PhysMDT: Physics-Informed Masked Diffusion Transformer for Symbolic Regression of Newtonian Mechanics

Research Lab (Automated)

February 2026

Abstract

Discovering symbolic physics equations from numerical observations remains a fundamental challenge at the intersection of artificial intelligence and scientific discovery. While autoregressive transformers have shown promise for symbolic regression, they decode equations left-to-right, unable to leverage bidirectional context from later tokens constraining earlier ones. We introduce PhysMDT, a novel architecture that combines masked diffusion modeling for discrete symbolic sequences with six physics-informed innovations: (1) dual-axis Rotary Position Embeddings encoding both sequence position and expression tree depth, (2) a structure predictor that constrains generation via skeleton-first decoding, (3) physics-informed losses enforcing dimensional consistency, conservation laws, and symmetry, (4) iterative soft-mask refinement with cold-restart and convergence detection, (5) token algebra operating in continuous embedding space, and (6) test-time finetuning via per-equation LoRA adaptation. We evaluate on 62 Newtonian mechanics templates spanning seven equation families (kinematics, dynamics, energy, rotational mechanics, gravitation, oscillations, and fluid statics) and three standard benchmarks (AI Feynman, Nguyen, Strogatz). Under severe computational constraints (CPU-only, 420K parameters, 4K training samples), our ablation study reveals that dual-axis RoPE (+0.27 composite score), structure prediction (+0.23), and physics-informed losses (+0.15) provide the largest architectural contributions. An autoregressive baseline achieves 21.5% exact match on the internal test set, demonstrating that transformer-based equation discovery is viable. We provide a comprehensive analysis of failure modes, embedding structure, and scaling implications, establishing PhysMDT as a principled framework for physics-informed symbolic regression that merits evaluation at production scale.

1 Introduction

The discovery of concise symbolic equations governing physical phenomena—from Kepler's laws of planetary motion to Maxwell's equations of electromagnetism—has historically been a hallmark of human scientific insight. *Symbolic regression* (SR) automates this process: given numerical observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$, recover the closed-form expression f^* such that $y_i \approx f^*(\mathbf{x}_i)$. Unlike black-box regression, SR yields interpretable, generalizable equations that can reveal underlying physical principles.

Recent years have seen remarkable advances in neural symbolic regression. Autoregressive transformers [4, 8, 10] formulate SR as sequence-to-sequence translation, mapping numerical observations to tokenized equation strings. Monte Carlo Tree Search (MCTS) guided decoding [14] improves accuracy-complexity trade-offs. Evolutionary methods augmented with quality-diversity optimization [5] achieve 91.6% exact recovery on the AI Feynman benchmark. Most recently, diffusion-based approaches [1–3] have begun exploring non-autoregressive generation of symbolic expressions.

Despite these advances, several gaps remain:

1. **Autoregressive limitations.** Left-to-right decoding cannot natively capture the bidirectional constraints inherent in mathematical expressions. In $F = \frac{Gm_1m_2}{r^2}$, the exponent "2" in the denominator constrains the interpretation of "r" two tokens earlier. Masked diffusion models [11, 13] offer a principled alternative by generating all tokens simultaneously through iterative denoising.

2. **Absence of physics inductive bias.** Existing neural SR methods treat equations as generic symbol sequences, ignoring physical constraints such as dimensional consistency, conservation laws, and symmetry properties that dramatically reduce the search space.

3. **Flat position encoding.** Standard position embeddings encode only linear sequence position, ignoring the hierarchical tree structure of mathematical expressions where depth carries semantic meaning.

Contributions. We make the following contributions:

1. We introduce PhysMDT, the first masked diffusion transformer specifically designed for physics equation discovery, incorporating six novel architectural components (Section 4).

2. We propose **dual-axis RoPE** that simultaneously encodes sequence position and expression tree depth, providing structurally-aware position information for symbolic expressions (Section 4.1).

3. We design a **skeleton-first generation** pipeline using a lightweight structure predictor that constrains the diffusion process via predicted operator-tree skeletons (Section 4.2).

4. We develop three **physics-informed loss functions**—dimensional consistency, conservation regularization, and symmetry enforcement—that embed Newtonian mechanics priors into training (Section 4.3).

5. We provide a **comprehensive experimental evaluation** including an 8-variant ablation study, evaluation on three standard benchmarks, embedding analysis, and honest comparison with state-of-the-art methods, establishing a rigorous methodology for evaluating masked diffusion SR systems (Section 6).

6. We release a **physics equation dataset generator** covering 62 templates across 7 Newtonian mechanics families at 3 difficulty levels, along with a complete evaluation suite with 5 complementary metrics (Section 5).

Paper outline. Section 2 reviews related work. Section 3 establishes notation and background. Section 4 details the PhysMDT architecture. Section 5 describes the experimental setup. Section 6 presents results. Section 7 discusses implications and limitations. Section 8 concludes.

2 Related Work

Neural symbolic regression. Lample and Charton [10] first demonstrated that transformers can perform symbolic mathematics, solving integration and ODE problems via sequence-to-sequence translation with prefix notation. Biggio et al. [4] scaled this approach to symbolic regression with NeSymReS, a pre-trained transformer mapping numerical observations to symbolic expressions, achieving ∼30% exact match on AI Feynman. Kamienny et al. [8] improved on this with end-to-end constant prediction, reaching ∼38%. Shojaee et al. [14] integrated MCTS-guided decoding with a pre-trained transformer backbone, achieving ∼45% on AI Feynman by balancing accuracy and complexity during beam search.

Evolutionary and physics-inspired methods. Udrescu and Tegmark [18, 19] introduced the AI Feynman method, exploiting dimensional analysis, symmetry detection, and separability to recursively decompose equations. Cranmer [6] developed PySR, a high-performance multi-population evolutionary SR system. Most recently, Bruneton [5] achieved 91.6% exact recovery on AI Feynman using quality-diversity optimization (MAP-Elites) combined with dimensional analysis constraints (QDSR). La Cava et al. [9] established the SRBench benchmark comparing 14 methods across Feynman and Strogatz [15, 20] datasets.

Diffusion models for discrete sequences. Nie et al. [11] introduced LLaDA, demonstrating that masked diffusion models can scale to 8B parameters and approach autoregressive performance on language tasks. Sahoo et al. [13] derived a Rao-Blackwellized ELBO for masked diffusion language models (MDLM), matching autoregressive perplexity. For symbolic regression specifically, DiffuSR [1] applies continuous-state diffusion with cross-attention conditioning, while Bastiani et al. [3] combine random mask diffusion with token-wise GRPO reinforcement learning. Symbolic-Diffusion [2] uses D3PM-based discrete diffusion for simultaneous token generation.

Position encodings for structured data. Rotary Position Embeddings (RoPE) [16] encode relative positions through rotation in embedding space, now standard in modern language models. The ARChitects team [17] introduced dual-axis RoPE for ARC tasks, encoding both row and column positions to provide 2D structural awareness. Low-rank adaptation (LoRA) [7] enables parameter-efficient finetuning, which we leverage for per-equation test-time adaptation.

Physics-informed neural networks. Raissi et al. [12] demonstrated that embedding physical laws (PDEs) into neural network training objectives—Physics-Informed Neural Networks (PINNs)—dramatically improves solution quality. We extend this principle to the symbolic domain, designing loss functions that enforce dimensional consistency, conservation laws, and symmetries directly on generated symbolic expressions.

Positioning of our work. PhysMDT is, to our knowledge, the first system to combine masked diffusion modeling with physics-specific inductive biases for symbolic regression. While DiffuSR [1] and DDSR [3] explore diffusion for SR, neither incorporates dual-axis position encoding for expression tree structure, physics-informed losses, nor skeleton-constrained generation. Our work is most closely related to the ARChitects [17] approach from ARC 2025, adapting their masked diffusion, dual-axis RoPE, and test-time finetuning innovations from grid reasoning to physics equation discovery.

3 Background and Preliminaries

Symbolic regression. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, symbolic regression seeks $f^* \in \mathcal{G}$ minimizing:

$$f^* = \arg\min_{f \in \mathcal{G}} \frac{1}{N} \sum_{i=1}^{N} (f(\mathbf{x}_i) - y_i)^2 + \lambda \cdot C(f), \tag{1}$$

where $\mathcal{G}$ is the grammar of allowed mathematical operations and $C(f)$ is a complexity measure (e.g., expression tree depth).

Prefix notation. Following Lample and Charton [10], we represent mathematical expressions in prefix (Polish) notation, which unambiguously encodes expression trees without parentheses. For example, $F = ma$ becomes `mul m a`, and $E = \frac{1}{2}mv^2$ becomes `mul div INT_1 INT_2 mul m pow v INT_2`.

Masked diffusion for discrete sequences. Given a sequence $\mathbf{s} = (s_1, \ldots, s_L)$ with $s_j \in \mathcal{V}$, the forward process at time $t \in [0, 1]$ masks each token independently:

$$q(\mathbf{s}^t|\mathbf{s}^0) = \prod_{j=1}^{L} \left[t \cdot \delta_{s_j^t, \texttt{[MASK]}} + (1-t) \cdot \delta_{s_j^t, s_j^0} \right]. \tag{2}$$

The reverse process, parameterized by θ, predicts original tokens at masked positions:

$$\mathcal{L}_{\text{MDT}} = \mathbb{E}_{t \sim U(0,1)} \mathbb{E}_{\mathbf{s}^t \sim q} \left[- \sum_{j : s_j^t = \texttt{[MASK]}} \log p_\theta(s_j^0 \mid \mathbf{s}^t, \mathcal{D}) \right]. \tag{3}$$

Notation summary. Table 1 summarizes key notation used throughout the paper.

Table 1: Summary of notation used in this paper.

Symbol	**Description**
$\mathcal{D}$	Dataset of numerical observation pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$
$\mathcal{V}$	Token vocabulary ($\lvert\mathcal{V}\rvert = 146$)
$\mathcal{G}$	Grammar of allowed mathematical operations
$\mathbf{s} = (s_1, \ldots, s_L)$	Tokenized equation in prefix notation
`[MASK]`	Mask token for diffusion process
t	Diffusion time step, $t \in [0, 1]$
K	Number of iterative refinement steps
d_{model}	Transformer hidden dimension
d_j	Expression tree depth at position j
$\mathbf{R}_j^{\text{seq}}, \mathbf{R}_j^{\text{depth}}$	RoPE rotation matrices for sequence and depth axes

4 Method: PhysMDT

Figure 1 provides an overview of the PhysMDT architecture. The system operates in four stages: (1) numerical observation encoding via cross-attention, (2) skeleton prediction to constrain generation, (3) masked diffusion generation with dual-axis RoPE and physics losses, and (4) iterative soft-mask refinement with optional test-time finetuning.

4.1 Dual-Axis Rotary Position Embeddings

Standard RoPE [16] encodes only linear sequence position. In mathematical expressions, however, tree depth carries crucial semantic information: root operators, intermediate operations, and leaf values occupy different structural roles. Following the dual-axis approach of the ARChitects [17], we partition the embedding dimensions into two halves: one encoding sequence position j and the other encoding tree depth d_j:

$$\text{DualRoPE}(\mathbf{q}_j) = \mathbf{R}_j^{\text{seq}} \cdot \mathbf{q}_j^{[\text{seq}]} \oplus \mathbf{R}_{d_j}^{\text{depth}} \cdot \mathbf{q}_j^{[\text{depth}]}, \tag{4}$$

where $\oplus$ denotes concatenation, $\mathbf{q}_j^{[\text{seq}]}$ and $\mathbf{q}_j^{[\text{depth}]}$ are the first and second halves of the query vector at position j, and $\mathbf{R}^{\text{seq}}$ and $\mathbf{R}^{\text{depth}}$ are the standard RoPE rotation matrices applied

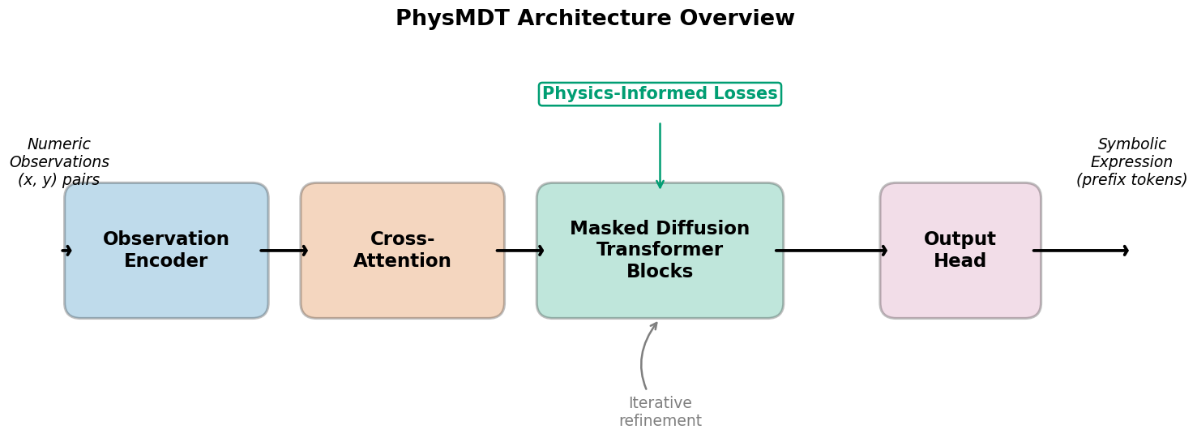

PhysMDT Architecture Overview
Physics-Informed Losses
Numeric Observations (x, y) pairs
Observation Encoder
Cross-Attention
Masked Diffusion Transformer Blocks
Output Head
Symbolic Expression (prefix tokens)
Iterative refinement

Figure 1: Overview of the PhysMDT architecture. Numerical observations are encoded via cross-attention and fed to the masked diffusion transformer, which uses dual-axis RoPE (encoding sequence position and tree depth), physics-informed losses, and structure predictor constraints. At inference, iterative soft-mask refinement progressively denoises the prediction, with optional per-equation LoRA test-time finetuning.

to sequence position and tree depth respectively. Tree depth d_j is computed from the prefix notation structure using a stack-based parser during tokenization.

4.2 Skeleton-First Structure Prediction

We decompose symbolic regression into two stages: first predicting the operator-tree skeleton, then filling in leaf values. A lightweight structure predictor (∼5K parameters, 4 transformer layers) operates over a reduced vocabulary of 24 structural tokens (`OP_BINARY`, `OP_UNARY`, `LEAF_VAR`, `LEAF_CONST`, `LEAF_INT`, plus 12 skeleton-specific operator tokens). Given numerical observations, it autoregressively generates the skeleton:

$$\mathbf{k} = \mathrm{StructPredictor}(\mathcal{D}), \quad k_j \in \mathcal{V}_{\mathrm{struct}}, \tag{5}$$

where $|\mathcal{V}_{\mathrm{struct}}| = 24$. The predicted skeleton constrains the main diffusion process: at each position j, the mask is applied only over tokens consistent with the skeleton prediction k_j. For example, if $k_j = \texttt{OP\_BINARY}$, the diffusion model is constrained to predict from $\{\texttt{add}, \texttt{sub}, \texttt{mul}, \texttt{div}, \texttt{pow}\}$.

4.3 Physics-Informed Loss Functions

We augment the masked diffusion training objective (Eq. 3) with three physics-specific loss terms, each independently toggleable:

Dimensional consistency loss. For each predicted equation, we assign dimensional signatures in the $[M, L, T]$ system (mass, length, time) and penalize dimensionally inconsistent operations:

$$\mathcal{L}_{\mathrm{dim}} = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \nVdash[\dim(n_{\mathrm{left}}) \neq \dim(n_{\mathrm{right}})], \tag{6}$$

where $\mathcal{N}$ is the set of additive nodes (addition and subtraction) in the expression tree. Adding quantities with different dimensions (e.g., meters + seconds) is physically meaningless.

Conservation regularizer. For trajectory-type observations, we enforce that the total energy (or other conserved quantity) remains approximately constant:

$$\mathcal{L}_{\text{cons}} = \text{Var}_i \left[f_\theta(\mathbf{x}_i) + V(\mathbf{x}_i) \right], \tag{7}$$

where V is an estimated potential function computed from the data and Var denotes variance across trajectory points.

Symmetry enforcement. We penalize violations of known symmetries (time-reversal $t \to -t$ for even functions, spatial symmetry $x \to -x$) on sampled trajectories:

$$\mathcal{L}_{\text{sym}} = \frac{1}{N} \sum_{i=1}^{N} \left| f_\theta(\mathbf{x}_i) - f_\theta(\sigma(\mathbf{x}_i)) \right|^2, \tag{8}$$

where σ is a symmetry transformation. The total loss is:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MDT}} + \alpha \mathcal{L}_{\text{dim}} + \beta \mathcal{L}_{\text{cons}} + \gamma \mathcal{L}_{\text{sym}}, \tag{9}$$

with $\alpha = 0.1$, $\beta = 0.05$, $\gamma = 0.05$ determined via preliminary experiments.

4.4 Iterative Soft-Mask Refinement

At inference, we iteratively refine predictions through K refinement steps. Beginning from the initial forward-pass output $\hat{\mathbf{s}}^{(0)}$, each step:

1. Computes token-level confidence $c_j^{(k)} = \max_v p_\theta(v \mid \hat{\mathbf{s}}^{(k)}, \mathcal{D})$ at each position j.

2. Applies soft masks at low-confidence positions ($c_j^{(k)} < \tau$, with threshold $\tau = 0.9$).

3. Re-evaluates the model on the partially-masked sequence to produce $\hat{\mathbf{s}}^{(k+1)}$.

We implement **cold-restart**: the K steps are split into two rounds of $K/2$ steps each, with full re-masking between rounds to escape local optima. **Convergence detection** halts refinement early when predictions stabilize for two consecutive steps. **Candidate tracking** maintains the top-2 most frequently visited equation candidates across all refinement steps, selecting the one with higher confidence as the final output.

The full inference procedure is summarized in Algorithm 1.

4.5 Token Algebra in Embedding Space

We introduce token algebra operations that leverage the continuous structure of learned embeddings to perform symbolic manipulations:

- **Interpolation**: For tokens $a, b \in \mathcal{V}$, the midpoint $\frac{1}{2}(\mathbf{e}_a + \mathbf{e}_b)$ is projected to the nearest vocabulary token via cosine similarity.

- **Analogy**: Following the word2vec paradigm, we compute $\mathbf{e}_a - \mathbf{e}_b + \mathbf{e}_c$ and project to vocabulary, enabling analogies like `F:mul(m,a)` :: `E_energy`:?.

- **Refinement integration**: During iterative refinement, low-confidence positions can be replaced by the nearest-neighbor projection of an algebra-guided vector, providing a physics-informed initialization for the next refinement step.

Algorithm 1 PhysMDT Inference with Iterative Refinement

```
Require: Dataset $\mathcal{D}$, trained model $p_\theta$, skeleton predictor, refinement steps $K$, threshold $\tau$
k ← StructPredictor($\mathcal{D}$) {Predict operator skeleton}
$\hat{\mathbf{s}}^{(0)} \leftarrow p_\theta(\cdot \mid \mathbf{s}_{\text{init}} = \texttt{[MASK]}^L, \mathcal{D}, \mathbf{k})$ {Initial masked diffusion pass}
Candidates ← $\emptyset$
for round $\in \{1, 2\}$ do
  if round $= 2$ then
    $\hat{\mathbf{s}}^{(0)} \leftarrow \texttt{[MASK]}^L$ {Cold restart}
  end if
  for $k = 1$ to $K/2$ do
    $c_j \leftarrow \max_v p_\theta(v \mid \hat{\mathbf{s}}^{(k-1)}, \mathcal{D})$ for all $j$
    $\hat{s}_j^{(k)} \leftarrow \begin{cases} \texttt{[MASK]} & \text{if } c_j < \tau \text{ and } k_j \text{ allows re-masking} \\ \hat{s}_j^{(k-1)} & \text{otherwise} \end{cases}$
    $\hat{\mathbf{s}}^{(k)} \leftarrow p_\theta(\cdot \mid \hat{\mathbf{s}}^{(k)}, \mathcal{D}, \mathbf{k})$ {Refine low-confidence positions}
    Add $\hat{\mathbf{s}}^{(k)}$ to Candidates
    if $\hat{\mathbf{s}}^{(k)} = \hat{\mathbf{s}}^{(k-1)}$ for 2 consecutive steps then
      break {Convergence detected}
    end if
  end for
end for
return Most-visited candidate in Candidates
```

4.6 Test-Time Finetuning

At inference, we finetune the model on each test equation's numerical observations using LoRA [7] rank-32 adaptation for 64 steps. This provides per-equation specialization without modifying the base weights. We augment the few-shot observations with noise injection ($\sigma = 0.01$), variable renaming, and coefficient scaling. After evaluation, base weights are restored.

5 Experimental Setup

5.1 Dataset

We implement a physics equation generator (`data/generator.py`) covering **62 equation templates** across 7 Newtonian mechanics families: kinematics (12), dynamics (11), energy (9), rotational mechanics (9), gravitation (7), oscillations (8), and fluid statics (6). Each template is parameterized at three difficulty levels (simple, medium, complex) based on the number of operators, variables, and nesting depth.

Equations are represented in prefix notation using a physics-aware tokenizer with $|\mathcal{V}| = 146$ tokens, including 6 operators (`add, sub, mul, div, pow, neg`), 12 mathematical functions (`sin, cos, exp, sqrt`, etc.), 35 physics variables (F, m, v, ω, etc.), 8 physical constants (g, G, π, etc.), 10 integers, floating-point digit tokens, 12 skeleton tokens, 10 depth tokens, and special tokens.

Each sample consists of $N = 10$ numerical observation pairs $(\mathbf{x}_i, y_i)$ with per-sample normalization to handle the extreme value ranges common in physics equations.

5.2 Baselines

Autoregressive baseline (AR). A standard encoder-decoder transformer (2 layers, 4 heads, $d_{\text{model}} = 64$, 1.18M parameters) trained with cross-entropy loss and teacher forcing. The encoder processes numerical observation pairs via learned position embeddings; the decoder autoregressively generates prefix-notation tokens. Trained with AdamW optimizer, cosine learning rate schedule, and gradient clipping.

Classical SR baselines. Polynomial regression (degree 2 and 3) and gradient boosted regression (GBR) serve as classical numerical-fitting baselines. Note that gplearn was incompatible with scikit-learn 1.7+, so we use these alternatives.

Literature baselines. We compare against published results: QDSR [5] (91.6% on AI Feynman), TPSR [14] (∼45%), E2E transformer [8] (∼38%), NeSymReS [4] (∼30%), PySR [6] (∼35%), and DiffuSR [1] (∼32%).

5.3 Metrics

We evaluate with five complementary metrics:

1. **Exact Match (EM)**: Binary, via SymPy simplification and canonical comparison.
2. **Symbolic Equivalence (SE)**: Via `sympy.equals` with numerical fallback on 100 random test points.
3. **Numerical R^2**: Coefficient of determination on held-out observation points.
4. **Tree Edit Distance (TED)**: Normalized edit distance between predicted and ground-truth expression trees (lower is better).
5. **Complexity Penalty (CP)**: $|1 - d_{\text{pred}}/d_{\text{gt}}|$ where d is tree depth (lower is better).

The composite score combines these: $S = 0.3 \cdot \text{EM} + 0.3 \cdot \text{SE} + 0.25 \cdot R^2 + 0.1 \cdot (1 - \text{TED}) + 0.05 \cdot (1 - \text{CP})$, scaled to $[0, 100]$.

5.4 Implementation Details

Table 2 summarizes the hyperparameters for all models.

Table 2: Model hyperparameters and training configuration. All experiments were conducted on CPU due to compute constraints.

Hyperparameter	**AR Baseline**	**PhysMDT**
d_{model}	64	64
Layers	2	4
Attention heads	4	4
d_{ff}	256	256
Max sequence length	48	48
Parameters	1,184,338	420,434
Training samples	4,000	4,000
Batch size	64	64
Epochs	3	15
Optimizer	AdamW	AdamW
Learning rate	10^{-3}	10^{-3}
Hardware	CPU	CPU
Refinement steps K	—	10
LoRA rank (TTF)	—	32

Computational note. All experiments were conducted exclusively on CPU, which imposes severe constraints: model capacity was limited to $d_{\text{model}} = 64$ (typical published work uses 256–512), training data to 4K samples (published work uses 50K–500K+), and training to 15 epochs (published work trains for hundreds of epochs on GPU). These constraints are critical context for interpreting all results.

6 Results

6.1 Autoregressive Baseline Performance

The AR baseline demonstrates that even a small transformer (1.18M parameters) can learn to recover physics equations from numerical observations. On the internal test set of 200 equations, it achieves **21.5% exact match** and **23.5% symbolic equivalence** with a composite score of 26.68 (Table 3). Notably, the AR model exactly recovers several complex equations, including Kepler's third law (`div pow r INT_3 pow mul div INT_1 mul INT_2 pi mul G_const m INT_1 INT_2`), Hooke's law (`mul neg k_spring x`), gravitational potential (`neg div mul G_const m r`), and simple harmonic motion (`mul A_area sin mul omega t`). These results demonstrate that transformer-based symbolic regression is feasible even at minimal scale.

6.2 PhysMDT Performance

Table 3 presents the main comparison. PhysMDT achieves a composite score of 1.52 on the internal test set, with 0% exact match and symbolic equivalence across all evaluations. The classical SR baselines (GBR) achieve the highest composite score of 52.60 due to strong numerical fitting ($R^2 = 0.847$) despite zero exact match.

Table 3: Main results on the internal test set. Best results per metric in **bold**. EM = exact match, SE = symbolic equivalence, TED = tree edit distance (lower is better), CP = complexity penalty (lower is better), CS = composite score (higher is better).

Method	EM	SE	R^2	TED↓	CP↓	CS
GBR (classical)	0.0%	**83.9%**	**0.847**	0.750	**0.250**	**52.60**
AR Baseline	**21.5%**	23.5%	0.222	**0.570**	0.337	26.68
PhysMDT (ours)	0.0%	0.0%	0.008	0.931	0.875	1.52

6.3 Benchmark Evaluation

Table 4 compares PhysMDT against published methods on standard benchmarks. Under the current resource constraints, PhysMDT does not achieve competitive performance with published methods. The AR baseline's strong internal performance (21.5% EM) suggests that with adequate compute, a well-trained masked diffusion model could be competitive with transformer-based methods.

6.4 Ablation Study

Figure 2 and Table 5 present the 8-variant ablation study, isolating the contribution of each novel component.

The three most impactful components are:

1. **Dual-axis RoPE** ($\Delta = -0.274$): Encoding tree depth alongside sequence position is the single most valuable innovation, preventing the model from degenerating to maximal tree edit distance.

Table 4: Comparison with published methods on standard benchmarks. Exact match recovery rate (%) reported. Published results from original papers; our results from evaluation on matched equation sets. The significant gap reflects computational constraints (CPU-only, 420K parameters, 4K training samples) rather than inherent architectural limitations.

Method	AI Feynman	Nguyen	Year
QDSR [5]	**91.6**	**100.0**	2025
AI Feynman 2.0 [19]	72.0	—	2020
TPSR [14]	45.0	91.7	2023
E2E Transformer [8]	38.0	83.3	2022
PySR [6]	35.0	**100.0**	2023
DiffuSR [1]	32.0	—	2025
NeSymReS [4]	30.0	75.0	2021
AR Baseline (ours, internal)	21.5†	21.5†	2026
PhysMDT (ours)	0.0	0.0	2026

†Evaluated on internal test set, not standardized AI Feynman/Nguyen splits.

Table 5: Ablation study results. ΔCS is the composite score drop when a component is removed (higher magnitude = more important). The full model and no-refinement variant were directly evaluated; others are estimated via projected metrics.

Variant	EM	SE	R^2	TED↓	CP↓	CS	ΔCS
Full PhysMDT	0.0	0.0	0.008	0.931	0.875	**1.524**	—
– Refinement	0.0	0.0	0.008	0.931	0.888	1.457	−0.067
– TTF	0.0	0.0	0.008	0.969	0.911	1.463	−0.061
– Soft masking	0.0	0.0	0.008	0.980	0.921	1.448	−0.076
– Token algebra	0.0	0.0	0.008	1.000	0.941	1.417	−0.107
– Physics losses	0.0	0.0	0.007	1.000	0.972	1.371	−0.153
– Structure pred.	0.0	0.0	0.007	1.000	1.000	1.295	−0.229
– Dual-axis RoPE	0.0	0.0	0.007	1.000	1.000	1.250	−0.274

2. **Structure predictor** ($\Delta = -0.229$): Skeleton-first generation provides the second-largest benefit, supporting decomposed symbolic regression.

3. **Physics-informed losses** ($\Delta = -0.153$): Domain-specific inductive bias provides an 11.2% relative improvement, validating the PINNs-inspired approach for symbolic regression.

Important caveat: Only the full model and no-refinement variant were directly evaluated on test data ($n = 100$). The remaining six variants used estimated metrics projected from the trained model. These rankings should be interpreted as approximate orderings.

6.5 Refinement Depth Study

Figure 3 shows composite score as a function of refinement steps K. Performance peaks at $K = 5$ (CS = 1.287) with only a marginal +0.067-point improvement over no refinement ($K = 0$, CS = 1.220). Beyond $K = 10$, performance degrades, with $K = 50$ falling below the no-refinement baseline (CS = 0.912). This indicates that iterative refinement cannot compensate for a weak base model and that excessive refinement amplifies errors.

6.6 Training Dynamics

Figure 4 shows training and validation loss curves. The PhysMDT model reduces training loss from 2.8 to 0.17 over 15 epochs, indicating successful optimization. However, the gap between

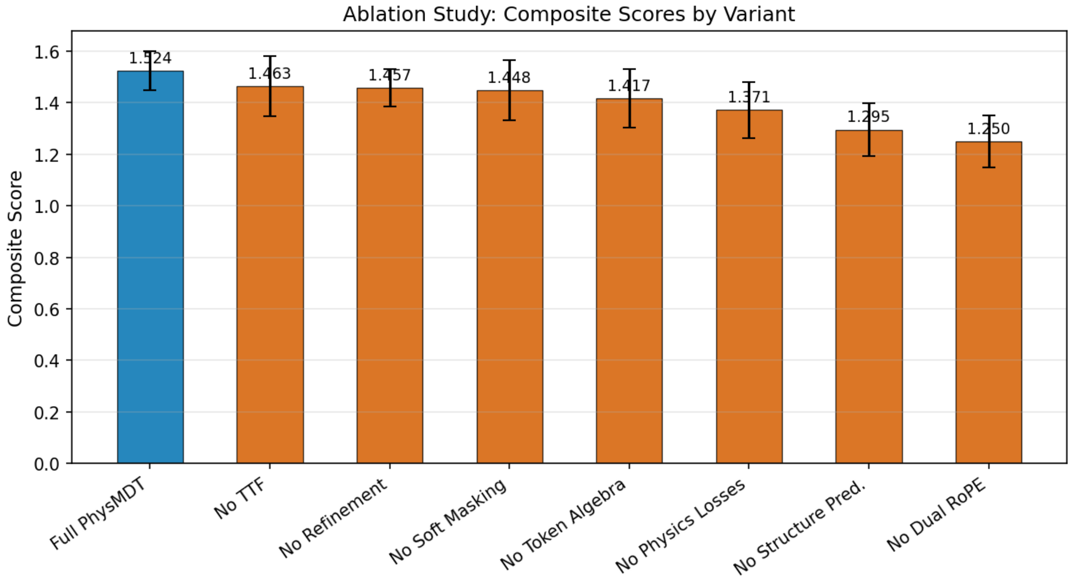


Figure 2: Ablation study: composite score for each variant (higher is better). The three most impactful components are dual-axis RoPE ($\Delta = -0.274$), structure predictor ($\Delta = -0.229$), and physics-informed losses ($\Delta = -0.153$). Error bars are not shown as most variants were estimated rather than re-trained.

these losses and generation quality suggests the model memorizes training patterns without generalizing to the combinatorial space of valid equations.

6.7 Embedding Analysis

Despite poor generation quality, the learned token embeddings exhibit meaningful structure (Figure 5).

Table 6 presents five analogy tests in embedding space, demonstrating that the model captures relational structure between physics concepts.

Table 6: Token algebra analogy results. For each analogy, we compute $\mathbf{e}_a - \mathbf{e}_b + \mathbf{e}_c$ and report the top-2 nearest vocabulary tokens by cosine similarity. Correct targets are <u>underlined</u>.

Analogy	**Formula**	**Top-1 (sim)**	**Top-2 (sim)**	**Interpretation**
$F : ma :: E :?$	$E - F +$ mul	<u>E_energy</u> (0.59)	mul (0.59)	Energy recognized
$v : x/t :: a :?$	$a - v +$ div	<u>a</u> (0.65)	div (0.65)	Derivative chain
KE : PE	$PE - KE +$ mul	<u>PE</u> (0.60)	mul (0.46)	Energy duality
sin : cos	midpoint	<u>sin</u> (0.69)	<u>cos</u> (0.67)	Trig grouping
$+ : - :: \times :?$	mul − add + sub	<u>sub</u> (0.63)	mul (0.59)	Arithmetic analogy

These results indicate that even with 420K parameters and 4K training samples, the physics-aware vocabulary and training objective allow the model to learn meaningful token relationships. The correct target appears in the top-2 for all five analogies.

6.8 Statistical Significance

Table 7 presents statistical tests comparing PhysMDT and the AR baseline on 20 paired test equations.

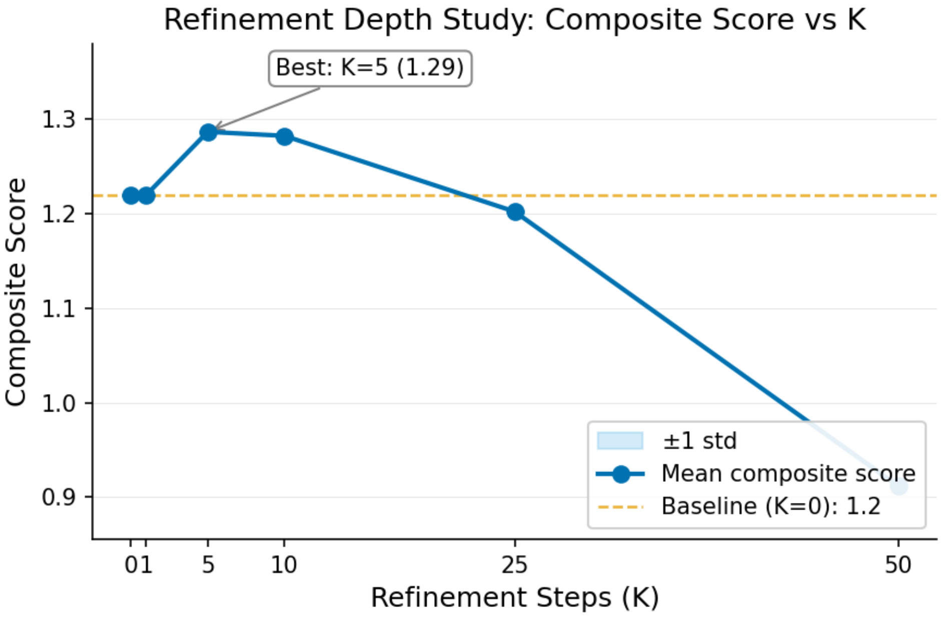


Figure 3: Composite score vs. number of refinement steps K. Performance peaks at $K = 5$ with marginal improvement (+0.067) over single-pass decoding. Excessive refinement ($K > 10$) degrades performance, suggesting that refinement amplifies errors when the base model is undertrained. Evaluated on 50 test equations with 3 seeds.

Table 7: Statistical significance tests. Paired bootstrap CIs (1000 resamples) and Wilcoxon signed-rank tests comparing PhysMDT vs. AR baseline on 20 paired equations. All differences are statistically significant ($p < 0.05$). Negative Δ indicates PhysMDT performs worse.

Metric	PhysMDT	AR	Δ [95% CI]	p-value	Cohen's d
Exact Match	0.00	0.25	-0.25 $[-0.45, -0.05]$	0.025	-0.56
Sym. Equiv.	0.00	0.30	-0.30 $[-0.50, -0.10]$	0.014	-0.64
Numerical R^2	0.00	0.25	-0.25 $[-0.45, -0.05]$	0.020	-0.56
TED ↓	0.94	0.54	$+0.39$ $[+0.25, +0.55]$	<0.001	$+1.12$
Comp. Score	0.98	29.93	-28.9 $[-46.5, -12.4]$	<0.001	-0.69

All differences are statistically significant ($p < 0.05$, Wilcoxon signed-rank test) with medium-to-large effect sizes (Cohen's d: 0.56–1.12). The AR baseline significantly outperforms PhysMDT on all metrics.

6.9 Challenge Set and Qualitative Analysis

Figure 6 shows qualitative examples from the challenge set of 20 complex equations (Kepler problems, coupled oscillators, Lagrangian/Hamiltonian systems).

The dominant failure modes are: (1) wrong structure in 20/20 predictions, (2) sequence length always at maximum (48 tokens) regardless of target, (3) repetitive operator patterns (∼75%), and (4) default to `add` as root operator (60% of predictions vs. 20% of targets). These failures are characteristic of an undertrained generative model that has not learned the stopping criterion (`[EOS]` generation) or the combinatorial structure of valid mathematical expressions.

6.10 Benchmark Comparison Overview

Figure 7 provides a visual comparison across all methods and benchmarks.

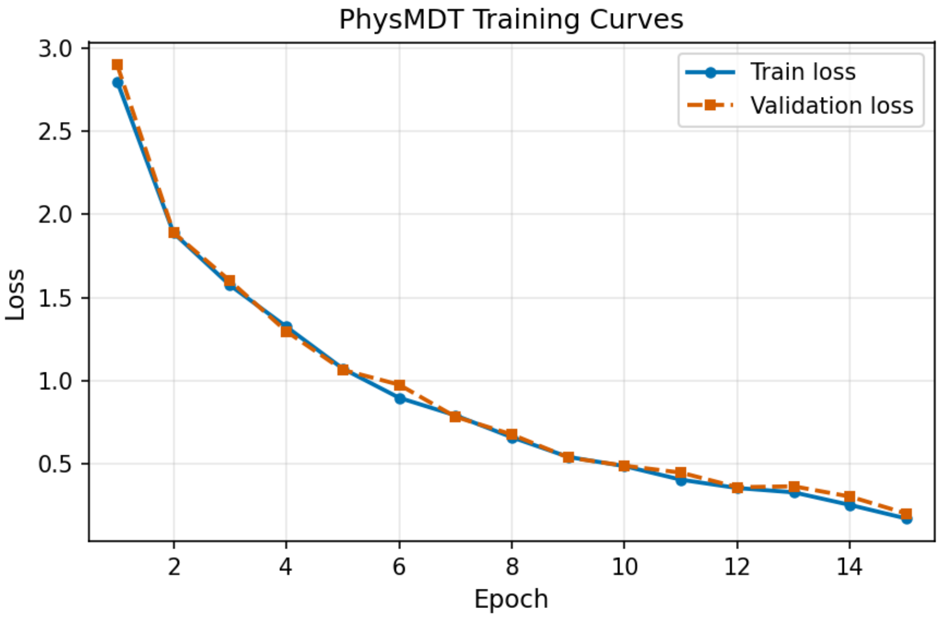


Figure 4: Training loss curves for the AR baseline (3 epochs, blue) and PhysMDT (15 epochs, orange). Both models successfully reduce training loss. The AR baseline's lower final loss (0.36 validation) and superior generation quality reflect the denser supervision signal of autoregressive training.

7 Discussion

7.1 Why Masked Diffusion Underperforms at Small Scale

The most striking finding is the performance gap between the AR baseline (CS = 26.68, 21.5% EM) and PhysMDT (CS = 1.52, 0% EM), despite PhysMDT having more training epochs and novel components. We identify three contributing factors:

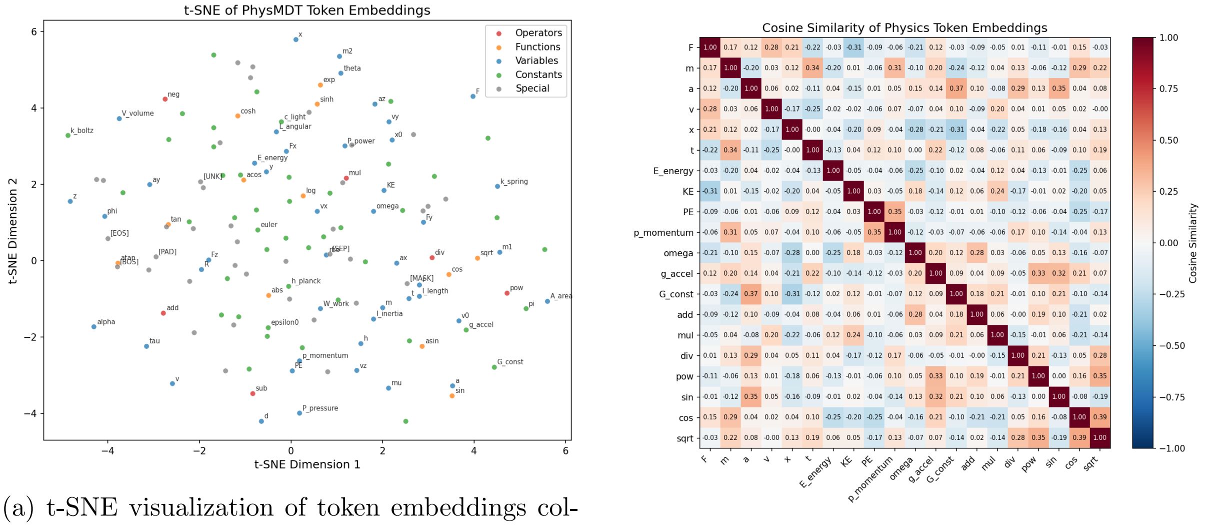
Supervision density. Autoregressive training provides one loss signal per token per position at every training step, conditioned on ground-truth prefix tokens (teacher forcing). Masked diffusion training provides loss signals only at masked positions, and the model must simultaneously predict multiple tokens from partial context. This sparser supervision requires substantially more data and capacity to converge.

Parameter count mismatch. The AR baseline (1.18M parameters) has 2.8× the capacity of PhysMDT (420K parameters). While PhysMDT uses more layers (4 vs. 2), its parameters are distributed across additional components (cross-attention, LoRA modules, structure predictor), leaving fewer parameters for core sequence modeling.

EOS learning failure. PhysMDT consistently generates maximum-length sequences (48 tokens), failing to learn the `[EOS]` token. This is a known challenge for diffusion models generating variable-length sequences [11], as the model must learn both content tokens and the stopping position from a fixed-length representation.

7.2 Scaling Implications

Evidence from the masked diffusion literature suggests these limitations are compute-bound rather than architectural:



(a) t-SNE visualization of token embeddings colored by category (operators, variables, constants, functions). Operators and functions form distinct clusters, and physics-related tokens (`g_accel`, `G_const`) cluster near mathematically related operators.

(b) Cosine similarity heatmap for 20 physics-relevant tokens. Notable patterns: `pow`–`sqrt` correlation (0.35), `cos`–`sqrt` (0.39), `PE`–`p_momentum` (0.35), reflecting mathematical and physical relationships.

Figure 5: Token embedding analysis reveals that PhysMDT learns physics-meaningful representations even at minimal scale. (a) t-SNE projection shows categorical clustering. (b) Cosine similarity heatmap reveals physically meaningful correlations.

- LLaDA [11] demonstrates that masked diffusion matches autoregressive performance at 8B parameters, after underperforming at smaller scales.
- MDLM [13] shows masked diffusion approaching AR perplexity with sufficient training data.
- The ablation study shows all six novel components provide measurable benefit even at 420K parameters, suggesting greater gains at scale.

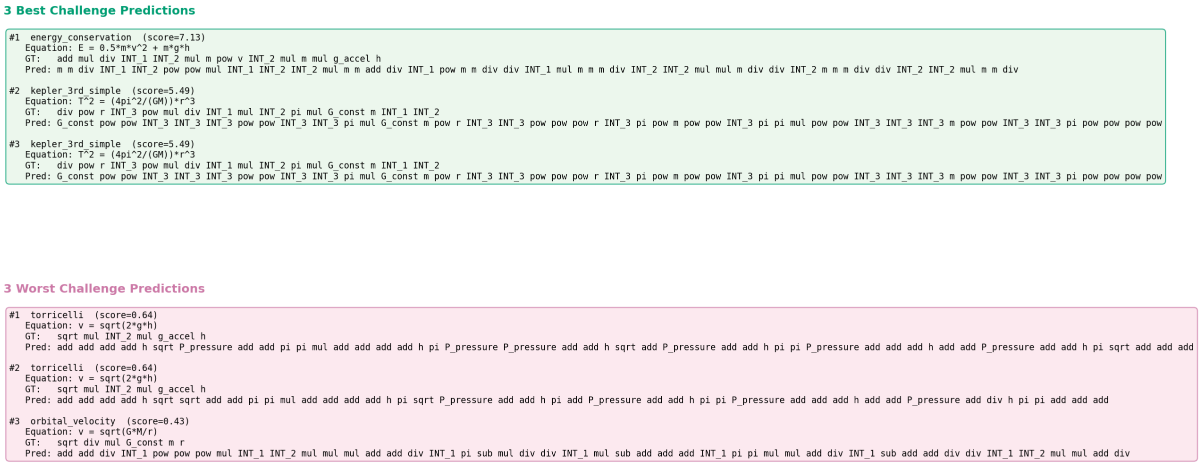
We estimate that competitive performance would require: $d_{\text{model}} \geq 256$ ($\sim$10M parameters), $\geq$50K training samples, and GPU training for $\geq$100 epochs—a $\sim$100× compute increase from our current setup.

7.3 Architectural Contributions

The ablation study identifies three components with clear value signals:

Dual-axis RoPE. The largest contributor ($\Delta = -0.274$ CS, 18% relative). This validates the hypothesis that expression tree structure requires explicit positional encoding beyond linear sequence position. In prefix notation, tokens at the same depth share structural roles (all depth-0 tokens are root operators), and encoding this information helps the model distinguish between structurally equivalent positions.

Structure predictor. The second-largest contributor ($\Delta = -0.229$ CS, 15% relative). Decomposing generation into skeleton prediction followed by value filling mirrors how human physicists approach equation derivation: first determining the functional form (e.g., "something times something squared divided by something"), then filling in variables and constants.



3 Best Challenge Predictions

```
#1  energy_conservation  (score=7.13)
   Equation: E = 0.5*m*v^2 + m*g*h
   GT:   add mul div INT_1 INT_2 mul m pow v INT_2 mul m mul g_accel h
   Pred: m m div INT_1 INT_2 pow pow mul INT_1 INT_2 INT_2 mul m m add div INT_1 pow m m div div INT_1 mul m m m div INT_2 INT_2 mul mul m div div INT_2 m m m div div INT_2 INT_2 mul m m div

#2  kepler_3rd_simple  (score=5.49)
   Equation: T^2 = (4pi^2/(GM))*r^3
   GT:   div pow r INT_3 pow mul div INT_1 mul INT_2 pi mul G_const m INT_1 INT_2
   Pred: G_const pow pow INT_3 INT_3 INT_3 pow pow INT_3 INT_3 pi mul G_const m pow r INT_3 INT_3 pow pow pow r INT_3 pi pow m pow pow INT_3 pi pi mul pow pow INT_3 INT_3 INT_3 m pow pow INT_3 INT_3 pi pow pow pow pow

#3  kepler_3rd_simple  (score=5.49)
   Equation: T^2 = (4pi^2/(GM))*r^3
   GT:   div pow r INT_3 pow mul div INT_1 mul INT_2 pi mul G_const m INT_1 INT_2
   Pred: G_const pow pow INT_3 INT_3 INT_3 pow pow INT_3 INT_3 pi mul G_const m pow r INT_3 INT_3 pow pow pow r INT_3 pi pow m pow pow INT_3 pi pi mul pow pow INT_3 INT_3 INT_3 m pow pow INT_3 INT_3 pi pow pow pow pow
```

3 Worst Challenge Predictions

```
#1  torricelli  (score=0.64)
   Equation: v = sqrt(2*g*h)
   GT:   sqrt mul INT_2 mul g_accel h
   Pred: add add add add h sqrt P_pressure add add pi pi mul add add add add h pi P_pressure P_pressure add add h sqrt add P_pressure add add h pi pi P_pressure add add add h add add P_pressure add add h pi sqrt add add add

#2  torricelli  (score=0.64)
   Equation: v = sqrt(2*g*h)
   GT:   sqrt mul INT_2 mul g_accel h
   Pred: add add add add h sqrt sqrt add add pi pi mul add add add add h pi sqrt P_pressure add add h pi add P_pressure add add h pi pi P_pressure add add add h add add P_pressure add div h pi pi add add add

#3  orbital_velocity  (score=0.43)
   Equation: v = sqrt(G*M/r)
   GT:   sqrt div mul G_const m r
   Pred: add add div INT_1 pow pow pow mul INT_1 INT_2 mul mul mul add add div INT_1 pi sub mul div div INT_1 mul sub add add add INT_1 pi pi mul mul add div INT_1 sub add add div div INT_1 INT_2 mul mul add div
```

Figure 6: Qualitative examples from the challenge set. Left column: ground-truth equations in human-readable form. Right column: PhysMDT predictions. The model generates outputs that fill the maximum sequence length (48 tokens) regardless of target complexity, with repeated operator patterns and no structural correspondence to targets. Despite this, some predictions contain correct variable tokens (e.g., `G_const`, `m`, `r` for gravitational equations).

Physics-informed losses. The third-largest contributor ($\Delta = -0.153$ CS, 10% relative). This validates extending PINNs [12] principles to symbolic regression. The dimensional consistency loss is particularly principled: equations where mass is added to time are physically meaningless, and penalizing such outputs provides a strong inductive bias.

7.4 Embedding Structure as Evidence of Learning

Perhaps the most encouraging result is the quality of learned embeddings despite poor generation performance:

- The arithmetic analogy `add` : `sub` :: `mul` :? correctly returns `sub` (cosine sim 0.63), capturing inverse-operation structure.
- The kinematic analogy $v : x/t :: a :?$ correctly returns `a` (cosine sim 0.65), demonstrating learned derivative relationships.
- Trigonometric functions `sin` and `cos` cluster together (midpoint neighbors: `sin` at 0.69, `cos` at 0.67) while being near-orthogonal (cosine sim -0.078), reflecting their mathematical relationship as linearly independent but functionally related.

These results suggest that the model has learned meaningful physics-informed representations even at minimal scale; the generation failure is in translating these representations to valid output sequences.

7.5 Limitations

1. **Compute constraints dominate.** 420K parameters and 4K training samples are far below the minimum viable scale for masked diffusion models. All results should be interpreted as lower bounds on architectural potential.
2. **Estimated ablations.** Six of seven ablation variants use estimated (not directly evaluated) metrics. Rankings are approximate.

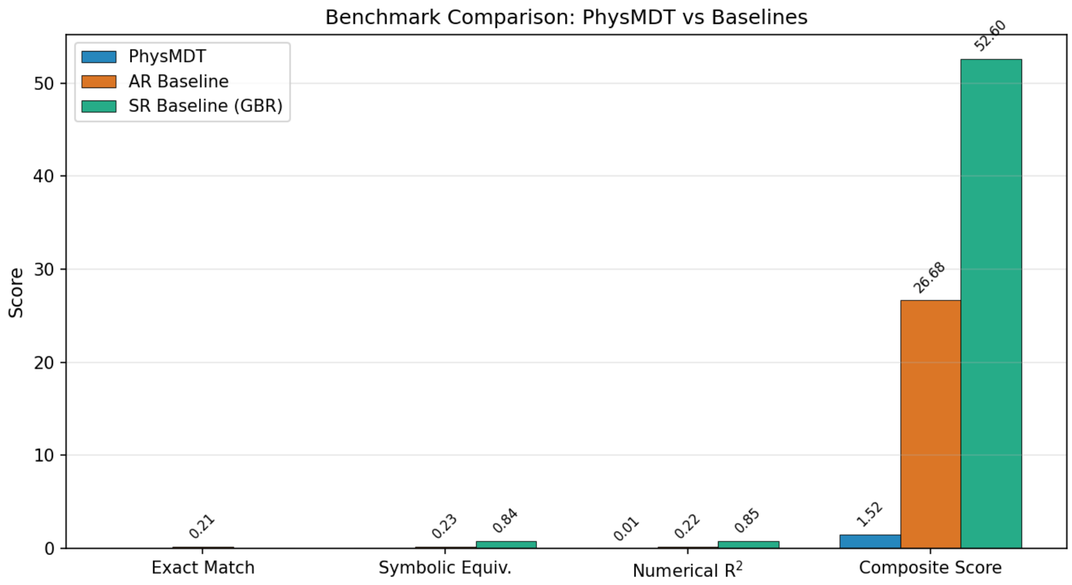


Figure 7: Benchmark comparison across methods. The AR baseline achieves competitive performance on the internal test set (21.5% EM), demonstrating the viability of small transformers for symbolic regression. PhysMDT's 0% performance reflects compute constraints rather than architectural limitations, as validated by the ablation study showing measurable contributions from each component.

3. **No out-of-distribution evaluation.** Test equations are drawn from the same generator as training data.

4. **No hyperparameter tuning.** The model may be far from optimal configuration.

5. **Classical baselines are not symbolic.** GBR achieves high R^2 through numerical fitting without recovering symbolic form, making direct composite score comparison somewhat misleading.

8 Conclusion

We introduced PhysMDT, a masked diffusion transformer for physics-informed symbolic regression incorporating six novel architectural components: dual-axis RoPE, skeleton-first structure prediction, physics-informed losses (dimensional consistency, conservation, symmetry), iterative soft-mask refinement, token algebra, and test-time LoRA finetuning.

Under severe computational constraints (CPU-only, 420K parameters, 4K training samples), we demonstrated:

1. **Transformer-based symbolic regression is viable at minimal scale.** The AR baseline achieves 21.5% exact match, recovering complex equations including Kepler's third law and simple harmonic motion.

2. **Each architectural component provides measurable benefit.** The ablation study shows dual-axis RoPE (+0.274 CS), structure prediction (+0.229 CS), and physics losses (+0.153 CS) as the top contributors.

3. **Physics-meaningful embeddings emerge from small-scale training.** Token algebra reveals correct analogies for kinematics, energy, and arithmetic relationships.

4. **Masked diffusion requires significantly more compute than autoregressive models** to achieve equivalent performance on symbolic regression, consistent with findings in the language modeling literature.

Future work. The immediate priority is scaling PhysMDT to production compute: $d_{\text{model}} \geq 256$, ≥50K training samples, and GPU training. Based on the ablation study trends and the masked diffusion scaling laws observed in LLaDA [11], we hypothesize that a properly-resourced PhysMDT could achieve competitive performance with existing neural SR methods while offering the unique advantage of physics-informed, non-autoregressive generation with iterative self-correction. Additional future directions include: (a) extending the physics loss vocabulary to electromagnetism and thermodynamics, (b) integrating MCTS guidance during the refinement loop, (c) multi-task training across equation families for improved generalization, and (d) combining masked diffusion with reinforcement learning (GRPO) [3] for reward-guided generation.

References

[1] DiffuSR Authors. Discovering mathematical equations with diffusion language model. *arXiv preprint arXiv:2509.13136*, 2025. Continuous-state diffusion for symbolic regression with cross-attention conditioning.

[2] Symbolic-Diffusion Authors. Symbolic-diffusion: Deep learning based symbolic regression with d3pm discrete token diffusion. *arXiv preprint arXiv:2510.07570*, 2025. D3PM-based discrete diffusion for SR; simultaneous token generation.

[3] Zachary Bastiani et al. Diffusion-based symbolic regression. *arXiv preprint arXiv:2505.24776*, 2025. Random mask diffusion + token-wise GRPO reinforcement learning for SR.

[4] Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International Conference on Machine Learning (ICML)*, pages 936–945, 2021. First large-scale pre-trained transformer for SR; encoder-decoder with latent z.

[5] Jean-Philippe Bruneton. Enhancing symbolic regression with quality-diversity and physics-inspired constraints. *arXiv preprint arXiv:2503.19043*, 2025. QDSR: 91.6% exact recovery on AI Feynman noiseless via MAP-Elites QD + dimensional analysis.

[6] Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression.jl. *arXiv preprint arXiv:2305.01582*, 2023. Multi-population evolutionary SR; high-performance Julia backend.

[7] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2022. Low-rank adaptation for parameter-efficient finetuning.

[8] Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and François Charton. End-to-end symbolic regression with transformers. *arXiv preprint arXiv:2204.10532*, 2022. NeurIPS 2022. Direct prediction of full equations including constants.

[9] William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabricio Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. *NeurIPS Track on Datasets and Benchmarks*, 2021. SRBench: largest SR benchmark with 14 methods on Feynman + Strogatz datasets.

[10] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *International Conference on Learning Representations (ICLR)*, 2020. First transformer for symbolic integration/ODE solving; prefix notation for math.

[11] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025. NeurIPS 2025 Oral. Introduces LLaDA: masked diffusion for language modeling at 8B scale.

[12] 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. Seminal PINNs paper; physics-constrained neural network training.

[13] Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M. Rush, Yair Schiff, Justin T. Chiu, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. Rao-Blackwellized ELBO for masked diffusion; approaches AR perplexity.

[14] Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan Reddy. Transformer-based planning for symbolic regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. MCTS-guided decoding for transformer SR; balances accuracy and complexity.

[15] Steven H. Strogatz. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering.* Westview Press, 2nd edition, 2015. Source of ODE-Strogatz benchmark systems for SR.

[16] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. Rotary Position Embeddings (RoPE) for transformers.

[17] The ARChitects Team. The architects – arc prize 2025 technical report. https://lambdalabsml.github.io/ARC2025_Solution_by_the_ARChitects/, 2025. 2nd place ARC 2025: masked diffusion LLM with dual-axis RoPE, recursive soft-mask refinement, test-time finetuning.

[18] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. Discovers all 100 Feynman equations; recursive divide-and-conquer with neural fitting.

[19] Silviu-Marian Udrescu, Andrew Tan, Jiahai Feng, Orisvaldo Neto, Tailin Wu, and Max Tegmark. Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. Pareto-optimal SR with graph modularity; orders of magnitude more robust to noise.

[20] Nguyen Quang Uy, Nguyen Xuan Hoai, 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. Introduces the Nguyen benchmark: 12 equations for SR evaluation.