. patterns in data) is equivalent to improving predictive prowess [38], [39]. This principle is mathematically precise and abstracts away the practical details of learning.

Many works attribute the remarkable success of deep learning to this principle and hypothesise that it manifests itself in the form of compact, geometric structures embedded in a neural network's high-dimensional latent space $\mathcal{X}$. One popular theory is the data manifold hypothesis [7], [12], which posits that many real-world systems such as vision, natural language, dynamical systems, etc. which to be appear highly complex in the native input space $\mathcal{I}$ in fact lie on some low-dimensional latent manifold embedded in $\mathcal{X}$ that captures the intrinsic properties of the underlying data-generating process. Producing such a latent space $\mathcal{X}$ is the primary goal of a particular model class: the *encoder*. Encoder-only models are trained to map inputs in $\mathcal{I}$ to latent vectors $z \in \mathcal{X}$, whose compact, general features are rich enough to support a wide variety of downstream tasks [1], [3].

On the other hand, *decoder-only* models have seen remarkable success in recent years, in part contributing to the current AI boom [5], [36]. Unlike encoders, decoder-only models learn to map input states to other input states (e.g. predicting the next word in a sentence [36]). Mechanistic interpretability attempts to explain their capabilities through the view that decoders internally organise abstract mathematical rules into smooth geometric structures in intermediate layers (the residual stream). Work by [30] and [32] suggests that decoders form continuous circuit mechanisms and low-dimensional latent structures spread throughout their residual stream to represent mathematical relations and dynamical transitions.

Moving away from machine learning, the principle of data compression can also be viewed as an explanation for the success of science itself. Li and Vitányi describe the matter qualitatively [22]:

> *"Science may be regarded as the art of data compression. Compression of a great number of experimental data into the form of a short natural law yields an increase in predictive power."*

In other words, summarising vast quantities of empirical data as simple scientific theories provides us with a highly accurate (albeit incomplete) model of our universe; that is, science creates *world models*.

Coming back to machine learning, a growing body of work suggests that large, pretrained foundation models (whether encoder- or decoder-only) produce latent representations which can be viewed as models of their environment [10], [11], [14], [21], [27], [32]. This is an exciting prospect for scientific discovery, as it hints at the possibility that foundation models trained on vast quantities of physical data (such as the motion of fluids or objects flying through the air) may encode governing laws of the world in their latent space $\mathcal{X}$ (Figure 1).

This identifies a promising gap for AI in scientific discovery. Existing work largely sits in two camps: On one end of the spectrum, there has been extensive effort focusing on raw predictive power for end-to-end results, treating the model as a black box, with noteworthy examples including [17], [20], [29]. Although excellent for predicting system behaviour, this approach provides little to no insight into the underlying science. On the other end, symbolic regression aims to fit a closed form function defined by some symbolic language $\mathcal{S}$ to numerical data. However, many popular frameworks such as PySR [9] and Operon [6] are driven by genetic programming and lack priors about the invariances and governing laws of a system (unless a user explicitly enforces such assumptions). In other words, they "blindly" attempt to provide a locally

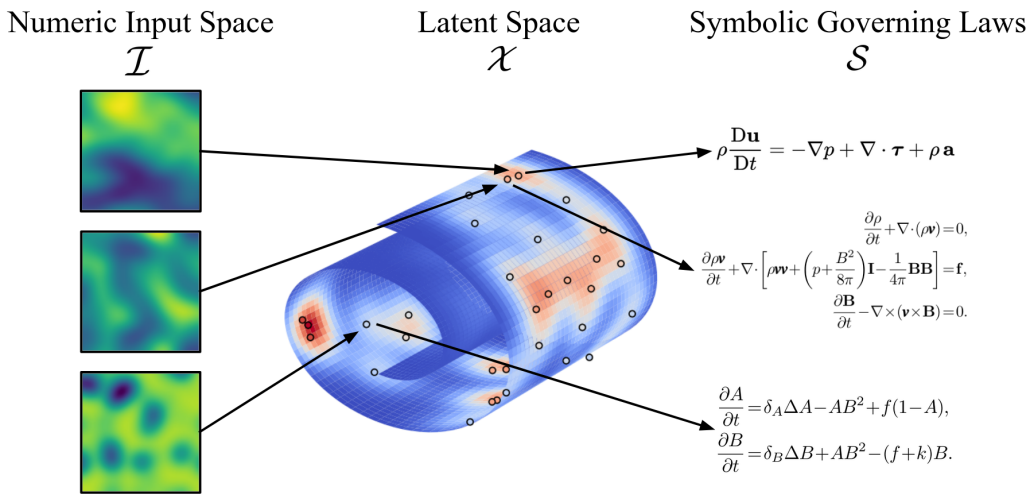


Fig. 1. Large foundation models trained on physics or image/video data depicting real-world dynamics (such as a basketball flying through the air) may implicitly represent a system's governing laws in their latent space $\mathcal{X}$. For example, two systems with qualitatively similar dynamics (e.g. compressible Navier-Stokes and magnetohydrodynamic plasmas) may lie close to one another on $\mathcal{X}$, whereas a system with substantially different dynamics (e.g. reaction-diffusion) may exist further away. Thus, it may be possible to recover the dynamics of some numeric measurements in $\mathcal{I}$ by tapping into the highly compressed structure in $\mathcal{X}$ through a learned map $D_\phi : \mathcal{X} \to \mathcal{S}$.

optimal best-fit, which struggles from long runtimes due to inefficient search and risks producing functions which generalise poorly. Some methods like NeSymReS [4] train a bespoke neural network to generate symbolic functions against numerical input. However, this involves training an entire model from scratch purely for this purpose (i.e. they learn $\mathcal{I} \to \mathcal{S}$ upfront), which may be a very sparse map and a difficult task to learn.

In this work, we bridge this gap and study whether it is possible to recover a closed-form analytic function $u(x, y, t)$ from a numerical representation U by instead learning an intermediate decoding $D_\phi : \mathcal{X} \to \mathcal{S}$, which maps a foundation model's black-boxed latent state into a symbolic law.

We demonstrate that by using pretrained foundation models trained on either domain-specific physics data or generic vision tasks, we can meaningfully learn the decoding $\mathcal{X} \to \mathcal{S}$. We show that this paradigm is able to robustly decode closed-form symbolic functions which provide strong fits for both analytical fields and highly non-linear real-world systems in science and engineering, comfortably outperforming alternative methods at a fraction of the runtime, laying out a clear path for scaling our method up to real-world use-cases and providing further evidence that pretrained foundation models learn internal world models.

A. Contributions

In summary, our main contributions are:

1) We demonstrate that we are able to recover symbolic regressions with strong performance both in and out of distribution using a wide range of foundation models, providing evidence in favour of the view that they model meaningful world dynamics.
2) We introduce **decoder compression** as a means to probe the latent space of decoder-only foundation models, and demonstrate that it is competitive with pure encoders.
3) We demonstrate that our approach outperforms existing benchmarks whilst being significantly faster, **demonstrating its viability for large-scale scientific discovery in physics, engineering and beyond.**

B. Related Work

1) **AI for Scientific Discovery:** Machine learning applications in the natural sciences largely span a spectrum between black-box numerical approximation and exact symbolic discovery. High-capacity surrogate models have achieved state-of-the-art predictive performance in an impressive range of domains, including protein folding [17], weather forecasting [20] and fluid dynamics [29], yet do little to shed light on the underlying dynamics of the systems they model. Conversely, symbolic methods extract governing equations directly from numerical measurements [6], [9], [37], but can struggle to recover canonical laws, instead producing overfit functions. Our approach bridges this divide by using a learned decoding to read out closed-form functions directly from the compressed representations of pretrained foundation models.
2) **Symbolic Regression** aims to uncover analytical, closed-form expressions from numerical data. Traditional symbolic regression frameworks rely on genetic programming and evolutionary search heuristics [6], [9], [37], which scale poorly with expression complexity and lack domain priors. To accelerate search, some deep learning approaches utilise sequence-to-sequence transformers to predict symbolic expression trees directly from input data trajectories [4], [18], [34]. These end-to-end models typically map raw numerical data directly to symbolic syntax ($\mathcal{I} \to \mathcal{S}$). However, this map is likely to be extremely sparse and may prove to be a challenging task to learn. Our paradigm avoids such

pitfalls by decoding expressions from rich pretrained latent spaces instead ($\mathcal{X} \to \mathcal{S}$).

3) **World Models** refer to a class of model broadly understood to develop some compact representation of its environment, typically encoded within a latent space (e.g. [14]). [21], [30], [32] argue that LLMs implicitly learn such representations when trained to complete end-to-end tasks. More recently, there has been an effort to create bespoke world models such as JEPA [27] and Genie [10], trained uniquely for this purpose. Of particular interest to us is [11], which demonstrated that a physics-specific foundation model (Walrus) learns surprisingly generic structures in its latent space, suggesting that it might have developed a more general "understanding" of physical systems.
4) **Multimodal Language Models** endow LLMs with the ability to perceive other modalities such as vision [1], [24] and audio [25]. These approaches frequently make use of two pretrained models: an LLM and an encoder for the incoming modality, which are typically fused using some adapter. Flamingo [1] introduced the Perceiver Resampler, a popular choice of adapter which we make use of in our work.

II. Methodology

A. Problem Setting

We consider the problem of symbolic regression. That is, given some set of systems $\mathcal{U}$ with numerical representation $\mathcal{I}[\mathcal{U}]$ and symbolic form $\mathcal{S}[\mathcal{U}]$, we wish to obtain the map $\mathcal{I} \to \mathcal{S}$.

We define our symbolic language $\mathcal{S}$ as the family of functions $u \in \mathcal{U}$ such that:

$$\mathcal{S}[u] : u(x, y, t) = a + \sum_{i=1}^{n} b_i f_i(x) g_i(y) h_i(t) \tag{1}$$

which is defined on the square $[0, L]^2 \times [0, T]$, with $L = T = 1$, $n \leq 2$ and (f, g, h) sampled from the basis $\{1, s, s^2, \sin(\pi s), \sin(2\pi s), \cos(\pi s), \cos(2\pi s), \exp(-s), \exp(-2s), \tanh(2s-1)\}$. a and b_i are normalising constants. The equivalent numerical form is thus the grid $\mathcal{I}[u] : U \in \mathbb{R}^{N_x \times N_y \times N_t}$. We pick this family of functions since it is:

1) analytic and well-behaved.
2) contains no singularities (which could blow up U).
3) easy to generate synthetic training data for.

We decompose the map as $\mathcal{I} \to \mathcal{X} \to \mathcal{S}$, where $E_\theta : \mathcal{I} \to \mathcal{X}$ is the latent embedding map induced by a foundation model E_θ. What remains is to learn a decoding $D_\phi : \mathcal{X} \to \mathcal{S}$ which translates a latent representation $z = E_\theta(U) \in \mathcal{X}$ into a human-interpretable, symbolic function $u(x, y, t)$.

Thus, we train a downstream decoding D_ϕ to predict the symbolic form $u(x, y, t)$ given its numerical representation U, that is, $u(x, y, t) \sim D_\phi(E_\theta(U))$.

B. Foundation Models

To evaluate how latent representation choices impact symbolic decoding performance, we select four distinct foundation models, summarised in Table I. Of particular interest are two variables: model domain (generic vs. physics-specific) and architecture (encoder-only vs. decoder-only).

TABLE I
Candidate Foundation Models

Model	Type	Physics-specific	Params	Pretraining Regime
Walrus [29]	Dec	✓	1.3B	Next patch MSE.
CLIP ViT-B/16 [35]	Enc	✗	86M	Image-text contrastive loss.
DINOv2 Giant [33]	Enc	✗	1.1B	Teacher-student loss with exponential moving averages.
I-JEPA [2]	Enc	✗	600M	Joint embedding predictive loss.

We consider three image encoders trained on open internet data (CLIP, DINO and JEPA) as well as one decoder-only model trained specifically on physics data (Walrus). These models span a wide array of sizes and pretraining regimes. "M" indicates million and "B" billion.

Model domain is motived by [28], which demonstrated that even generic vision foundation models exhibit strong performance on downstream physics tasks compared to training from scratch, which could be a result of these models implicitly learning weak physical laws as a byproduct of training on internet-scale video data. As such, we compare domain-agnostic vision models against physics-specific foundation models to investigate whether pretraining on physical systems meaningfully improves decoding capability.

Despite being commonplace in computer vision and natural language processing, foundation models remain rare in physics. To the best of our knowledge, the only truly multi-physics foundation models available in the public domain are Poseidon [15] and Walrus [29], with the latter generally regarded as more recent and comprehensive. However, these are both decoder-only models. Motivated by this lack of conventional encoders, we compare encoder embeddings against decoder residual streams as candidates for $\mathcal{X}$ (Figure 2b), as outlined in II-C.

C. Architecture

We pass numeric input U through a foundation model to obtain its latent tokens $z = E_\theta(U)$, $z \in \mathbb{R}^{n \times d_h} = \mathcal{X}$, which we decode into a symbolic function $u(x, y, t)$ using an LLM. Since the amount of these embeddings n is typically quite large and needlessly sparse when translating into text, we save on memory by first passing the embeddings through a Perceiver Resampler [1] to further compress the embeddings into eight compressed latents z^* which finally enter the LLM. That is, the LLM and perceiver collectively form D_ϕ to complete the $\mathcal{I} \to \mathcal{X} \to \mathcal{S}$ map $u(x, y, t) \sim D_\phi(E_\theta(U))$ (Figure 2a).

When E_θ is decoder-only, it is unlikely that simply using the output of its final layer is sufficient to define z, as [30] and [23] both find that using a larger portion of the residual stream better-resolves the model dynamics. As

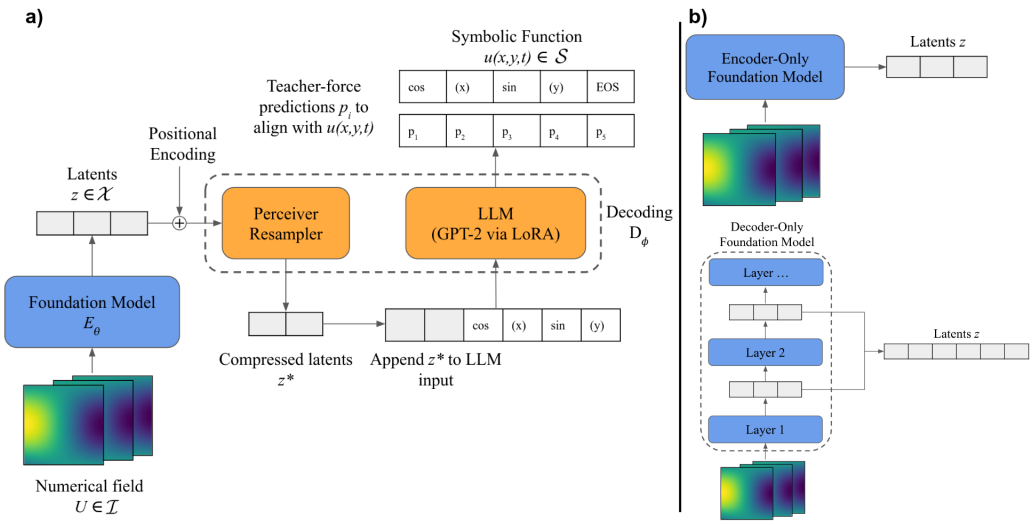


Fig. 2. a) End-to-end architecture. A sequence of 2D snapshots in time U are ingested by a frozen foundation model E_θ, which maps them into a series of n latent embedding tokens $z \in \mathbb{R}^{n \times d_h} = \mathcal{X}$. To save on memory and limit redundancy, these tokens are first compressed into a tight bottleneck z^* which is then prefix-appended onto LLM input. The decoding D_ϕ is teacher-forced on the symbolic function $u(x, y, t)$. Blue indicates a frozen module while orange means it is trainable. b) Encoder- vs decoder-only models. Whereas encoder-only foundation models are trained to natively return embeddings z as their output, decoder-only models are instead designed to generate data in the input space (e.g. the next step in a sequence). So in this case, we define z as the concatenation of all latents in its residual stream.

such, we define a decoder latent state as the concatenation $z = z_1 \oplus z_2 \oplus ... \oplus z_{l_D} \in \mathbb{R}^{l_D \times S \times d_h}$, where $z_i \in \mathbb{R}^{S \times d_h}$ are the outputs of model layer $i \in \{1, 2, ..., l_D\}$ (that is, z contains $n = l_D S$ tokens in total; see Figure 2b). In practice however, we subsample the full set of tokens with a layer stride of 2 and spatial stride 4 to save memory. Since many tokens in the residual stream are likely to be redundant (particularly in the earlier layers of the decoder, which primarily encode localised structure such as edges), we once again benefit from compressing z with a perceiver resampler. We call this procedure *decoder compression*.

All foundation models used accept multi-channel inputs. While the generic vision encoders accept RGB input, Walrus accepts a variety of domain-specific channels (pressure, velocity, etc.). Since $u(x, y, t)$ is a scalar field, we insert it into the R channel in vision models and the "concentration" channel in Walrus (which we take to be the most generic quantity available). All other channels are set to 0.

To allow the perceiver resampler to discern tokens, we apply learned, additive positional encodings to identify:

1) Layer index i for decoder-only models.
2) Spatial grid index j, since earlier layers in decoders may not have encoded the spatial position of each token yet. We also apply this to encoder-only models for consistency, though [1] shows this is likely to be unnecessary.
3) Temporal index k, since the vision encoders (CLIP, DINO and I-JEPA) are *image* encoders, so we must specify causality. Once again, this may also help earlier layers in decoders.

D. Training and Inference

Similarly to [4], we train D_ϕ by teacher-forcing against $u(x, y, t)$'s text tokens. Since we are interested in the map $\mathcal{X} \to \mathcal{S}$, we keep E_θ frozen during this procedure and only train D_ϕ. We use GPT-2 finetuned with LoRA [16] as the LLM, with the perceiver resampler being trained from scratch. We train for 200k steps.

To train D_ϕ, we create a synthetic dataset $D = \{(U_i, u_i(x, y, t))\}_{i=1}^n$ of $n = 10^6$ functions $u_i(x, y, t)$ randomly sampled from Equation 1. The tensors U_i were sampled at a resolution of 128×128 pixels for six timesteps.

This dataset is split into mutually-exclusive training and held-out evaluation sets D_t and D_h, of size $\sim 10^6$ and 127 respectively. We ensure this split is strict by enforcing that the mean-square error (MSE) between the functions in D_h and D_t is $\leq 10^{-6}$, which avoids data leakage from functional equivalence (e.g. $\sin(2x) = 2\sin(x)\cos(x)$, etc.).

At inference, the predicted symbolic function is compiled and passed to an L-BFGS-B optimiser to fit constants. That is, our model predicts the *form* of $u(x, y, t)$, whereas precise coefficients are offloaded to conventional optimisers. This is standard practice in symbolic regression [4], [9]. We generate 64 samples in parallel at an LLM temperature of 0.75 and pick the best-scoring fit based on L-BFGS-B residual.

However, our models *are* trained to predict constants to two decimal places, rather than to full float-point precision. We do this for two reasons:

1) Empirically, we find that around 80% of the total loss was concentrated on predicting negligible decimal places, so truncating to two limits the strain on the model.
2) However, entirely replacing floats with constant symbols c introduces two challenges. First, geometry which is strongly dependent on c is lost in this process (e.g. $\sin(0.1x)$ and $\sin(100x)$ appear very different on $x \in [0, 1]$), which may make it harder for the model to discern features. Second, training the model to generate

approximate constants also provides a good initialisation for faster L-BFGS-B convergence, especially on non-linear functions with strongly non-convex loss geometry.

Finally, we remove scale from any input using Reversible Instance Normalization (RevIN) [19] (consistent with Walrus' own internal normalisation [29]). Since RevIN is an affine map, it amounts to simply updating the constants a, b_i in Equation 1 to align $u(x, y, t)$ with the updated U.

III. Experiments

To address our broader question of whether symbolic functions can be recovered from a foundation model's latent space, we consider the following fine-grained questions:

1) Does our approach extract governing laws from numerical data in a way that's competitive against existing approaches?
2) Can we apply our approach to real-world systems and get meaningful results?

To answer 1), we consider a head-to-head comparison between our method and existing alternatives with no handicaps. Since this question is interested in raw performance alone, we conduct this comparison against U which are spanned by Equation 1 so D_ϕ is able to generate expressions $u(x, y, t)$ which fully capture the system dynamics (i.e. it is in-distribution).

For 2), we are interested in whether we still recover meaningful results for real-world, out-of-distribution (OOD) examples spanning domains such as fluids, chemistry and nuclear physics. Such systems are unlikely to have closed-form solutions, let alone ones which sit within the span of Equation 1. Instead, we are interested in seeing whether a) our $\mathcal{X} \to \mathcal{S}$ decoding is generic enough to capture reasonable approximations of these systems and b) if these approximations are superior to alternatives. For an apples-to-apples comparison, we only consider alternative methods which can be restricted to a search space approximately identical to Equation 1.

A. Datasets

By construction, the held-out dataset D_h is spanned by Equation 1, meaning that D_ϕ is capable of producing expressions exactly equal to U sampled from this dataset, so we use it to address question 1).

To study 2), we use a collection of 170 samples taken from PDEgym [15] and The Well [31], selected for their wide scope across the sciences and proximity to the 128×128 input resolution. In particular, we consider the following systems:

1) **CE-Gauss (PDEgym):** The compressible Euler (CE) equations describe inviscid compressible flow in two dimensions, coupling density, momentum and energy [15]. This dataset yields density ρ, velocity $\mathbf{v}$, pressure p and total energy E in 2D.
2) **NS-PwC (PDEgym):** The incompressible Navier-Stokes equations govern near-inviscid fluid flow under the divergence-free constraint [15]. The dataset yields velocity $\mathbf{v}$ and a passive tracer c_{tr} in 2D. Initial vorticity is piecewise constant on a 10×10 partition, from which the velocity is recovered. This initial condition produces complex vortex interactions.
3) **Gray-Scott Reaction-Diffusion (The Well):** The Gray-Scott equations model the behaviour of two interacting chemical species A and B. The equations yield 2D fields $A(x, y, t)$ and $B(x, y, t)$ describing the concentrations of their respective species. Varying equation parameters can produce a diverse range of qualitative structures, including gliders, bubbles, mazes, worms, spirals and spots [13], [31].
4) **MHD-64 (The Well):** The ideal magnetohydrodynamic (MHD) equations govern the dynamics of plasmas, which couple compressible fluids with electromagnetic forces [8], [31]. These equations yield density ρ, velocity $\mathbf{v}$ and magnetic field $\mathbf{B}$ in 3D. To map to 2D, we take the slice $z = 0$. Additionally, since this dataset has a resolution of only 64, we upsample to 128.

TABLE II
Real-World Datasets

System	Dataset	Native Resolution	Timesteps	Quantities
CE-Gauss	PDEgym [15]	128×128	21	ρ, v_x, v_y, p, E
NS-PwC	PDEgym [15]	128×128	21	v_x, v_y, c_{tr}
Gray-Scott	The Well [31]	128×128	1001	A, B
MHD-64	The Well [31]	$64 \times 64 \times 64$	100	ρ, v_x, v_y, v_z, B_x, B_y, B_z

We take snapshots from four systems. While CE-Gauss and NS-PwC consider fluids, MHD-64 and Gray-Scott model plasmas and chemical reactions respectively. Between them, these four datasets span a wide range of geometries and qualitative phenomena.

These datasets are summarised in Table II. Since we only consider scalar symbolic regression, we take each quantity from these datasets as seperate inputs (e.g. B_z is taken in isolation from the rest of $\rho, \mathbf{v}, \mathbf{B}$ in MHD-64). Additionally, since these systems contain high-frequency structures which cannot be predicted with the limited number of terms in our symbolic regressions, we smooth these systems by applying Gaussian blur with $\sigma = 8$.

B. Benchmarks

To study the efficacy of our method, we compare against three types of baseline:

1) **Classical Symbolic Regressors:** PySR [9] and Operon [6] are two popular symbolic regression frameworks driven by genetic algorithms. Their overall effectiveness and popularity makes them a good representation of the status quo in symbolic regression.
2) **Neural Symbolic Regressors:** Several works investigate the potential of deep learning in symbolic regression, such as [4], [18], [34]. We use NeSymReS 10M and 100M [4] as representative baselines.
3) **Sparse Linear Basis:** As a default base-case which we should outperform to attain a meaningful result, we

consider a classical sparse linear fit of Equation 1's basis terms. We greedily select factors $f(x)g(y)h(t)$ by matching pursuit on residual variance, then jointly least-squares refit their coefficients with an additive constant.

To additionally provide an apples-to-apples comparison, we also consider NeSymReS after it has been finetuned on our training dataset for 200k steps (NeSymReS FT) as well as PySR with its search space restricted as closely as possible to Equation 1 (PySR Basis). In this sense, we consider PySR on default settings, Operon and NeSymReS without finetuning to be *unconstrained* methods, as they can produce fits not spanned by Equation 1.

IV. Results

We present quantitative results for accuracy and runtime in Tables III and IV respectively. Across both benchmarks, our approach consistently achieves either the best or second-best performance across all metrics.

We observe the qualitative reconstruction fidelity in Figure 3, where our improved performance is reflected in more physically sensible solution geometry. Despite being constrained to a basis which only spans the held-out set, our models produce remarkably faithful reconstructions of highly-complex non-linear systems, which even unconstrained methods struggle to capture.

Despite the clear increase in performance, our method runs in a fraction of the time, far outperforming all but sparse linear search on runtime (Table IV).

A. In-Distribution Held-Out Performance

On the held-out dataset D_h, where exact closed-form expressions exist within the span of our functional basis (Equation 1), our candidate world models achieve state-of-the-art fitting accuracy, outperforming almost all baseline algorithms by over an *order of magnitude* in MSE, with the sparse linear fit being the only exception. Specifically, the physics decoder Walrus achieves the lowest overall held-out error (MSE $= 0.00205 \pm 0.00135$), followed closely by our other foundation models (generic image encoders).

Since the held-out set has known closed-form solutions, we can meaningfully compare symbolic regressions across baselines. We present one such example in Table V. All four foundation models successfully recover expressions which yield near-exact agreement with the dominant ground-truth terms, whereas most baselines struggle to find a canonical match.

To assess whether training on in-distribution data gives our models an unfair advantage, we also compare against baselines which have a bias towards Equation 1 (Sparse Linear, PySR Basis and NeSymReS FT). While a sparse search remains competitive with our approach, even recovering a faithful symbolic form in Table V, both PySR and NeSymReS trail far behind in both MSE and symbolic form.

It is worth noting that we used a beam size of 4 for NeSymRes predictions (lowered from the default 32 [4]), potentially leading to poorer performance on NeSymReS. However, this was necessary to prevent its fit times from exceeding 1000 seconds per trajectory. In comparison, our method is able to obtain frontier performance in just 3-8 seconds.

B. Real-World Out-of-Distribution Generalisation

Our models remain useful outside of their training regime, retaining strong generalisation capabilities across all OOD real-world benchmarks, obtaining state-of-the-art results on three of the four datasets and placing second on the fourth (NS-PwC) (Table III).

Unconstrained symbolic regression algorithms (Operon, PySR Default, and NeSymReS without finetuning) are free to compose arbitrary functions which sit outside our basis (e.g., $\exp(\sin(\tanh(x^2))))$. As such, comparing our basis-restricted models against such algorithms in the OOD dataset is inherently biased toward unconstrained algorithms, as the ground truth may be better approximated by arbitrary functions which D_ϕ is not trained to compose. To correct for this bias, we exclude such algorithms from Tables III and IV on the OOD set. Nonetheless, running unconstrained models on the OOD dataset revealed that Operon was capable of outperforming our basis-restricted models with arbitrary functions not spanned by our basis (albeit at slower runtimes), while PySR Default yielded performance comparable to our approach. Conversely, NeSymReS without fine-tuning produced predictions substantially inferior to all other evaluated methods (Figure 3).

C. Computational Efficiency and Runtime

As reported in Table IV, our framework exhibits exceptional computational efficiency. Although the sparse linear method records the lowest overall fit time ($0.06 - 0.10$ s) due to only running a small linear algebra solve, it scales poorly with system complexity (V-B). Our foundation model approach on the other hand has no such limitations and is substantially faster than all other symbolic regression paradigms. All of our models complete inference and coefficient optimisation in approximately 3 to 8 s. This represents a $2\times$ to $7\times$ speed-up over PySR (~ 20 s), $2\times$ speed-up over Operon (~ 6 s) and a $10\times$ to $48\times$ speed-up over NeSymReS FT (~ 82 to 145 s).

V. Discussion

A. Implicit World Models

Our experimental findings clearly indicate that training a downstream decoding D_ϕ to convert a foundation model's latent states into symbolic functions yields remarkable in-distribution and very strong out-of-distribution performance.

One explanation for this could be that foundation models, whether trained explicitly on physics data or on generic internet images, compress the input field U onto a rich latent geometry. In this latent space, certain input features (such as waves, blobs or moving particles) may have highly separable and compact representations, which may make learning $D_\phi : \mathcal{X} \to \mathcal{S}$ a comparatively simple task. Under this view, $\mathcal{X} \to \mathcal{S}$ could be seen as a *translation* between two approximately equivalent representations of the world; that is, the upstream foundation models E_θ appear to exhibit *world model*-like behaviour in their latent space. This aligns with

TABLE III
MSE (↓) BETWEEN PREDICTED $u(x, y, t)$ AND GROUND TRUTH FIELD U

Model	Held-out	CE-Gauss	NS-PwC	Gray-Scott	MHD-64
NeSymReS 10M	1.04 ± 0.132	N/A	N/A	N/A	N/A
NeSymReS 100M	0.927 ± 0.123	N/A	N/A	N/A	N/A
NeSymReS 10M FT	0.23 ± 0.0432	0.411 ± 0.053	0.376 ± 0.0763	0.661 ± 0.111	0.728 ± 0.0328
NeSymReS 100M FT	0.234 ± 0.0432	0.433 ± 0.0485	0.491 ± 0.0744	0.667 ± 0.115	0.768 ± 0.0304
Operon	0.147 ± 0.0324	N/A	N/A	N/A	N/A
PySR Basis	0.169 ± 0.0331	0.316 ± 0.039	$\mathbf{0.195 \pm 0.0513}$	0.601 ± 0.0933	0.564 ± 0.0299
PySR Default	0.141 ± 0.0272	N/A	N/A	N/A	N/A
Sparse Linear	0.00951 ± 0.00434	0.319 ± 0.03	0.233 ± 0.0474	0.54 ± 0.0777	0.536 ± 0.023
CLIP (ours)	0.00718 ± 0.00885	0.317 ± 0.0433	0.211 ± 0.0309	0.549 ± 0.0569	$\underline{0.503 \pm 0.0241}$
JEPA (ours)	0.0167 ± 0.00611	$\underline{0.299 \pm 0.0348}$	0.228 ± 0.0342	0.54 ± 0.0691	0.51 ± 0.0228
DINO (ours)	$\underline{0.00638 \pm 0.00544}$	$\mathbf{0.282 \pm 0.0358}$	0.207 ± 0.0299	$\underline{0.535 \pm 0.0513}$	$\mathbf{0.473 \pm 0.0217}$
Walrus (ours)	$\mathbf{0.00205 \pm 0.00135}$	0.369 ± 0.0493	$\underline{0.203 \pm 0.0369}$	$\mathbf{0.533 \pm 0.0705}$	0.52 ± 0.0242

TABLE IV
FIT TIME IN SECONDS (↓)

Model	Held-out	CE-Gauss	NS-PwC	Gray-Scott	MHD-64
NeSymReS 10M	205 ± 13	N/A	N/A	N/A	N/A
NeSymReS 100M	205 ± 9.15	N/A	N/A	N/A	N/A
NeSymReS 10M FT	85.8 ± 8.99	134 ± 15.9	145 ± 20.4	106 ± 28.1	110 ± 11.6
NeSymReS 100M FT	90.5 ± 10.3	98.5 ± 10	127 ± 23.6	82.3 ± 18.8	104 ± 11.2
Operon	5.84 ± 0.303	N/A	N/A	N/A	N/A
PySR Basis	25.1 ± 0.189	24.2 ± 0.392	23.7 ± 0.492	23.6 ± 0.452	23.3 ± 0.236
PySR Default	22.6 ± 0.204	N/A	N/A	N/A	N/A
Sparse Linear	$\mathbf{0.0671 \pm 0.0154}$	$\mathbf{0.103 \pm 0.0635}$	$\mathbf{0.0859 \pm 0.00966}$	$\mathbf{0.0877 \pm 0.00759}$	$\mathbf{0.086 \pm 0.00448}$
CLIP (ours)	3.46 ± 0.277	5.47 ± 0.579	$\underline{3.61 \pm 0.316}$	7.09 ± 0.991	6.37 ± 0.641
JEPA (ours)	3.74 ± 0.285	$\underline{4.75 \pm 0.439}$	4.08 ± 0.31	7.36 ± 1.03	$\underline{5.8 \pm 0.462}$
DINO (ours)	$\underline{3.29 \pm 0.242}$	6.2 ± 0.93	4.99 ± 0.375	7.43 ± 0.961	6.9 ± 0.501
Walrus (ours)	3.35 ± 0.282	5.61 ± 0.502	5.59 ± 0.99	$\underline{6.43 \pm 1.1}$	5.86 ± 0.489

Experimental results. We report mean scores with 95% CI. Best is **bolded** and second-best is underlined. Our method achieves strong performance across both accuracy and runtime, achieving state-of-the-art accuracy in all cases except NS-PwC, where it places second. Our method is either best or second-best across all metrics.

the mechanistic school of thought which argues that smooth geometric structures in decoders are weakly equivalent to symbolic algorithms [30], [32].

However, it remains an open question as to whether $\mathcal{X}$ contains implicit governing laws which D_ϕ taps into to aid in decoding symbolic functions or whether it is simply performing naïve pattern matching. One piece of evidence supporting the former is that, even when trained on our dataset, NeSymReS performs substantially worse than our approach. This result still holds even when taking model size into account (CLIP clearly outperforms NeSymReS 100M FT despite having just 86M parameters). This seems to suggest that learning a map from a pretrained foundation model's latent space $\mathcal{X} \to \mathcal{S}$ is somehow superior to a direct input-symbol map $\mathcal{I} \to \mathcal{S}$, which may suggest that $\mathcal{X}$ is encoding deeper structure.

B. Sidestepping the Curse of Dimensionality

Throughout our experiments, a simple sparse linear fit remained surprisingly competitive against our approach, on occasion outperforming some of its more sophisticated peer baselines.

While sparse linear fits are fast on a small candidate basis, they suffer from severe combinatorial explosion when scaled to larger basis sets, nested functional compositions, or multi-variable systems. As the search space grows, classical linear search algorithms suffer from rapid growth in memory and runtime. As such, although it serves as a meaningful baseline for our purposes, it rapidly becomes infeasible as complexity grows.

Although they don't run out of memory, genetic algorithms also suffer from ballooning runtimes with combinatorial growth in the search space.

Neural methods on the other hand elegantly sidestep this issue entirely. By encoding inputs into a dense latent space, we can describe U in great detail with a few high-dimensional vectors. In addition, we also enjoy the benefits of using LLMs to decode $u(x, y, t)$, which excel at modelling combinatorially complex sequences such as natural language. Consequently, adding complexity to the symbolic basis has a minimal impact on our resource consumption at both train and test-time, allowing our paradigm to scale effortlessly to more complex systems.

C. Comparative Analysis of Foundation Model Architectures

Although the accuracies among our foundation models all cluster relatively closely to each other (Table III), subtle performance variances do emerge between specific architectures. For example, Walrus and DINOv2 are superior in

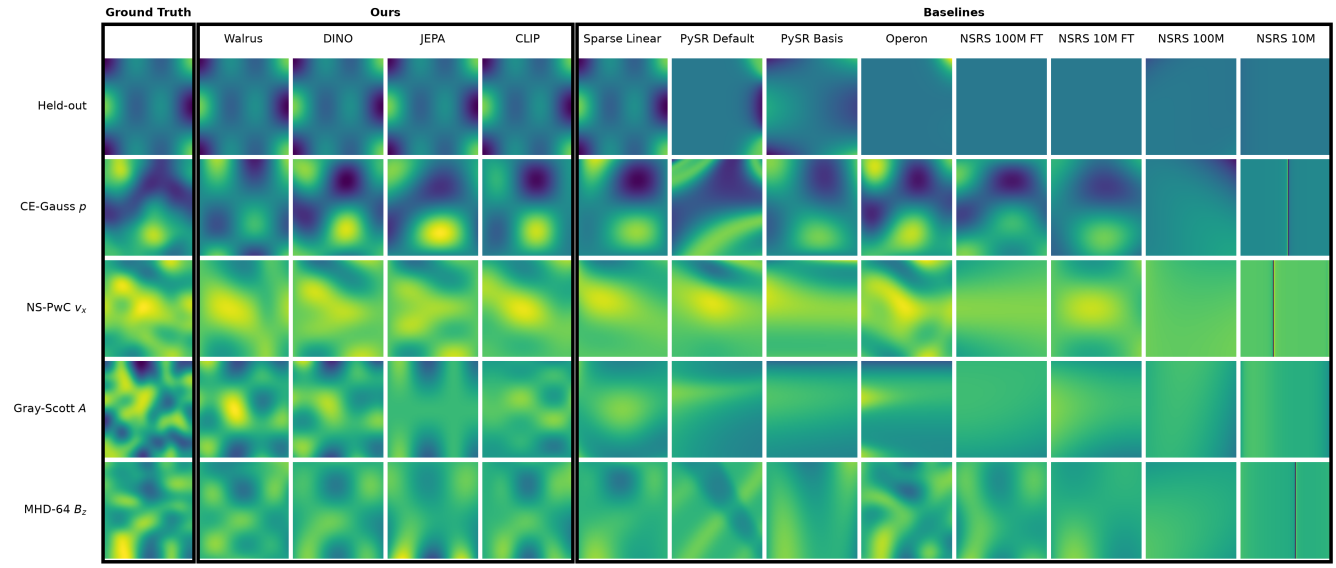


Fig. 3. Example snapshots from each dataset. NSRS is NeSymReS. Across both the in-distribution held-out set and OOD real-world datasets, our method is capable of producing strong reconstructions, despite being limited to a basis which only spans the held-out set. PySR Default, Operon and NeSymReS without finetuning are not constrained to our basis and can therefore produce more exotic functions (e.g. $\exp(\sin(\tanh(x^2)))$), which biases the OOD datasets in their favour. As such, we exclude them from the OOD datasets in Tables III and IV. Due to this additional flexibility, Operon is able to produce skillful reconstructions in some cases, whereas NeSymReS is hampered by producing regressions containing singularities (visible as black lines) due to including $\tan(x)$ in its predictions.

TABLE V
HELD-OUT SYMBOLIC EXPRESSIONS

Ground Truth	`5.11*cos(pi*t)*cos(2*pi*x)*cos(2*pi*y)*tanh(2*x - 1) + 0.64*x**2*y**2*sin(pi*t)*sin(pi*x)*sin(pi*...`
Walrus (ours)	`5.09*cos(pi*t)*cos(2.00*pi*x)*cos(2.00*pi*y)*tanh(2.00*x - 1.00) - 0.93*x*exp(-2.00*x)*exp(-1.00*...`
DINO (ours)	`5.13*cos(pi*t)*cos(2.00*pi*x)*cos(2.00*pi*y)*tanh(2.00*x - 1.00) + 0.83*x**2*y**2*exp(-2.00*y)*si...`
JEPA (ours)	`5.11*cos(pi*t)*cos(2*pi*x)*cos(2*pi*y)*tanh(2*x - 1)`
CLIP (ours)	`5.11*cos(pi*t)*cos(2.00*pi*x)*cos(2.00*pi*y)*tanh(2.00*x - 1.00) + 0.58*x*y**2*exp(-2.00*t)*exp(-...`
Sparse Linear	`5.11*cos(pi*t)*cos(2*pi*x)*cos(2*pi*y)*tanh(2*x - 1) + 0.64*x**2*y**2*sin(pi*t)*sin(pi*x)*sin(pi*...`
PySR Default	`tanh(x * ((t + (x * -0.52)) * ((((x * (((x + -0.69) * 2.24) + 0.67)) + -0.15) * ((((((y + -0.75) ...`
PySR Basis	`(-6.27 + (sin(y) * 32.22)) * ((t - 0.45) * ((((((y - 0.48) * -3.17) + 0.99) * (-0.52 + x)) - -0.00))`
Operon	`-309.31*tanh(cos(cos(-0.80*t + 1.28*y + cos(tanh(0.21*y)) + 0.54 - exp(-0.84*y))*tanh(1.16*x*(-1....`
NeSymReS 100M FT	`-21.06*t*x**2*y**2*sin(0.15*t)*sin(0.11*y) + 2.04*sin(5.48*t)*cos(0.00*x)*cos(6.35*y)*tanh(2*x - ...`
NeSymReS 10M FT	`61.84*sin(0.02*t)*cos(0.00*x)*cos(2.84*x)*cos(5.98*y) - 6.48*t*x**2*y*sin(5.36*t)*cos(2.21*x)*tan...`
NeSymReS 100M	`-1.71*(-0.85*x + y)**5 - 0.20*x + 0.09`
NeSymReS 10M	`tan(0.36*(1.58*x - 1)**5)`

Symbolic functions $u(x, y, t)$ for the held-out example in Figure 3 truncated to 100 characters. Terms are sorted in descending order of coefficient magnitude (i.e. dominant terms first). Qualitatively, it is plain to see that our method as well as sparse linear search capture the dominant `5.11*cos(pi*t)...` term with excellent accuracy, with the rest struggling to do so. NeSymReS FT, which has been finetuned on the same training data, is able to capture the general form, however key errors in the functional form of its predictions cause its numeric optimiser to fit poor coefficients.

overall performance, sharing the top-scoring MSEs among the foundation models. We hypothesise that this stems from their underlying pretraining objectives: Walrus's next-patch MSE loss and DINOv2's dense patch-level teacher-student distillation force the models to preserve high-frequency spatial details and fine-grained local gradients. In contrast, CLIP's global text-alignment and I-JEPA's abstract vector-prediction losses may throw away fine spatial variations. Because non-linear systems often depend heavily on fine structures, retaining high-frequency structural information in $\mathcal{X}$ may provide a stronger inductive bias for symbolic recovery. In addition, it is quite possible that being pretrained on physics data, the domain-specificty of Walrus also contributes to its elevated performance (consistent with findings from [28]). However, because Walrus (1.3B) and DINOv2 (1.1B) are significantly larger than CLIP (86M) and I-JEPA (600M), model size remains a confounding variable.

D. Efficacy of Decoder Compression

The competitive performance of Walrus validates our proposed *decoder compression* mechanism. Our results demonstrate that it is able to outperform encoder-only foundation models which have been specifically trained to generate strong latent representations. This result aligns with mechanistic interpretability literature [30], [32], supporting the belief that meaningful structure exists within the residual stream.

Our results also demonstrate that decoder compression is a useful tool in practice, enabling a way to adapt decoders for downstream use-cases, which may have applications in

fields such as mechanistic interpretability, model adaptation and multi-token prediction [1], [23], [30], especially as the field increasingly moves toward decoder-only architectures.

E. Limitations

While our results demonstrate great promise for discovering governing laws by decoding foundation model latent spaces, a few limitations suggest possible future improvement:

1) **Confounding Architectural Variables:** The limited scope of open-weight physics foundation models makes it difficult to isolate the effects of model size, domain and architecture. As such, we encourage the community to continue to develop and open-source these models, especially physics *encoders*, to enable a more granular isolation of these factors.
2) **Temporal Encoding in Vision Backbones:** Unlike Walrus, the vision encoders (CLIP, DINOv2, I-JEPA) are image-only and thus lack native temporal attention mechanisms across frame sequences, delegating temporal dynamics entirely to D_ϕ. Incorporating *video* foundation models (e.g. V-JEPA [3]) could shed further light on whether foundation models learn governing dynamics.
3) **Ablation Studies:** Several variables which could play a role in determining performance went unstudied. Noteworthy examples include: is the entire decoder residual stream necessary or do just a few layers do most of the work? How tight should the perceiver bottleneck be? What is the effect of normalisation (RevIN vs alternative methods)? etc.

VI. Future Work

Our results lay out a clear roadmap for scaling our method up to real-world scientific and engineering use-cases. In particular, we identify the following as points of interest:

1) **Extend to PDEs:** Many real-world problems in physics and engineering are often succinctly described by a partial differential equation (PDE), whose form provides deep insights into the properties of the system and is often more useful than a closed-form function $u(x, y, t)$. Our method carries over directly; it is simply a matter of curating a dataset mapping numerical inputs to PDEs.
2) **Enable "Soft" Constraints:** Since we decode governing laws using an LLM, it may be possible to leverage existing methods such as instruction tuning or RLHF to enable specifying "soft" stylistic constraints (e.g. the form of an equation, order of a PDE, etc.), which are much harder to encode into classical approaches like genetic search or linear regression.
3) **Enable Iterative Search:** One key limitation our model suffers from is that it is zero-shot; that is, it is forced to generate a prediction in a single attempt. It would be highly beneficial to enable iterative search to refine predictions. One possibility could be to pass the numeric form of the decoded law back into the foundation model as an error term (thus closing the iteration loop).

VII. Conclusion

In this work, we propose that foundation models pretrained on a large corpus of image or physics data learn latent structures which provide a strong prior for extracting governing laws from numerical data. We study this through the lens of symbolic regression, using a downstream LLM-based decoding D_ϕ to map latent representations onto a symbolic basis. To include a physics foundation model in our study, we introduce *decoder compression* to assimilate its internal latent features.

In so doing, we push the frontier in symbolic regression, obtaining state-of-the-art performance in recovering symbolic equations both in-distribution and even against challenging real-world datasets with no closed-form solutions. We speculate that our results could join a growing body of work suggesting that foundation models learn implicit *world models* in their latent space [10], [11], [14], [21], [27], [32], which we argue may be the driving force behind our strong performance. Additionally, we also demonstrate that decoder compression is an effective means of deploying decoder-only foundation models in encoder-like settings, sometimes even outperforming native encoders in the process.

Finally, we lay out a roadmap for future work to build on our findings, which we hope could have great promise for accelerating scientific discovery and engineering use-cases, as well as shedding light on the underlying behaviour of foundation models.

VIII. Acknowledgement

Generative AI was used in two contexts:

1) **Code editing:** Cursor (https://cursor.com/agents) was used to write code to enable rapid iteration and testing, as well as conduct quick Q&A about the codebase. Example prompts included "refactor X to do Y", "What is the input shape expected by our encoders?", etc. Code was human-audited in GitHub pull requests among other stages.
2) **Drafting Report Sections:** Gemini (https://gemini.google.com) was used to draft sections of the report by converting bullet-point plans into paragraphs, as well as to assist with LaTeXformatting (e.g. "refactor this table to look like X"). All content was thoroughly human-reviewed and edited to align with the authors' intent as well as to ensure no false or misleading content was present.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

[2] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael G. Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, 2023.

[3] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, A. P. Sarath Chandar, Franziska Meier, Yann LeCun, Michael G. Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *ArXiv*, abs/2506.09985, 2025.

[4] Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurélien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. *ArXiv*, abs/2106.06427, 2021.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Thomas Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *ArXiv*, abs/2005.14165, 2020.

[6] Bogdan Burlacu, Gabriel Kronberger, and Michael Kommenda. Operon c++: An efficient genetic programming framework for symbolic regression. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, GECCO '20, page 1562–1570, New York, NY, USA, 2020. Association for Computing Machinery.

[7] Lawrence Cayton. Algorithms for manifold learning. 2005.

[8] Jungyeon Cho and Alex Lazarian. Compressible magnetohydrodynamic turbulence : mode coupling , scaling relations , anisotropy , new regime and astrophysical implications. 2008.

[9] Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl. *ArXiv*, abs/2305.01582, 2023.

[10] Jake Bruce et al. Genie: Generative interactive environments. *ArXiv*, abs/2402.15391, 2024.

[11] Rio Alexa Fear, Payel Mukhopadhyay, Michael McCabe, Alberto Bietti, and Miles Cranmer. Physics steering: Causal control of cross-domain concepts in a physics foundation model. *ArXiv*, abs/2511.20798, 2025.

[12] Charles Fefferman, Sanjoy K. Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *arXiv: Statistics Theory*, 2013.

[13] Peter Gray and Stephen K. Scott. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: Oscillations and instabilities in the system a + 2b $\rightarrow$ 3b; b $\rightarrow$ c. *Chemical Engineering Science*, 39:1087–1097, 1984.

[14] David R Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Neural Information Processing Systems*, 2018.

[15] Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *ArXiv*, abs/2405.19101, 2024.

[16] J. Edward Hu, Ye-Long Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, She-An Wang, Lu Wang, and Wei-Zhu Chen. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685, 2021.

[17] John M. Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Andy Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589, 2021.

[18] Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. *ArXiv*, abs/2204.10532, 2022.

[19] Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jangho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022.

[20] Re-Mi Lam, Álvaro Sánchez-González, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman V. Ravuri, Timo Ewalds, Zach Eaton-Rosen, Wei-Hua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter W. Battaglia. Learning skillful medium-range global weather forecasting. *Science*, 382:1416 – 1421, 2023.

[21] Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Vi'egas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. *ArXiv*, abs/2210.13382, 2022.

[22] Ming Li and Paul M. B. Vitányi. An introduction to kolmogorov complexity and its applications. In *Texts in Computer Science*, 2019.

[23] Yuhui Li, Fang-Yun Wei, Chao Zhang, and Hong-Yang Zhang. Eagle-3: Scaling up inference acceleration of large language models via training-time test. *ArXiv*, abs/2503.01840, 2025.

[24] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *ArXiv*, abs/2304.08485, 2023.

[25] Ziyang Ma et al. Slam-llm: A modular, open-source multimodal large language model framework and best practice for speech, language, audio and music processing. *IEEE Journal of Selected Topics in Signal Processing*, 20:63–76, 2026.

[26] David J. C. MacKay. *Information theory, inference, and learning algorithms*. Cambridge University Press, 2003.

[27] Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworldmodel: Stable end-to-end joint-embedding predictive architecture from pixels. *ArXiv*, abs/2603.19312, 2026.

[28] Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Géraud Krawezik, François Lanusse, Mariel Pettee, Tiberiu Teşileanu, Kyunghyun Cho, and Shirley Ho. Multiple physics pretraining for physical surrogate models. *ArXiv*, abs/2310.02994, 2023.

[29] Michael McCabe, Payel Mukhopadhyay, Tanya Marwah, Bruno Régaldo-Saint Blancard, François Rozet, Cristiana-Diana Diaconu, Lucas Meyer, Kaze W. K. Wong, Hadi Sotoudeh, Alberto Bietti, Irina Espejo, Rio Alexa Fear, Siavash Golkar, Tom Hehir, Keiya Hirashima, Géraud Krawezik, François Lanusse, Rudy Morel, Ruben Ohana, Liam Holden Parker, Mariel Pettee, Jeff Shen, Kyunghyun Cho, Miles Cranmer, and Shirley Ho. Walrus: A cross-domain foundation model for continuum dynamics. *ArXiv*, abs/2511.15684, 2025.

[30] Neel Nanda, Lawrence Chan, Tom Lieberum, Jessie J. Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. *ArXiv*, abs/2301.05217, 2023.

[31] Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Julia Agocs, Miguel Beneitez, Marsha Berger, Blakesley Burkhart, Keaton J. Burns, Stuart B. Dalziel, Drummond B. Fielding, Daniel Fortunato, Jared A. Goldberg, Keiya Hirashima, Yan-Fei Jiang, Rich R. Kerswell, Suryanarayana Maddu, Jonah M. Miller, Payel Mukhopadhyay, Stefan S. Nixon, Jeff Shen, Romain Watteaux, Bruno Régaldo-Saint Blancard, François Rozet, Liam Holden Parker, Miles Cranmer, and Shirley Ho. The well: a large-scale collection of diverse physics simulations for machine learning. *ArXiv*, abs/2412.00568, 2024.

[32] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Thomas Henighan, Benjamin Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, John Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom B. Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. *ArXiv*, abs/2209.11895, 2022.

[33] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russ Howes, Po-Yao (Bernie) Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *ArXiv*, abs/2304.07193, 2023.

[34] Brenden K. Petersen and Mikel Landajuela. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *arXiv: Learning*, 2019.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *ArXiv*, abs/2103.00020, 2021.

[36] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.

[37] Michael Douglas Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324:81 – 85, 2009.

[38] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[39] R.J. Solomonoff. A formal theory of inductive inference. part i. *Information and Control*, 7(1):1–22, 1964.