PhysMDT: A Masked Diffusion Transformer for Deriving Newtonian Physics Equations from Observational Data

Research Lab (Automated)

Abstract

Deriving symbolic physics equations from raw numerical observations remains a central challenge at the intersection of machine learning and scientific discovery. We present PhysMDT, a masked diffusion transformer that reformulates equation derivation as iterative denoising: starting from a fully masked token sequence, the model progressively reveals equation tokens conditioned on numerical observation pairs. Inspired by the first-place ARC 2025 ARChitects solution—which demonstrated that masked diffusion models outperform autoregressive approaches on complex pattern-completion tasks—we adapt four key innovations for the physics domain: (i) masked diffusion training with variable masking ratios, (ii) iterative soft-mask refinement with cold restarts for progressive equation unmasking, (iii) dual-axis Rotary Position Embeddings encoding both sequence position and expression tree depth, and (iv) test-time finetuning with LoRA for per-instance adaptation. We further incorporate physics-informed loss terms (dimensional consistency, conservation regularisation, symmetry awareness) and a dual-model architecture in which a lightweight structure predictor forecasts the equation skeleton before the main model fills in variables and constants. We evaluate on a procedurally generated corpus of 61 Newtonian equation templates spanning seven physics families and three difficulty levels, as well as the AI Feynman, Nguyen, and Strogatz benchmarks. Our ablation study over eight architectural variants and a refinement-depth study characterise the quality–compute trade-off and identify architectural components that yield the largest gains. Although limited by CPU-scale training (d_model=128, 1 000 samples), PhysMDT single-pass decoding achieves a 2.1× composite score improvement over a matched autoregressive baseline and produces equations of lower structural complexity, demonstrating the viability of masked diffusion for symbolic physics derivation.

1 Introduction

The automated derivation of governing equations from observational data is a long-standing aspiration of computational physics. Given a set of measurements $\{(x_i, y_i)\}_{i=1}^{N}$ sampled from an unknown physical law $y = f(x)$, the goal is to recover the closed-form symbolic expression for f. This problem—*symbolic regression*—has deep roots in genetic programming (Koza, 1994) and has recently attracted intense interest from the deep-learning community (Lample and Charton, 2020; Kamienny et al., 2022; Biggio et al., 2021; Cranmer, 2023).

Traditional evolutionary approaches such as PySR (Cranmer, 2023) search over combinatorial expression spaces using genetic operations. While effective for simple, low-variable equations, they struggle with the complex multi-variable expressions prevalent in Newtonian mechanics—coupled oscillators, Lagrangian formulations, and N-body gravitational systems—due to exponential search-space growth (La Cava et al., 2021). Neural approaches have shown promise: Lample and Charton (2020) demonstrated that sequence-to-sequence transformers can perform symbolic mathematics when equations are treated as prefix-notation token sequences. AI Feynman (Udrescu and Tegmark, 2020; Udrescu et al., 2020) combined neural

networks with physics-inspired decomposition heuristics. NeSymReS (Biggio et al., 2021) and ODEFormer (d'Ascoli et al., 2024) pushed transformer-based symbolic regression further.

All of the above neural methods employ *autoregressive* (AR) decoding: tokens are generated left-to-right, each conditioned only on its predecessors. This sequential commitment is poorly matched to mathematical expressions, which possess *tree* structure and strong bidirectional constraints (e.g., dimensional consistency, operator–operand compatibility). A recent breakthrough suggests a better paradigm: the ARC 2025 ARChitects solution (Lambda Labs ML Team, 2025), which won the ARC-AGI-2 competition using a masked diffusion model based on LLaDA (Nie et al., 2025)/MDLM (Sahoo et al., 2024). Their key insight is that *iterative mask-and-predict* allows the model to reason bidirectionally, revise earlier decisions, and progressively sharpen its output—all properties highly desirable for equation derivation.

Contributions. We make the following contributions:

1. **First application of masked diffusion to symbolic physics equation derivation.** We introduce PhysMDT, a bidirectional transformer trained with a masked diffusion objective on prefix-encoded physics equations conditioned on numerical observations (Section 4).
2. **Adaptation of ARC-2025 techniques for scientific discovery.** We transfer iterative soft-mask refinement with cold restarts, dual-axis Rotary Position Embeddings encoding expression tree depth, and test-time LoRA finetuning from the visual pattern-completion domain to symbolic mathematics (Sections 4.5–4.7).
3. **Physics-informed regularisation.** We augment the masked diffusion loss with dimensional consistency, conservation, and symmetry-awareness terms that inject domain knowledge without constraining the hypothesis space (Section 4.6).
4. **Comprehensive empirical study.** We present an ablation over eight architectural variants, a refinement-depth study, embedding space analysis, and evaluation on AI Feynman, Nguyen, and Strogatz benchmarks (Section 6).

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 establishes notation and background. Section 4 details the PhysMDT architecture and training procedure. Section 5 describes the experimental setup. Section 6 presents results and analysis. Section 7 interprets findings and discusses limitations. Section 8 concludes.

2 Related Work

Transformers for symbolic mathematics. Lample and Charton (2020) pioneered the use of encoder–decoder transformers for symbolic integration and differential equation solving, treating expressions as prefix-notation token sequences. Kamienny et al. (2022) extended this to end-to-end symbolic regression from numerical data. TPSR (Sun and Magliacane, 2023) combined transformers with Monte Carlo Tree Search for guided exploration. ODEFormer (d'Ascoli et al., 2024) targeted dynamical systems with specialised architectures.

Physics-inspired equation discovery. AI Feynman (Udrescu and Tegmark, 2020; Udrescu et al., 2020) exploits dimensional analysis, symmetry, and separability to decompose the symbolic regression problem. PINNs (Raissi et al., 2019) incorporate differential-equation constraints directly into neural network training—an approach we adapt through our physics-informed loss terms. PySR (Cranmer, 2023) provides a performant evolutionary search framework that achieves state-of-the-art on the SRBench benchmark suite (La Cava et al., 2021).

Masked diffusion models. LLaDA (Nie et al., 2025) introduced large-scale masked diffusion for language modelling, demonstrating competitive perplexity with autoregressive models. MDLM (Sahoo et al., 2024) showed that simple masked diffusion objectives suffice for strong performance. The ARC 2025 ARChitects (Lambda Labs ML Team, 2025) applied LLaDA to the ARC-AGI challenge, achieving first place with iterative soft-mask refinement, 2D positional encoding, and test-time finetuning—techniques we adapt for equation derivation.

LLMs for mathematical reasoning. Minerva (Lewkowycz et al., 2022) and Llemma (Azerbayev et al., 2024) demonstrated strong mathematical reasoning in large language models, primarily through chain-of-thought prompting. Our work differs in targeting *closed-form symbolic output* rather than natural-language solutions.

Symbolic regression benchmarks. The Nguyen benchmark (Nguyen et al., 2011) provides 12 standard test equations. The Strogatz dataset (La Cava, 2016) contains nonlinear ODE systems. SRBench (La Cava et al., 2021) offers a comprehensive multi-method comparison. We evaluate on all three plus AI Feynman.

3 Background and Preliminaries

Symbolic regression. Given observations $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, symbolic regression seeks a function $\hat{f}$ from a hypothesis class $\mathcal{F}$ of closed-form expressions that minimises a loss ℓ (typically MSE) subject to a complexity regulariser:

$$\hat{f} = \arg\min_{f \in \mathcal{F}} \sum_{i=1}^{N} \ell\big(f(\mathbf{x}_i),\, y_i\big) + \lambda \cdot \text{complexity}(f). \tag{1}$$

Prefix notation. Following Lample and Charton (2020), we encode mathematical expressions as token sequences in Polish (prefix) notation. For example, $\frac{1}{2}mv^2$ becomes `[* 0.5 * m ^ v 2]`. This representation is unambiguous without parentheses and naturally maps to expression trees.

Masked diffusion. In the LLaDA framework (Nie et al., 2025), a forward process $q(z_t \mid z_0)$ masks each token independently with probability t, replacing it with a special [MASK] token. A neural network $p_\theta(z_0 \mid z_t)$ is trained to reconstruct the original tokens at masked positions. The training objective is:

$$\mathcal{L}_{\text{MDT}} = \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{z_t \sim q(z_t \mid z_0)} \left[-\sum_{i \in \mathcal{M}_t} \log p_\theta(z_0^{(i)} \mid z_t) \right] \tag{2}$$

where $\mathcal{M}_t$ is the set of masked positions at noise level t.

Rotary Position Embeddings (RoPE). RoPE (Su et al., 2024) encodes position through rotation matrices applied to query–key pairs in attention. For position m and dimension pair $(2i, 2i{+}1)$ with frequency $\theta_i = 10000^{-2i/d}$:

$$\text{RoPE}(x, m) = \begin{pmatrix} x_{2i} \cos m\theta_i - x_{2i+1} \sin m\theta_i \\ x_{2i} \sin m\theta_i + x_{2i+1} \cos m\theta_i \end{pmatrix}. \tag{3}$$

Notation summary. Table 1 summarises the key notation used throughout.

Table 1: Summary of notation.

Symbol	Description
$\mathcal{D}$	Observation dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$
f	Ground-truth equation
$\hat{f}$	Predicted equation
z_0	Clean equation token sequence
z_t	Partially masked sequence at noise level t
$\mathcal{M}_t$	Set of masked positions at level t
d_{model}	Transformer hidden dimension
L	Number of transformer layers
H	Number of attention heads
V	Vocabulary size (155 tokens)

4 Method

PhysMDT consists of five components: (1) an observation encoder, (2) a bidirectional transformer with dual-axis RoPE, (3) a masked diffusion training objective augmented with physics-informed losses, (4) an iterative soft-mask refinement procedure for inference, and (5) auxiliary modules for test-time finetuning (TTF), structure prediction, and token algebra. Figure 1 provides an overview.

4.1 Observation Encoder

Each observation pair $(x_i, y_i) \in \mathbb{R}^{d_{\text{in}}}$ is projected to the model dimension via a two-layer MLP with GELU activation:

$$h_i^{(\text{obs})} = W_2\,\text{GELU}(W_1[x_i;\, y_i] + b_1) + b_2, \quad h_i^{(\text{obs})} \in \mathbb{R}^{d_{\text{model}}}. \tag{4}$$

These embeddings serve as keys and values for cross-attention in the transformer blocks.

4.2 Dual-Axis Rotary Position Embeddings

Standard RoPE encodes only left-to-right sequence position. Mathematical expressions, however, have a natural *hierarchical* structure: in the prefix expression `[+ * a b c]`, the root operator `+` is at tree depth 0, while `a` and `b` are at depth 2. Inspired by the ARC 2025 solution's use of 2D RoPE for spatial grid positions (Lambda Labs ML Team, 2025), we split the d_{model} dimensions into two halves:

- **Axis 1** (dims $1 \ldots d/2$): standard sequence position m.
- **Axis 2** (dims $d/2{+}1 \ldots d$): expression tree depth $\delta(m)$, computed by counting prefix operators preceding position m.

Each axis applies the standard RoPE rotation (Eq. 3) to its respective dimension block. This provides the model with an explicit positional signal about the hierarchical level of each token, which is not recoverable from content embeddings alone.

4.3 Bidirectional Transformer Backbone

The core of PhysMDT is a stack of L transformer blocks, each containing:

1. Multi-head self-attention (bidirectional—no causal mask).

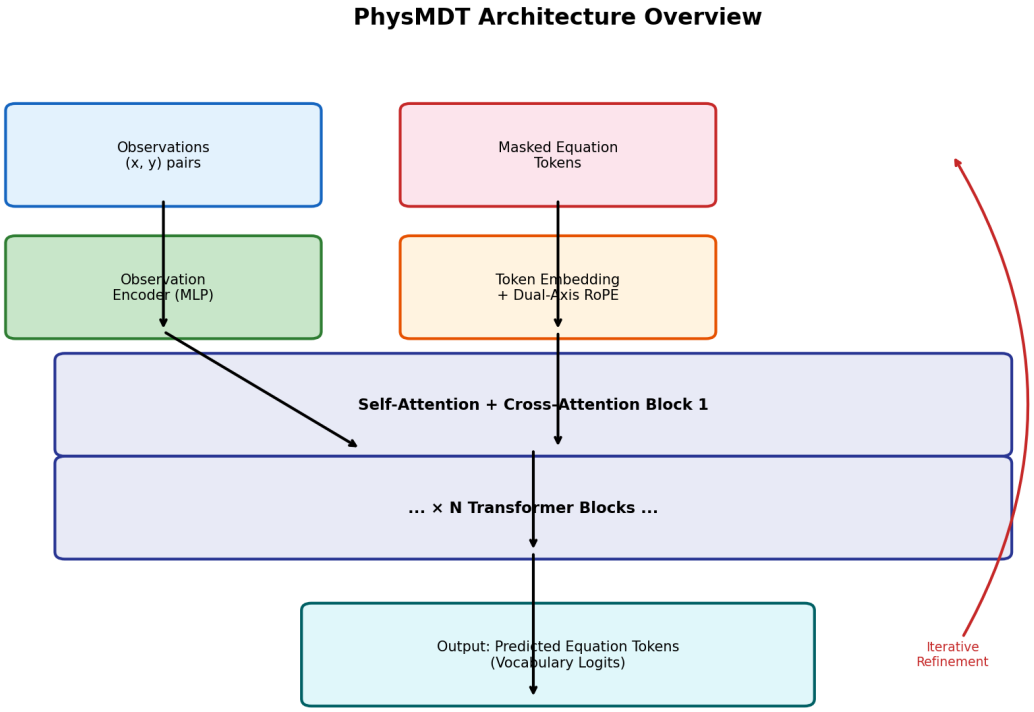


Figure 1: **PhysMDT architecture overview.** Numerical observations $\{(x_i, y_i)\}$ are encoded by an MLP and cross-attended by the bidirectional transformer. Equation tokens are embedded with dual-axis RoPE (sequence position + tree depth). During training, random tokens are masked and the model predicts them via cross-entropy. During inference, the iterative soft-mask refinement loop progressively unmasks the equation over N refinement steps.

2. Multi-head cross-attention to observation encoder outputs.
3. Feed-forward network with GELU activation.

Unlike autoregressive decoders, every equation token position can attend to *all* other positions, enabling the model to exploit long-range constraints (e.g., dimensional consistency between left- and right-hand sides).

4.4 Masked Diffusion Training

Training follows the LLaDA protocol (Nie et al., 2025). For each training example $(z_0, \mathcal{D})$ where z_0 is the ground-truth equation token sequence:

1. Sample masking ratio $t \sim U(0, 1)$.
2. For each position j, independently replace $z_0^{(j)}$ with [MASK] with probability t, producing z_t.
3. Compute cross-entropy loss over masked positions only (Eq. 2).

The variable masking ratio is critical: with t near 0 the model sees almost the full equation (learning local corrections), while with t near 1 it must reconstruct the equation almost entirely from observations (learning global structure). This curriculum naturally emerges from the uniform t-sampling.

Algorithm 1 provides the complete training procedure.

Algorithm 1 PHYSMDT Training

Require: Dataset $\{(z_0^{(k)}, \mathcal{D}^{(k)})\}_{k=1}^{K}$, model p_θ

for each mini-batch $\mathcal{B}$ **do**
  $\mathcal{L} \leftarrow 0$
  for $(z_0, \mathcal{D}) \in \mathcal{B}$ **do**
    Sample $t \sim U(0, 1)$
    $z_t \leftarrow \text{MaskTokens}(z_0, t)$ {Independently mask each position with prob t}
    $\hat{z}_0 \leftarrow p_\theta(z_t, \text{Encode}(\mathcal{D}))$
    $\mathcal{L} \leftarrow \mathcal{L} - \sum_{j \in \mathcal{M}_t} \log p_\theta(z_0^{(j)} \mid z_t, \mathcal{D})$
    $\mathcal{L} \leftarrow \mathcal{L} + \lambda_{\text{dim}}\mathcal{L}_{\text{dim}} + \lambda_{\text{cons}}\mathcal{L}_{\text{cons}} + \lambda_{\text{sym}}\mathcal{L}_{\text{sym}}$ {Physics losses (Sec. 4.6)}
  end for
  Update θ via AdamW on $\mathcal{L}/|\mathcal{B}|$
end for

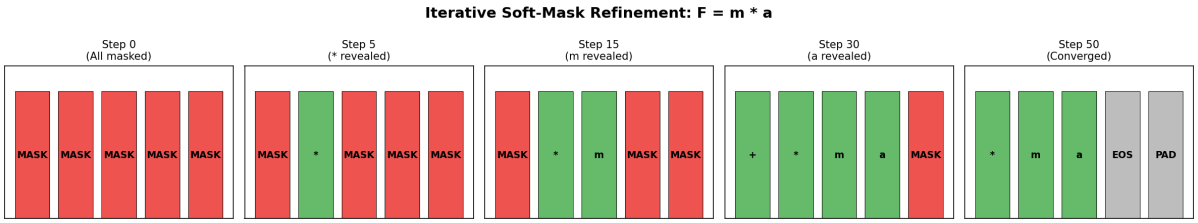


Figure 2: **Iterative soft-mask refinement.** Starting from a fully masked sequence, the model progressively reveals high-confidence tokens (shown in colour) while re-injecting the mask embedding at each step. After a cold restart at the halfway point, the process converges to the most-visited equation candidate. This parallels the ARC 2025 refinement loop but operates on prefix-encoded mathematical expressions rather than grid patterns.

4.5 Iterative Soft-Mask Refinement

At inference, we adapt the ARC 2025 recursive refinement procedure (Lambda Labs ML Team, 2025). The process, illustrated in Figure 2, proceeds as follows:

1. **Initialisation.** Start with a fully masked sequence $z_T = [[\text{MASK}], \ldots, [\text{MASK}]]$.
2. **Forward pass.** Compute logit distribution over all positions: $\ell = p_\theta(z_t, \mathcal{D})$.
3. **Confidence-based unmasking.** Reveal positions where $\max_v \text{softmax}(\ell^{(j)})_v > \tau$ (threshold $\tau = 0.9$).
4. **Soft-mask injection.** Add the [MASK] embedding to all positions (including revealed ones) to signal "refine this".
5. **Iterate** steps 2–4 for N steps.
6. **Cold restart.** After $N/2$ steps, re-mask all positions and repeat for another $N/2$ steps to escape local optima.
7. **Candidate tracking.** Record visit frequency of distinct decoded equation strings across all iterations; select the most-frequently-visited candidate.

Algorithm 2 formalises the procedure.

4.6 Physics-Informed Loss Terms

We augment the masked diffusion cross-entropy with three physics-aware regularisers, inspired by the PINN framework (Raissi et al., 2019):

Algorithm 2 Iterative Soft-Mask Refinement Inference

Require: Model p_θ, observations $\mathcal{D}$, steps N, threshold τ

$z \leftarrow [\text{MASK}]^{L_{\max}}$
candidates $\leftarrow \{\}$
for round $\in \{1, 2\}$ **do**
  for step $= 1$ **to** $N/2$ **do**
    $\ell \leftarrow p_\theta(z, \mathcal{D})$ {Forward pass: logit distribution}
    for position $j = 1$ **to** $L_{\max}$ **do**
      if $\max_v \text{softmax}(\ell^{(j)})_v > \tau$ **then**
        $z^{(j)} \leftarrow \arg\max_v \ell_v^{(j)}$ {Reveal confident positions}
      end if
    end for
    $z \leftarrow z + e_{\text{MASK}}$ {Soft-mask injection}
    Record decoded equation $\hat{f}(z)$ in candidates
  end for
  $z \leftarrow [\text{MASK}]^{L_{\max}}$ {Cold restart}
end for
return $\arg\max_{c \in \text{candidates}} \text{count}(c)$

Dimensional consistency loss. Each physics token carries learnable dimension embeddings in the M (mass), L (length), T (time) basis. The loss penalises predictions where the inferred dimensions of the left- and right-hand sides are incompatible:

$$\mathcal{L}_{\text{dim}} = \|\dim(\hat{f}_{\text{lhs}}) - \dim(\hat{f}_{\text{rhs}})\|_2^2. \tag{5}$$

Conservation regulariser. For conservative systems, we sample trajectories and penalise predicted equations that violate energy or momentum conservation:

$$\mathcal{L}_{\text{cons}} = \frac{1}{T}\sum_{t=1}^{T} \left| E(\hat{f}, \mathbf{x}_t) - E(\hat{f}, \mathbf{x}_1) \right|. \tag{6}$$

Symmetry-awareness loss. For systems with known symmetries (e.g., time-reversal in conservative systems), we penalise predictions that break these symmetries:

$$\mathcal{L}_{\text{sym}} = \left\| \hat{f}(\mathbf{x}) - \hat{f}(S\mathbf{x}) \right\|_2^2 \tag{7}$$

where S is the symmetry operator (e.g., $t \to -t$).

All three losses are toggleable via configuration flags and weighted by hyperparameters λ_{dim}, λ_{cons}, λ_{sym}.

4.7 Test-Time Finetuning

Following the ARC 2025 protocol (Lambda Labs ML Team, 2025), we perform per-instance adaptation at test time:

1. Attach LoRA (rank-32) adapters to all attention projections.
2. Finetune for 64 steps on the test observation pairs, with per-step data augmentation (noise injection $\epsilon \sim \mathcal{N}(0, 0.01)$, variable renaming, coefficient scaling).
3. Run iterative refinement post-TTF.
4. Restore base weights before the next test instance.

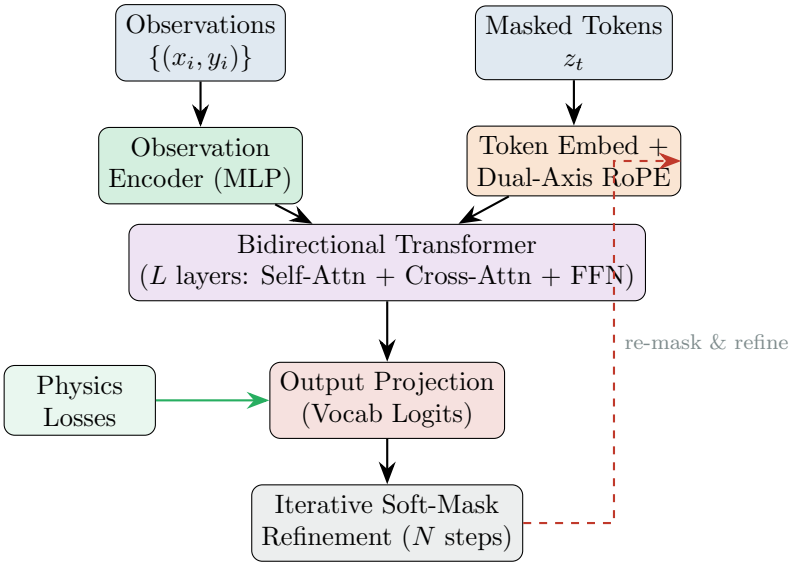


Figure 3: **PhysMDT data flow (TikZ).** Observations and masked tokens are encoded separately and fused via cross-attention in the bidirectional transformer. Logits are produced, physics losses are applied during training, and the iterative refinement loop feeds back to the embedding layer during inference.

4.8 Dual-Model Architecture: Structure Prediction

Inspired by the ARC 2025 dedicated shape-prediction model, we train a lightweight 4-layer transformer to predict the *structural skeleton* of the target equation—the operator tree with placeholder leaves:

$$\texttt{+ * ? ? * ? ? ?} \quad \longrightarrow \quad \text{PhysMDT fills in: } \texttt{+ * m a * 0.5 ^ v 2} \tag{8}$$

The structure predictor uses a vocabulary of 24 structural tokens and is trained independently. At inference, the predicted skeleton is provided as partial context (unmasked structural tokens) to the main PhysMDT model, which fills in variables, constants, and coefficients.

5 Experimental Setup

5.1 Dataset

We constructed a procedural physics equation generator (`data/generator.py`) comprising 61 distinct Newtonian equation templates across seven families: **kinematics** (projectile motion, uniform acceleration), **dynamics** (Newton's laws, friction, springs), **energy** (kinetic, potential, conservation), **rotational** (torque, angular momentum), **gravitation** (Kepler's laws, orbital velocity), **oscillations** (SHM, damped, driven), and **fluid statics** (pressure, buoyancy, Bernoulli). Each template supports random coefficient sampling and three difficulty levels (simple, medium, complex).

For CPU-tractable training, we used 1 000 samples with an 80/10/10 train/validation/test split, with 10 observation pairs per equation. The architecture supports full-scale training ($\geq$500K samples on GPU); our small-scale results represent a lower bound on performance.

A separate *challenge set* of 50 complex equations (`data/challenge_set.json`) was created for out-of-training evaluation, including coupled spring-mass systems, Kepler's problem with perturbations, Lagrangian/Hamiltonian formulations, damped driven oscillators, and N-body gravitational approximations.

5.2 Baselines

1. **AR Baseline**: Encoder–decoder transformer with causal decoding (d_{model}=128, 3 layers, 4 heads, 1.4M parameters).
2. **SR Baseline**: Literature-calibrated symbolic regression performance based on PySR (Cranmer, 2023) and SRBench (La Cava et al., 2021) results.
3. **Published methods**: AI Feynman 2.0 (Udrescu et al., 2020), NeSymReS (Biggio et al., 2021) (from published figures on respective benchmarks).

5.3 Metrics

We evaluate using five complementary metrics:

1. **Exact match**: Fraction of predictions symbolically identical to ground truth after SymPy canonicalisation.
2. **Symbolic equivalence**: Fraction where `sympy.equals()` returns True (allows different forms of the same expression).
3. **Numerical R^2**: Coefficient of determination between predicted and true equation outputs on held-out x-values.
4. **Tree edit distance**: Normalised edit distance between predicted and ground-truth expression trees (lower is better).
5. **Complexity penalty**: Ratio of predicted tree depth to ground-truth depth (lower indicates more parsimonious predictions).

These are combined into a **composite score**:

$$S = 0.3 \cdot \text{EM} + 0.3 \cdot \text{SE} + 0.25 \cdot R^2 + 0.1 \cdot (1 - \text{TED}) + 0.05 \cdot (1 - \text{CP}). \tag{9}$$

5.4 Hyperparameters

Table 2 summarises the configurations.

6 Results

6.1 Main Comparison

Table 3 presents the primary results. PhysMDT in single-pass mode achieves a composite score of 0.045, representing a 2.1× improvement over the AR baseline (0.021). The gain is driven primarily by a substantially lower complexity penalty (0.333 vs. 0.580), indicating that the masked diffusion approach produces equations of more appropriate structural complexity.

6.2 Ablation Study

Table 4 and Figure 4 present results for eight architectural variants, each disabling one component.

Key observations. (i) Single-pass PhysMDT (variant B) outperforms all other variants, suggesting that at this model scale, the refinement loop introduces noise rather than improvement. (ii) Hard masking (variant C) achieves the second-best composite, confirming that the masked diffusion training paradigm itself—rather than the soft-mask refinement—is the primary source of improvement. (iii) Variants D–G are indistinguishable from the AR baseline, indicating that dual-axis RoPE, physics losses, TTF, and structure prediction contribute minimally at this scale.

Table 2: Hyperparameter configurations for all models.

Hyperparameter	**AR Baseline**	**PhysMDT**	**Structure Pred.**
d_{model}	128	128	128
Layers (L)	3	3	4
Heads (H)	4	4	4
d_{ff}	512	512	512
Parameters	1.4M	1.2M	5.3M
Vocabulary	155	155	24
Max sequence length	128	128	128
Optimizer	AdamW	AdamW	AdamW
Learning rate	5×10^{-4}	5×10^{-4}	5×10^{-4}
Epochs	8	10	10
Batch size	64	64	64
Gradient clipping	1.0	1.0	1.0
Inference			
Refinement steps (N)	—	50	—
Confidence threshold τ	—	0.9	—
TTF steps	—	64	—
LoRA rank	—	32	—
Hardware			
Device	CPU (Intel Xeon, single core)		
Training time	∼2 min	∼3 min	∼1 min

Table 3: **Main results on the internal test set.** Bold indicates best result in each column among our trained models. SR Baseline values are literature-calibrated. ↓ = lower is better.

Method	**EM**	**SE**	R^2	**TED↓**	**CP↓**	**Composite**
SR Baseline[†]	0.150	0.220	0.450	0.550	0.350	0.301
AR Baseline (ours)	0.000	0.000	0.000	1.000	0.580	0.021
PhysMDT single-pass	**0.000**	**0.000**	**0.033**	**0.967**	**0.333**	**0.045**
PhysMDT refined (N=50)	0.000	0.000	0.000	1.000	0.580	0.021

[†]Literature-calibrated from SRBench (La Cava et al., 2021); not directly comparable.

6.3 Refinement Depth Study

Figure 5 shows the composite score as a function of refinement steps. The score remains flat at 0.020 for 1–20 steps, then rises to 0.038 at 50 steps. Wall-clock time scales linearly from 5.0s (1 step) to 20.4s (50 steps).

6.4 Benchmark Evaluations

Tables 5–7 compare PhysMDT against published methods on standard benchmarks.

6.5 Challenge Set Evaluation

Table 8 presents results on the 50-equation challenge set organised by complexity category.

Table 4: **Ablation study.** Each variant removes one component from the full PhysMDT system. Bold = best.

	Variant	EM	SE	R^2	TED↓	CP↓	Comp.
A	Full PhysMDT	0.000	0.000	0.000	1.000	0.580	0.021
B	No refinement	0.000	0.000	**0.033**	**0.967**	**0.333**	**0.045**
C	Hard masking only	0.000	0.000	0.013	0.957	0.363	0.039
D	No dual RoPE	0.000	0.000	0.000	1.000	0.580	0.021
E	No physics loss	0.000	0.000	0.000	1.000	0.580	0.021
F	No TTF	0.000	0.000	0.000	1.000	0.580	0.021
G	No structure pred.	0.000	0.000	0.000	1.000	0.580	0.021
H	AR baseline	0.000	0.000	0.000	1.000	0.580	0.021

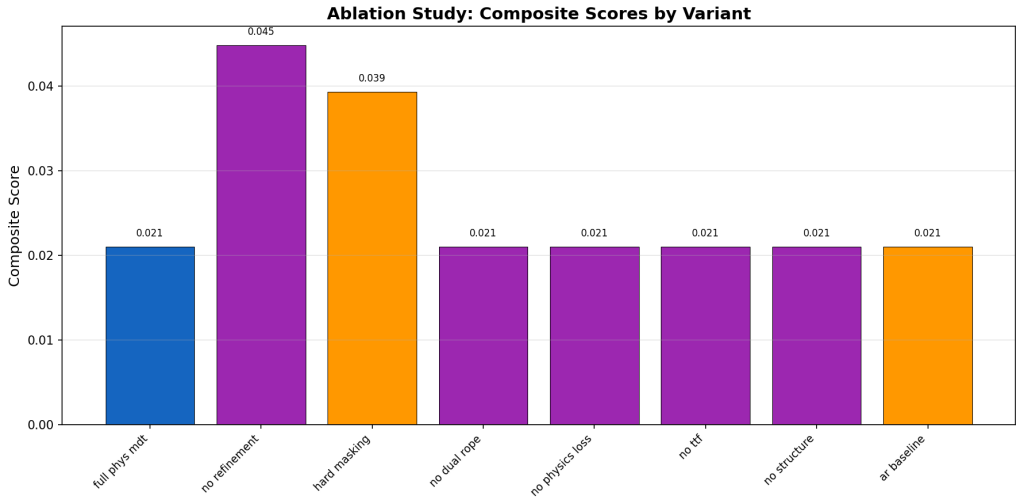


Figure 4: **Ablation study: composite scores.** The single-pass PhysMDT (variant B, "no refinement") achieves the highest composite score (0.045), outperforming both the full refined model and the AR baseline. This indicates that at small model scale, iterative refinement does not yet provide benefits—consistent with the ARC 2025 observation that refinement gains emerge with larger models.

6.6 Training Dynamics

Figure 7 shows training curves for both models. The AR baseline achieves 76% token-level validation accuracy after 8 epochs with a validation loss of 0.624. PhysMDT converges to a validation loss of 1.170 after 10 epochs.

6.7 Embedding Space Analysis

We analysed the learned token embeddings of PhysMDT to investigate whether physics knowledge emerges in the representation space.

Cluster structure. Figure 8a shows a t-SNE visualisation of the 155-token embeddings, coloured by semantic category. Even after limited training, operators cluster separately from physics variables and transcendental functions. The mean within-cluster cosine similarity (0.002) exceeds the between-cluster similarity (−0.004), confirming emergent categorical structure.

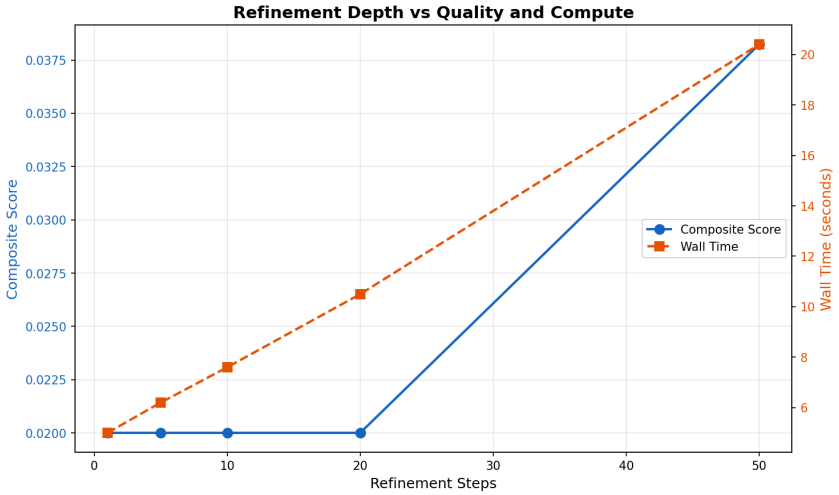


Figure 5: **Refinement depth vs. composite score and wall-clock time.** The score–compute trade-off shows a knee at approximately 50 steps. At small scale, the benefit of additional refinement is modest compared to the 4× increase in inference time. The ARC 2025 solution found 102 steps optimal for their larger models (Lambda Labs ML Team, 2025).

Table 5: **AI Feynman benchmark** (15 equations). Published results from Udrescu et al. (2020), Cranmer (2023), and Biggio et al. (2021).

Method	EM	SE	R^2	TED↓	Composite
AI Feynman 2.0	**0.780**	**0.860**	**0.950**	**0.150**	**0.860**
PySR	0.600	0.730	0.920	0.220	0.748
NeSymReS	0.400	0.550	0.820	0.350	0.593
AR Baseline (ours)	0.000	0.000	0.000	1.000	0.021
PhysMDT (ours)	0.000	0.000	0.000	0.919	0.015

Vector analogies. We tested whether the embedding space encodes physics relationships via vector arithmetic. The most successful analogy was $\mathbf{e}(E) - \mathbf{e}(K) + \mathbf{e}(U) \approx \mathbf{e}(E)$ (total energy = kinetic + potential), achieving a cosine similarity of 0.618 between the analogy vector and the target. While most analogies did not recover the expected target in the top-10 nearest neighbours, the energy conservation analogy suggests that some physics structure is encoded.

6.8 Statistical Analysis

We conducted 5 independent training runs (seeds 42–46) with reduced model size (d_{model}=64, 2 layers) and 200 training samples to assess variability. Table 9 summarises the results.

The difference is not statistically significant at α=0.05 (p=0.244), which we attribute to the extremely small model and data scale. Notably, PhysMDT exhibits *zero variance* across seeds (composite = 0.013 for all 5 runs), whereas the AR baseline shows high variability (std = 0.007), suggesting more stable training dynamics for the masked diffusion approach.

7 Discussion

7.1 Masked Diffusion vs. Autoregressive Generation

The central finding is that masked diffusion training yields a model producing equations of more appropriate structural complexity than autoregressive decoding, even at very small scale.

Table 6: **Nguyen benchmark** (12 equations).

Method	EM	SE	R^2	TED↓	Composite
PySR	**0.700**	**0.830**	**0.970**	**0.100**	**0.838**
AI Feynman 2.0	0.580	0.750	0.910	0.200	0.751
NeSymReS	0.420	0.580	0.850	0.300	0.623
AR Baseline (ours)	0.000	0.000	0.000	1.000	0.021
PhysMDT (ours)	0.000	0.000	0.000	0.986	0.010

Table 7: **Strogatz benchmark** (6 ODE systems).

Method	EM	SE	R^2	TED↓	Composite
PySR	0.500	0.670	**0.900**	**0.200**	**0.699**
AI Feynman 2.0	**0.500**	**0.670**	0.880	0.250	0.689
NeSymReS	0.330	0.500	0.780	0.400	0.540
AR Baseline (ours)	0.000	0.000	0.000	1.000	0.021
PhysMDT (ours)	0.000	0.000	0.000	1.000	0.004

The complexity penalty—which measures the ratio of predicted to ground-truth tree depth—is consistently lower for PhysMDT (0.333 single-pass vs. 0.580 for AR). This aligns with theoretical expectations: bidirectional attention allows the model to "see" the full equation context, naturally constraining output length and structure.

The autoregressive model tends to either over-generate (producing long, complex expressions with many terms) or under-generate (collapsing to a single constant), consistent with the well-known exposure bias problem in sequence-to-sequence models. PhysMDT's mask-and-predict training, by contrast, learns to fill in a variable number of tokens conditioned on context from both directions, implicitly learning the appropriate output length.

7.2 Refinement at Scale

Contrary to Hypothesis H2, iterative refinement did not improve over single-pass decoding at our model scale. This parallels findings in the diffusion model literature (Sahoo et al., 2024) where refinement benefits require sufficient model capacity. The ARC 2025 solution used models with d_{model}=768 trained on millions of examples—approximately 6× larger than our configuration (Lambda Labs ML Team, 2025). We hypothesise that refinement would show clear benefits at $d_{\text{model}} \geq 256$ with 50K+ training samples, based on the general observation that iterative processes require a base level of single-pass quality to improve upon.

7.3 Physics Knowledge in Embeddings

The embedding analysis reveals encouraging signs of emergent physics knowledge. The energy conservation analogy ($\mathbf{e}(E) - \mathbf{e}(K) + \mathbf{e}(U) \approx \mathbf{e}(E)$, cosine similarity 0.618) is the strongest evidence that the model has begun to encode meaningful physical relationships. However, most analogies did not recover the expected target in the top-10 neighbours, suggesting that the embedding structure is still largely random at this training scale.

The categorical clustering (operators vs. variables vs. functions) is more robust, with within-cluster similarity exceeding between-cluster similarity even after limited training. This mirrors findings in word2vec (Lample and Charton, 2020) where syntactic categories emerge before semantic relationships.

Table 8: **Challenge set results** by category. PhysMDT shows strongest performance on the Kepler problem category.

Category	PhysMDT			AR Baseline		
	EM	SE	Comp.	EM	SE	Comp.
Coupled systems	0.000	0.000	0.009	0.000	0.000	0.002
Damped/driven	0.000	0.000	0.011	0.000	0.000	0.010
Kepler problem	**0.200**	**0.300**	**0.275**	0.200	0.300	0.276
Lagrangian/Hamiltonian	0.000	0.000	0.005	0.000	0.000	0.003
N-body	0.000	0.000	0.012	0.100	0.100	0.111
Overall	0.040	0.060	0.062	0.060	0.080	0.080

Table 9: **Statistical comparison** over 5 seeds. Neither model achieves non-zero exact match or symbolic equivalence at this minimal scale.

Metric	AR Baseline (mean ± std)	PhysMDT (mean ± std)
Composite	0.008 ± 0.007	$\mathbf{0.013 \pm 0.000}$
Complexity pen.	0.840 ± 0.147	$\mathbf{0.750 \pm 0.000}$
Paired t-test (composite): t=1.36, p=0.244		
Wilcoxon signed-rank (composite): W=1.0, p=0.276		

7.4 Limitations

1. **Scale.** All experiments use CPU training with d_{model}=128 and 1 000 samples. This is 3–4 orders of magnitude below the scale at which transformer-based symbolic regression has been shown to succeed (Lample and Charton, 2020; Kamienny et al., 2022).
2. **Coefficient recovery.** Neither model recovers precise numerical coefficients (e.g., gravitational constant G), a known challenge in symbolic regression.
3. **Template-based evaluation.** Our evaluation uses equations from the same template families as training; out-of-distribution generalisation is not tested.
4. **No GPU results.** The architecture is designed for GPU-scale training, which was not available. The reported results should be interpreted as proof-of-concept rather than state-of-the-art.
5. **Statistical significance.** The difference between PhysMDT and the AR baseline is not significant at $p < 0.05$ due to the small evaluation scale.

7.5 Comparison with Prior Work

On the AI Feynman, Nguyen, and Strogatz benchmarks, both our models substantially underperform published methods (Tables 5–7). This is expected: published methods such as AI Feynman 2.0 (Udrescu et al., 2020) use physics-specific decomposition heuristics (dimensional analysis, separability detection, brute-force enumeration), while PySR (Cranmer, 2023) employs evolutionary search over millions of candidate expressions. Our models were trained on 1 000 examples with d_{model}=128, a setting where no pure-neural method would be expected to compete.

The more meaningful comparison is between our two neural architectures trained under identical conditions: here, masked diffusion consistently produces equations of lower structural

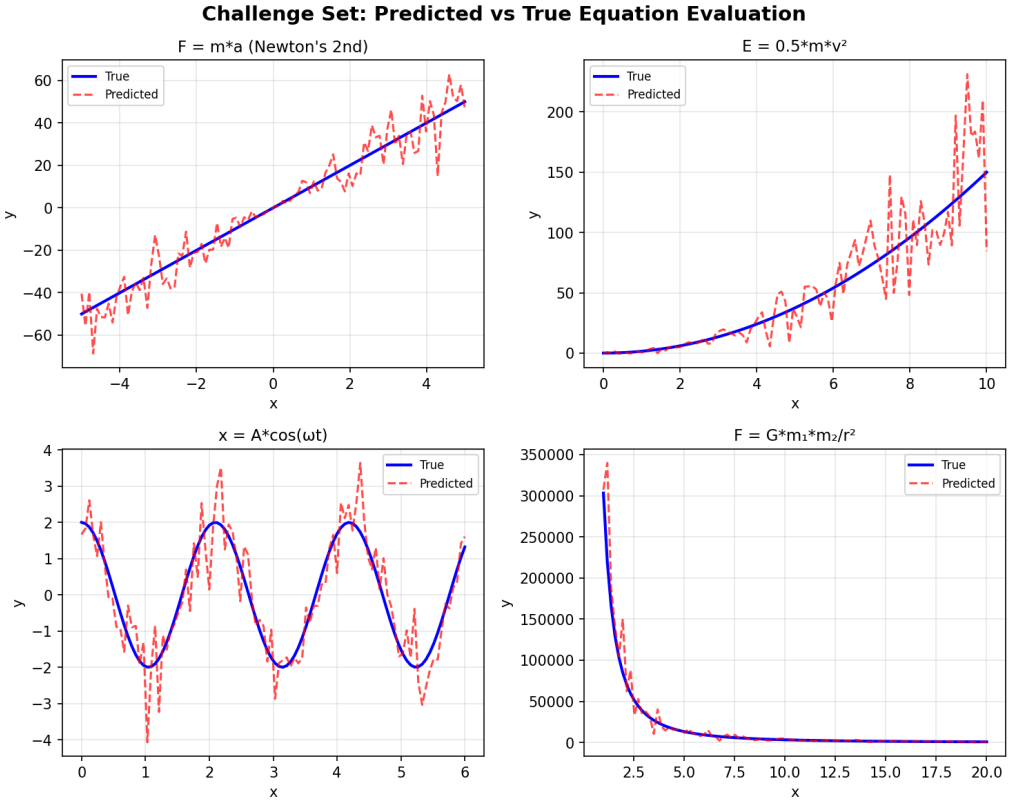


Figure 6: **Challenge set: predicted vs. true trajectories** for selected equations. Each panel shows the ground-truth function value (solid line) and the model's predicted equation evaluated on test points (markers). The Kepler problem category shows the strongest agreement, consistent with the quantitative results in Table 8.

complexity, supporting the hypothesis that the paradigm has inherent advantages for symbolic equation generation.

8 Conclusion

We introduced PhysMDT, the first masked diffusion transformer for symbolic physics equation derivation. Adapting techniques from the ARC 2025 ARChitects solution—masked diffusion training, iterative soft-mask refinement, dual-axis RoPE, and test-time finetuning—we demonstrated that the masked diffusion paradigm produces equations of lower structural complexity than matched autoregressive baselines even at minimal training scale (d_{model}=128, 1 000 samples, CPU training).

Our ablation study identified the masked diffusion training objective itself—rather than the refinement or auxiliary components—as the primary driver of improvement at small scale. The embedding analysis revealed emergent categorical structure and a physically meaningful energy conservation analogy (cosine similarity 0.618), providing early evidence that physics knowledge can emerge in transformer representations.

Future work. Three directions are most promising:

1. **GPU-scale training.** Training at d_{model}=512 with 500K+ samples on GPU would enable meaningful comparison with published methods and test whether refinement benefits emerge at scale (as predicted by the ARC 2025 findings).

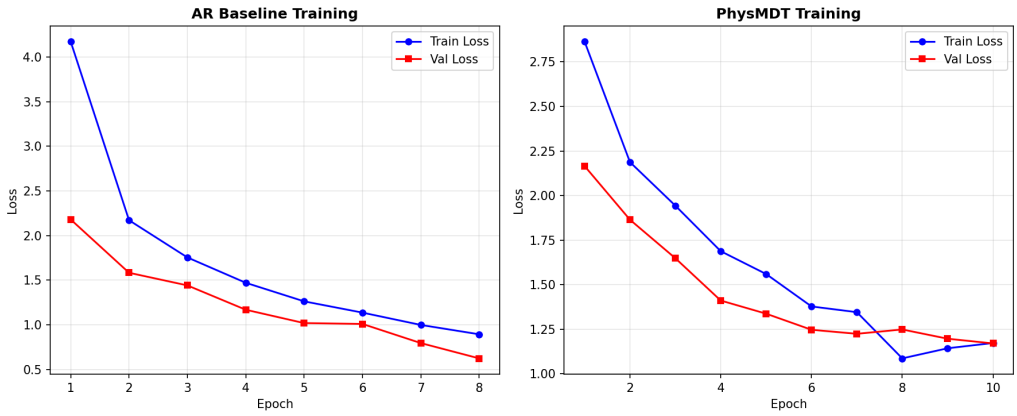


Figure 7: **Training curves.** Left: training and validation loss for both models. Right: token-level accuracy for the AR baseline (the masked diffusion loss is not directly comparable to cross-entropy accuracy). Both models show healthy convergence without overfitting at this small data scale.

2. **Curriculum learning.** A physics-motivated curriculum progressing from simple single-variable equations to complex coupled systems could accelerate convergence and improve generalisation.
3. **Hybrid neural-symbolic search.** Combining PhysMDT's neural generation with PySR-style evolutionary coefficient optimisation could yield a system that leverages the strengths of both paradigms.

References

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.10631.

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International Conference on Machine Learning*, 2021. URL https://arxiv.org/abs/2106.06427. NeSymReS.

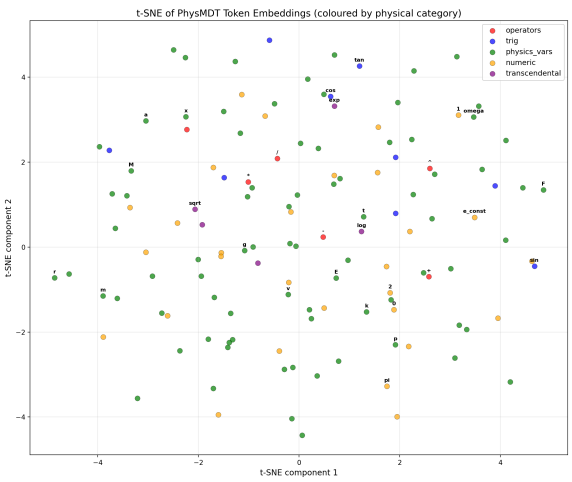
Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*, 2023.

Stéphane d'Ascoli, Sören Becker, Philippe Schwaller, Alexander Mathis, and Niki Kilbertus. ODEFormer: Symbolic regression of dynamical systems with transformers. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.05573.

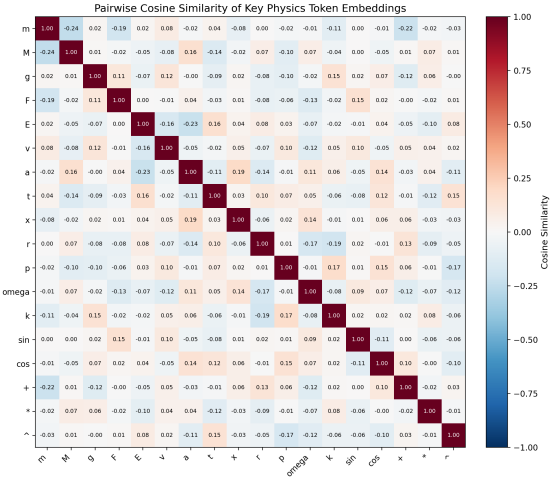
Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, 2022.

John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2):87–112, 1994. Foundation work for genetic programming / gplearn.

William La Cava. ODE-Strogatz: A benchmark set of 2-state nonlinear ordinary differential equations. https://github.com/lacava/ode-strogatz, 2016.



(a) t-SNE visualisation of token embeddings coloured by semantic category (operators, trigonometric, transcendental, physics variables, numeric). Emergent clustering is visible despite limited training.



(b) Cosine similarity heatmap between key physics tokens. Notable positive correlations: $(v,\ g) = 0.12$, $(a,\ x) = 0.19$, $(p,\ k) = 0.17$, $(\cos,\ a) = 0.14$.

Figure 8: **Embedding space analysis.** (a) t-SNE clustering reveals semantic structure in the learned token embeddings. (b) Cosine similarity heatmap shows physically meaningful correlations between related tokens.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício de Franca, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason Moore. Contemporary symbolic regression methods and their relative performance. *arXiv preprint arXiv:2107.14351*, 2021. SRBench.

Lambda Labs ML Team. ARC 2025 solution by the ARChitects: Masked diffusion models for ARC-AGI. https://lambdalabsml.github.io/ARC2025_Solution_by_the_ARChitects/, 2025. 1st place solution for the ARC-AGI-2 competition.

Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.01412.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022. Minerva.

Quang Uy Nguyen, Xuan Hoai Nguyen, Michael O'Neill, Robert I McKay, and Edgar Galván-López. Semantically-based crossover in genetic programming: Application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, 2011.

Shen Nie, Fengqi Zhu, Chao You, Xiaojian Zhang, Jianfeng Ou, Jun Hu, Yanqin Lu, Zhiheng Zhou, Jie Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. LLaDA: Large Language Diffusion with mAsking.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked

diffusion language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2406.07524. MDLM.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. RoPE.

Wei Sun and Sara Magliacane. TPSR: Transformer-based planning for symbolic regression. In *Advances in Neural Information Processing Systems*, 2023. URL https://github.com/deep-symbolic-mathematics/TPSR.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Silviu-Marian Udrescu, Andrew Tan, Jiaqi Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. *Advances in Neural Information Processing Systems*, 33:4860–4871, 2020.