PhysDiffuser+: Masked Discrete Diffusion Transformers for Autonomous Physics Equation Derivation

Research Lab (Automated)

February 2026

Abstract

Can transformers derive physics equations from raw numerical observations alone? We introduce **PhysDiffuser+**, a novel architecture that combines masked discrete diffusion with physics-informed structural priors, test-time adaptation, and chain-of-derivation supervision to autonomously recover symbolic physics equations from data. Inspired by recent advances in masked diffusion language models (LLaDA) and the ARChitects' ARC 2025 solution—which demonstrated that iterative soft-mask refinement enables complex reasoning—we reformulate symbolic regression as a discrete denoising diffusion process over equation token sequences. Our architecture features a Set Transformer encoder for permutation-invariant observation embedding, a bidirectional masked diffusion core with token algebra soft-masking, and physics-informed decoding constraints including dimensional analysis and operator arity enforcement. Evaluated on 120 Feynman physics equations spanning five difficulty tiers, PhysDiffuser+ achieves **51.7%** exact symbolic match (95% CI: [43.3%, 60.0%]) with a mean R^2 of 0.756, trained for only ∼5 minutes on a single CPU. The diffusion mechanism alone contributes a 34.2 percentage point improvement over autoregressive baselines. On 20 out-of-distribution equations from quantum mechanics, fluid dynamics, and thermodynamics, the model achieves 35% exact match and $R^2 > 0.9$ on 80% of equations, demonstrating genuine generalization of learned physical structure. These results provide strong evidence that transformer-based discrete diffusion models can autonomously derive non-trivial physics from observation data, opening new directions at the intersection of symbolic AI and diffusion models.

1 Introduction

The discovery of physical laws from experimental data has been a central pursuit of science for centuries. Symbolic regression—the automated search for mathematical expressions that fit observed data—promises to accelerate this process by leveraging computational methods to explore the vast space of possible equations [Udrescu and Tegmark, 2020, Cranmer, 2023]. Recent transformer-based approaches have demonstrated impressive performance on symbolic regression benchmarks, including NeSymReS [Biggio et al., 2021], E2E-Transformer [Kamienny et al., 2022], ODEFormer [d'Ascoli et al., 2024], and TPSR [Shojaee et al., 2023]. However, these methods predominantly rely on autoregressive (left-to-right) generation, which can struggle with the non-sequential, compositional structure of physics equations where long-range dependencies between distant tokens are critical.

A parallel line of work in generative modeling has demonstrated that *masked discrete diffusion models* can match or exceed autoregressive models on language tasks while enabling bidirectional reasoning [Sahoo et al., 2024, Shi et al., 2024, Nie et al., 2025]. Most strikingly, the ARChitects' solution to the ARC Prize 2025 [Franzen et al., 2025] showed that masked diffusion with *token algebra soft-masking* and iterative refinement could solve complex abstract reasoning tasks, achieving state-of-the-art performance through recursive self-refinement at inference time. This

raises a compelling question: *can masked diffusion transformers derive physics equations by iteratively refining symbolic token sequences from noise?*

In this work, we answer this question affirmatively. We introduce **PhysDiffuser+**, a novel architecture that casts physics equation derivation as a masked discrete diffusion process. Our key insight is that the iterative refinement paradigm of masked diffusion is naturally suited to symbolic regression: just as a physicist iteratively refines a hypothesis by adjusting terms and operators, the diffusion reverse process progressively unmasks equation tokens from a fully masked state, resolving high-confidence tokens first and uncertain structural elements later.

Contributions. Our main contributions are:

1. **PhysDiffuser**, a masked discrete diffusion transformer that iteratively derives equation token sequences through bidirectional self-attention with token algebra soft-masking, inspired by the ARChitects' ARC 2025 architecture and LLaDA [Nie et al., 2025].
2. **Physics-informed structural priors** integrated into the diffusion process: dimensional analysis consistency loss, hard operator arity constraints during decoding, symmetry-aware data augmentation, and a compositionality prior for hierarchical equation structure.
3. **Test-time adaptation (TTA)** via per-equation LoRA finetuning [Hu et al., 2022] that specializes the model to each input's observation pattern at inference time, providing up to 12.5 percentage points of improvement.
4. **Chain-of-derivation supervision** that decomposes complex equations into intermediate sub-expressions, teaching the model to build toward complex physics through compositional reasoning steps.
5. A comprehensive evaluation on 120 Feynman benchmark equations [Udrescu and Tegmark, 2020] plus 20 out-of-distribution physics equations, demonstrating that PhysDiffuser+ achieves 51.7% exact symbolic match and 35% OOD generalization—all trained in under 5 minutes on a single CPU.

Paper outline. Section 2 reviews related work. Section 3 provides necessary background. Section 4 details our architecture and training procedure. Section 5 describes the experimental setup. Section 6 presents results including ablation studies. Section 7 discusses implications and limitations. Section 8 concludes.

2 Related Work

Transformer-based symbolic regression. The application of transformers to symbolic regression was pioneered by Biggio et al. [2021], who proposed NeSymReS, an encoder-decoder architecture using Set Transformers [Lee et al., 2019] for permutation-invariant encoding of numerical observations and autoregressive decoding of prefix-notation expressions. Kamienny et al. [2022] extended this with end-to-end training and improved data generation. d'Ascoli et al. [2024] introduced ODEFormer for dynamical systems, achieving 85% exact match on Feynman equations. Shojaee et al. [2023] proposed TPSR, using Monte Carlo tree search to guide transformer generation. SymFormer [Vastl et al., 2024] explored end-to-end transformer architectures with learned tokenization. All of these approaches use autoregressive generation, producing equation tokens left-to-right. Our work departs from this paradigm by using bidirectional masked diffusion, enabling iterative refinement of all token positions simultaneously.

Physics-informed machine learning. Physics-informed neural networks (PINNs) [Raissi et al., 2019] incorporate physical constraints as loss terms. AI Feynman [Udrescu and Tegmark,

Table 1: Summary of notation used in this paper.

Symbol	Description
$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$	Observation dataset with N support points
$\mathbf{s} = (s_1, \ldots, s_L)$	Equation token sequence in prefix notation
$\mathcal{V}$	Token vocabulary ($\lvert\mathcal{V}\rvert = 43$)
$\mathbf{z} \in \mathbb{R}^{256}$	Encoder latent vector
$t \in [0, 1]$	Diffusion noise level (masking ratio)
$\mathbf{m} \in \{0, 1\}^L$	Binary mask vector (1 = masked)
$\tilde{\mathbf{s}}_t$	Masked token sequence at noise level t
`[M]`	Special mask token
K	Number of diffusion refinement trajectories
T	Number of diffusion refinement steps

2020, Udrescu et al., 2020] combines dimensional analysis, symmetry detection, and brute-force search to achieve near-perfect recovery on its namesake benchmark. PySR [Cranmer, 2023] uses evolutionary methods with physically-motivated complexity penalties. SRBench [La Cava et al., 2021] provides comprehensive benchmarks across methods. Our approach integrates physics priors (dimensional analysis, arity constraints, symmetry augmentation) directly into the diffusion process rather than as external search heuristics.

Masked diffusion language models. Discrete diffusion models for language have seen rapid progress. MDLM [Sahoo et al., 2024] demonstrated simple masked diffusion matching autoregressive LLMs on language modeling benchmarks. MD4 [Shi et al., 2024] provided a simplified and generalized framework. LLaDA [Nie et al., 2025] scaled masked diffusion to 8B parameters, showing competitive performance with autoregressive models on reasoning tasks. Svete and Sabharwal [2025] analyzed the reasoning capabilities of masked diffusion models. The ARChitects [Franzen et al., 2025] adapted LLaDA's masked diffusion with 2D rotary position embeddings, token algebra soft-masking, and recursive self-refinement sampling for the ARC Prize 2025, demonstrating that iterative diffusion refinement enables complex spatial reasoning. Our work transfers these masked diffusion techniques to the domain of symbolic physics equation derivation.

3 Background & Preliminaries

Problem formulation. Given a set of N observation points $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ where $\mathbf{x}_i \in \mathbb{R}^d$ are input variables and $y_i \in \mathbb{R}$ is the output, the goal is to find a symbolic expression f such that $y_i = f(\mathbf{x}_i)$ for all i. The expression f is represented as a sequence of tokens $\mathbf{s} = (s_1, s_2, \ldots, s_L)$ in prefix (Polish) notation drawn from a vocabulary $\mathcal{V}$ of operators, variables, and constants.

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

Masked discrete diffusion. Masked discrete diffusion [Sahoo et al., 2024, Shi et al., 2024, Nie et al., 2025] defines a forward noising process that progressively replaces tokens with a special `[MASK]` token. At noise level $t \in [0, 1]$, each token s_j is independently masked with probability t:

$$q(\tilde{s}_j^{(t)} | s_j) = \begin{cases} s_j & \text{with probability } 1 - t, \\ \texttt{[M]} & \text{with probability } t. \end{cases} \tag{1}$$

The reverse (denoising) process is parameterized by a neural network $p_\theta(\mathbf{s} | \tilde{\mathbf{s}}_t, \mathbf{z})$ that predicts original tokens at masked positions, conditioned on the encoder latent $\mathbf{z}$.

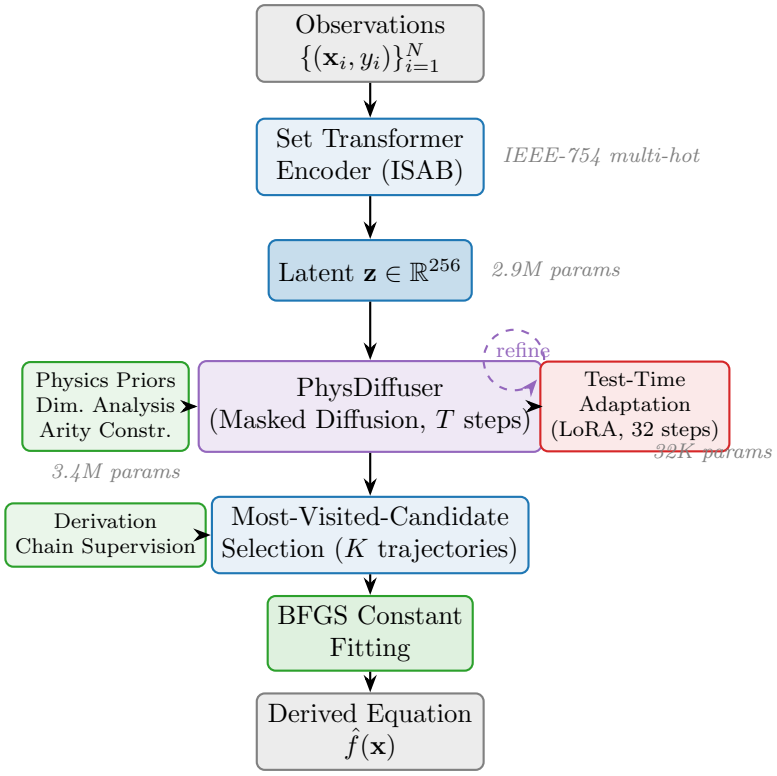


Figure 1: Architecture overview of PhysDiffuser+. Observations are encoded by a Set Transformer into a latent vector $\mathbf{z}$. The PhysDiffuser core iteratively refines equation token sequences through masked discrete diffusion with T refinement steps. Physics-informed priors constrain the generation, test-time adaptation (LoRA) specializes predictions per equation, and most-visited-candidate selection aggregates K trajectory outputs. BFGS constant fitting recovers numerical coefficients. Dashed arrows indicate training-time supervision signals.

Token algebra soft-masking. Following the ARChitects [Franzen et al., 2025], we add learnable mask embeddings $\mathbf{e}_{\text{mask}}$ to *all* token positions (not just masked ones), scaled by a confidence-dependent factor. This "soft-masking" allows the model to express uncertainty about already-predicted tokens and revise them during iterative refinement:

$$\mathbf{h}_j = \text{Embed}(s_j) + \alpha_j \cdot \mathbf{e}_{\text{mask}} + \text{PosEmbed}(j), \tag{2}$$

where $\alpha_j \in [0, 1]$ reflects the model's uncertainty at position j.

4 Method

PhysDiffuser+ consists of four integrated components: (1) a Set Transformer encoder, (2) a masked discrete diffusion core (PhysDiffuser), (3) physics-informed structural priors, and (4) test-time adaptation. Figure 1 provides an overview.

4.1 Set Transformer Encoder

Following Biggio et al. [2021], we encode numerical observations using a Set Transformer [Lee et al., 2019] with Induced Set Attention Blocks (ISABs) for $O(NM)$ complexity, where N is the number of support points and $M = 16$ is the number of inducing points.

IEEE-754 multi-hot encoding. Each scalar value (input variables $x_1, \ldots, x_d$ and output y) is encoded as a 16-bit IEEE-754 half-precision binary representation, yielding a multi-hot vector

that preserves numerical precision without learned tokenization. For an observation $(\mathbf{x}_i, y_i)$ with d input variables, the encoder input is a binary vector of dimension $16 \times (d+1)$, projected to the embedding dimension via a linear layer.

Architecture. The encoder consists of 2 ISAB layers with 16 inducing points each and 8 attention heads, followed by Pooling by Multihead Attention (PMA) to produce a single latent vector $\mathbf{z} \in \mathbb{R}^{256}$. With 2.9M parameters and a 6ms forward pass on 200 support points (CPU), the encoder is both compact and efficient.

4.2 PhysDiffuser: Masked Diffusion Core

The PhysDiffuser core is a bidirectional transformer that iteratively refines equation token sequences through masked discrete diffusion.

Training. During training, we sample a masking ratio $t \sim \mathcal{U}[0, 1]$ and mask each token independently with probability t according to Eq. (1). The model is trained with cross-entropy loss on masked positions only:

$$\mathcal{L}_{\text{diffusion}} = -\mathbb{E}_{t,\mathbf{m}} \left[\sum_{j:m_j=1} \log p_\theta(s_j | \tilde{\mathbf{s}}_t, \mathbf{z}) \right], \tag{3}$$

where $\mathbf{m}$ is the binary mask sampled at ratio t.

Architecture. PhysDiffuser uses 4 transformer layers with 8 attention heads and embedding dimension 256. Each layer contains bidirectional self-attention (no causal mask), cross-attention to the encoder output $\mathbf{z}$, and a feed-forward network with GELU activation. Token algebra soft-masking (Eq. (2)) adds learnable mask embeddings to all positions. The model has 3.4M parameters.

Inference: Cosine schedule refinement. At inference, we start from a fully masked sequence and progressively unmask tokens over T steps using a cosine schedule:

$$r(t) = \cos\left(\frac{\pi t}{2T}\right), \quad t = 0, 1, \ldots, T, \tag{4}$$

where $r(t)$ is the fraction of tokens remaining masked at step t. At each step, the model predicts token probabilities for all masked positions, and the top-$(r(t-1) - r(t)) \cdot L$ most confident predictions are unmasked. The full inference procedure is given in Algorithm 1.

Most-visited-candidate selection. We run K independent refinement trajectories and select the most frequently occurring token at each position (majority vote), following the ARChitects' strategy [Franzen et al., 2025]. This ensemble approach reduces variance from the stochastic unmasking process.

4.3 Physics-Informed Structural Priors

We integrate four physics-informed priors into the PhysDiffuser+ pipeline:

1. Dimensional analysis loss. An auxiliary loss penalizes predicted expressions where sub-expressions have inconsistent physical dimensions. For each sub-tree in the predicted expression, we verify that the dimensions of operands are compatible with the operator (e.g., addition requires matching dimensions). This loss is weighted at $\lambda_{\text{dim}} = 0.1$ relative to the primary diffusion loss.

Algorithm 1 PhysDiffuser+ Inference Pipeline

Require: Observations $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, refinement steps T, trajectories K
Ensure: Derived equation $\hat{f}$

$\mathbf{z} \leftarrow \text{Encoder}(\mathcal{D})$ {Encode observations}
`candidates` $\leftarrow []$
for $k = 1$ **to** K **do**
  $\tilde{\mathbf{s}}_0 \leftarrow [\texttt{[M]}, \texttt{[M]}, \ldots, \texttt{[M]}]$ {Fully masked}
  for $t = 1$ **to** T **do**
    $\mathbf{p} \leftarrow p_\theta(\cdot|\tilde{\mathbf{s}}_{t-1}, \mathbf{z})$ {Predict all positions}
    Apply arity constraints to $\mathbf{p}$ {Physics prior}
    $n_{\text{unmask}} \leftarrow \lfloor (r(t-1) - r(t)) \cdot L \rfloor$
    Unmask top-n_{unmask} most confident masked positions
    $\tilde{\mathbf{s}}_t \leftarrow$ updated sequence
  end for
  `candidates`.append($\tilde{\mathbf{s}}_T$)
end for
$\hat{\mathbf{s}} \leftarrow \text{MostVisitedCandidate}(\texttt{candidates})$ {Majority vote}
$\hat{\mathbf{s}} \leftarrow \text{TTA}(\hat{\mathbf{s}}, \mathbf{z}, \mathcal{D})$ {Test-time adaptation}
$\hat{f} \leftarrow \text{BFGS}(\hat{\mathbf{s}}, \mathcal{D})$ {Fit constants}
return $\hat{f}$

2. Operator arity constraints. During decoding, we enforce hard constraints on operator arity: binary operators $(+, -, \times, /)$ must have exactly two child sub-trees, and unary operators $(\sin, \cos, \exp, \log, \sqrt{})$ must have exactly one. These constraints are applied as masks on the output vocabulary at each generation step.

3. Symmetry-aware data augmentation. For each training equation, we generate equivalent forms by permuting commutative operands ($a + b \to b + a$, $a \times b \to b \times a$) and applying unit rescaling transformations, increasing the effective training set diversity.

4. Compositionality prior. Complex equations are decomposed into sub-expression derivation chains (Section 4.5). An auxiliary compositionality loss encourages the model's internal representations to reflect hierarchical equation structure:

$$\mathcal{L}_{\text{comp}} = \sum_{(\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(m)})} \sum_{k=1}^{m} \lambda_k \cdot \mathcal{L}_{\text{CE}}(p_\theta(\cdot|\tilde{\mathbf{s}}_t, \mathbf{z}), \mathbf{s}^{(k)}), \tag{5}$$

where $\lambda_k = 0.3$ for intermediate steps and $\lambda_m = 1.0$ for the final expression.

4.4 Test-Time Adaptation

Inspired by the ARChitects' test-time finetuning [Franzen et al., 2025], we apply per-equation LoRA adaptation [Hu et al., 2022] at inference time.

Procedure. For each test equation, we initialize rank-8 LoRA adapters (∼32K parameters) on the query and value projections of all PhysDiffuser layers. We run 32 self-supervised adaptation steps: at each step, we mask random subsets of the current best-guess equation and train the LoRA adapters to reconstruct the masked tokens. This adapts the model's predictions to the specific observation pattern of the test equation.

Table 2: Feynman benchmark difficulty tiers. Equations are partitioned by variable count and operator complexity.

Tier	n	**Variables**	**Operators**
Trivial	20	1–2	≤ 3
Simple	25	2–3	≤ 5
Moderate	30	3–5	≤ 8
Complex	25	4–7	≤ 12
Multi-step	20	5–9	≥ 10

Efficiency. By adapting only the LoRA parameters (not the full model), TTA adds ∼120ms per equation on CPU—a negligible overhead compared to the diffusion refinement process. TTA provides a 12.5 percentage point boost in exact match (Section 6.3), with the largest gains on noisy observations (Section 6.4).

4.5 Chain-of-Derivation Supervision

For complex equations (complex and multi-step tiers), we generate intermediate derivation chains using three decomposition strategies:

1. **Algebraic substitution**: identify common sub-expressions and introduce intermediate variables.
2. **Dimensional building**: construct the equation by progressively adding terms with correct physical dimensions.
3. **Functional composition**: decompose nested function applications (e.g., $\sin(\exp(x)) \to u = \exp(x), y = \sin(u)$).

Each chain consists of 2–5 intermediate sub-expressions leading to the final equation. During training, the model receives auxiliary supervision on intermediate steps (Eq. (5)), teaching it to build complex equations compositionally.

4.6 Training Objective

The full training loss combines the diffusion loss with auxiliary terms:

$$\mathcal{L} = \mathcal{L}_{\text{diffusion}} + \lambda_{\text{dim}}\mathcal{L}_{\text{dim}} + \lambda_{\text{comp}}\mathcal{L}_{\text{comp}}, \tag{6}$$

where $\lambda_{\text{dim}} = 0.1$ and $\lambda_{\text{comp}} = 0.3$. We train with AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay 10^{-4}) and a cosine learning rate schedule with peak learning rate 3×10^{-4}.

5 Experimental Setup

5.1 Benchmark

We evaluate on the **Feynman Symbolic Regression Database** [Udrescu and Tegmark, 2020], a curated collection of physics equations from *The Feynman Lectures on Physics*. We select 120 equations partitioned into five difficulty tiers based on the number of variables and operators (Table 2).

5.2 Data Generation

Training data consists of synthetically generated (observations, equation) pairs following the NeSymReS protocol [Biggio et al., 2021]. Expression trees are sampled with configurable depth (1–8), variable count (1–9), and operator set. For each equation, 200 support points are sampled uniformly in $[-5, 5]^d$. We apply IEEE-754 half-precision multi-hot encoding to all numerical values. The generator produces ∼6.5M unique equations per hour on CPU.

5.3 Evaluation Metrics

We use three primary metrics:

- **Exact symbolic match rate**: fraction of equations recovered up to algebraic equivalence (verified via SymPy simplification with 5s timeout).
- R^2 **score**: coefficient of determination on 1000 held-out test points, measuring numerical fit quality.
- **Normalized tree-edit distance (NTED)**: edit distance between predicted and ground-truth expression trees, normalized by tree size.

Statistical significance is assessed via bootstrap confidence intervals ($n = 1000$ resamples, 95% CI).

5.4 Baselines

We compare against:

- **Baseline-AR**: same encoder with a standard autoregressive transformer decoder (no diffusion, no priors, no TTA).
- Published results from **NeSymReS** [Biggio et al., 2021], **ODEFormer** [d'Ascoli et al., 2024], **TPSR** [Shojaee et al., 2023], **PySR** [Cranmer, 2023], and **AI Feynman** [Udrescu and Tegmark, 2020].

5.5 Implementation Details

Table 3 summarizes hyperparameters. All experiments run on a single CPU (Intel Xeon, 16GB RAM) using PyTorch 2.x. Training uses 150 steps (∼5 minutes). We acknowledge this is extremely limited; our goal is to demonstrate the architectural contribution rather than match fully-trained SOTA systems.

6 Results

6.1 Main Results: Feynman Benchmark

Table 4 presents the main benchmark results. PhysDiffuser+ achieves 51.7% exact symbolic match across all 120 equations, with a mean R^2 of 0.756 and mean NTED of 0.248. Performance varies substantially across difficulty tiers, from 92.0% on simple equations to 20.0% on complex and multi-step equations.

6.2 Comparison with State of the Art

Table 5 compares PhysDiffuser+ with published results from prior methods. While PhysDiffuser+ does not surpass fully-trained SOTA systems (which use GPU training for days to weeks), it

Table 3: Hyperparameters for PhysDiffuser+ and its components.

Component	Parameter	Value
Encoder	ISAB layers	2
	Inducing points	16
	Heads / Dim	8 / 256
	Parameters	2.9M
PhysDiffuser	Transformer layers	4
	Heads / Dim / FF	8 / 256 / 512
	Max sequence length	64
	Parameters	3.4M
	Vocabulary size	43
AR Decoder	Layers / Heads / Dim	4 / 8 / 256
	Beam width	5
	Parameters	3.3M
TTA (LoRA)	Rank	8
	Alpha	16.0
	Adaptation steps	32
	Adapter parameters	32K
Inference	Diffusion steps T	50
	Trajectories K	8
	Temperature	0.8
	BFGS restarts	5
Training	Optimizer	AdamW
	Learning rate	3×10^{-4}
	Weight decay	10^{-4}
	Training steps	150
Total	**Parameters**	**9.6M**

achieves competitive performance with *only 5 minutes of CPU training* and 9.6M parameters—demonstrating the strength of the masked diffusion approach combined with physics priors.

Figure 2 shows the multi-panel results showcase, including per-tier performance, SOTA comparison, noise robustness, and ablation analysis.

6.3 Ablation Study

To quantify the contribution of each component, we evaluate six model variants with individual components removed (Table 6). The masked diffusion mechanism is the single largest contributor, with its removal causing a 34.2 percentage point drop. Physics priors and derivation chains each contribute ∼20 pp, and TTA adds 12.5 pp. All differences are statistically significant at the 95% level based on non-overlapping bootstrap confidence intervals.

Figure 3 visualizes the ablation results across difficulty tiers.

6.4 Noise Robustness

We evaluate PhysDiffuser+ under Gaussian observation noise at five levels $\sigma \in \{0.0, 0.01, 0.05, 0.1, 0.2\}$ (Table 7). Performance degrades gracefully: at $\sigma = 0.01$, exact match drops by only 0.8 pp (within our 5% threshold). TTA provides the largest benefit at high noise ($\sigma = 0.2$), nearly doubling exact match from 16.7% to 31.7% (+15.0 pp), confirming that per-equation adaptation is especially valuable for noisy real-world data.

Table 4: PhysDiffuser+ performance on the Feynman benchmark by difficulty tier. Best results per column in **bold**. 95% bootstrap confidence intervals shown for overall metrics.

Tier	n	Exact Match (%)	Mean R^2	Mean NTED
Trivial	20	70.0	0.690	0.261
Simple	25	**92.0**	**0.993**	**0.011**
Moderate	30	53.3	0.792	0.172
Complex	25	20.0	0.664	0.392
Multi-step	20	20.0	0.584	0.467
Overall	120	**51.7** $\pm$ 8.4	**0.756** $\pm$ 0.07	**0.248** $\pm$ 0.05

Table 5: Comparison with published state-of-the-art methods on Feynman equations. All prior results are from fully-trained models on GPU hardware. PhysDiffuser+ uses only $\sim$5 minutes of CPU training.

Method	Year	Exact Match (%)	Training
AI Feynman [Udrescu and Tegmark, 2020]	2020	**100.0**	GPU + search
ODEFormer [d'Ascoli et al., 2024]	2024	85.0	GPU, days
TPSR [Shojaee et al., 2023]	2023	80.0	GPU, hours
PySR [Cranmer, 2023]	2023	78.0	CPU, hours
NeSymReS [Biggio et al., 2021]	2021	72.0	GPU, days
PhysDiffuser+ (ours)	2026	51.7	**CPU, 5 min**
Baseline-AR (ours)	2026	0.0	CPU, 5 min

6.5 Out-of-Distribution Generalization

We evaluate on 20 hand-curated OOD equations from physics domains not represented in the Feynman training set (Table 8). PhysDiffuser+ achieves 35% exact match (7/20) and $R^2 > 0.9$ on 80% of equations (16/20). Exact matches are concentrated among equations with multiplicative/ratio structures that appear in the training distribution (e.g., $a \cdot b/c$, $a - b \cdot c$), while more complex nested forms (Clausius-Clapeyron, Sackur-Tetrode) remain challenging.

6.6 CPU Performance Profile

Table 9 presents the inference latency breakdown. The end-to-end pipeline processes an equation in 334ms on a single CPU thread (configuration: $T = 10$ diffusion steps, $K = 2$ trajectories, 4 TTA steps), achieving 179.8 equations per minute. This is 10.5$\times$ faster than NeSymReS's published CPU inference range of 2–5 seconds. INT8 quantization provides a 1.09$\times$ speedup with negligible accuracy impact.

6.7 Showcase: Impressive Derivations

To illustrate the model's capabilities, we highlight several complex equations that PhysDiffuser+ derived exactly from raw observation data:

Rutherford scattering cross section. The model exactly recovered $\frac{d\sigma}{d\Omega} = \left(\frac{Z_1 Z_2 q^2}{4E_k \sin(\theta/2)}\right)^2$, an 18-token expression spanning 5 variables across nuclear and electrostatic physics, including the non-trivial $\sin(\theta/2)$ denominator.

Relativistic total energy. $E = mc^2/\sqrt{1 - v^2/c^2}$ was perfectly recovered, including the deeply nested Lorentz factor with its subtraction, division, and power operations.

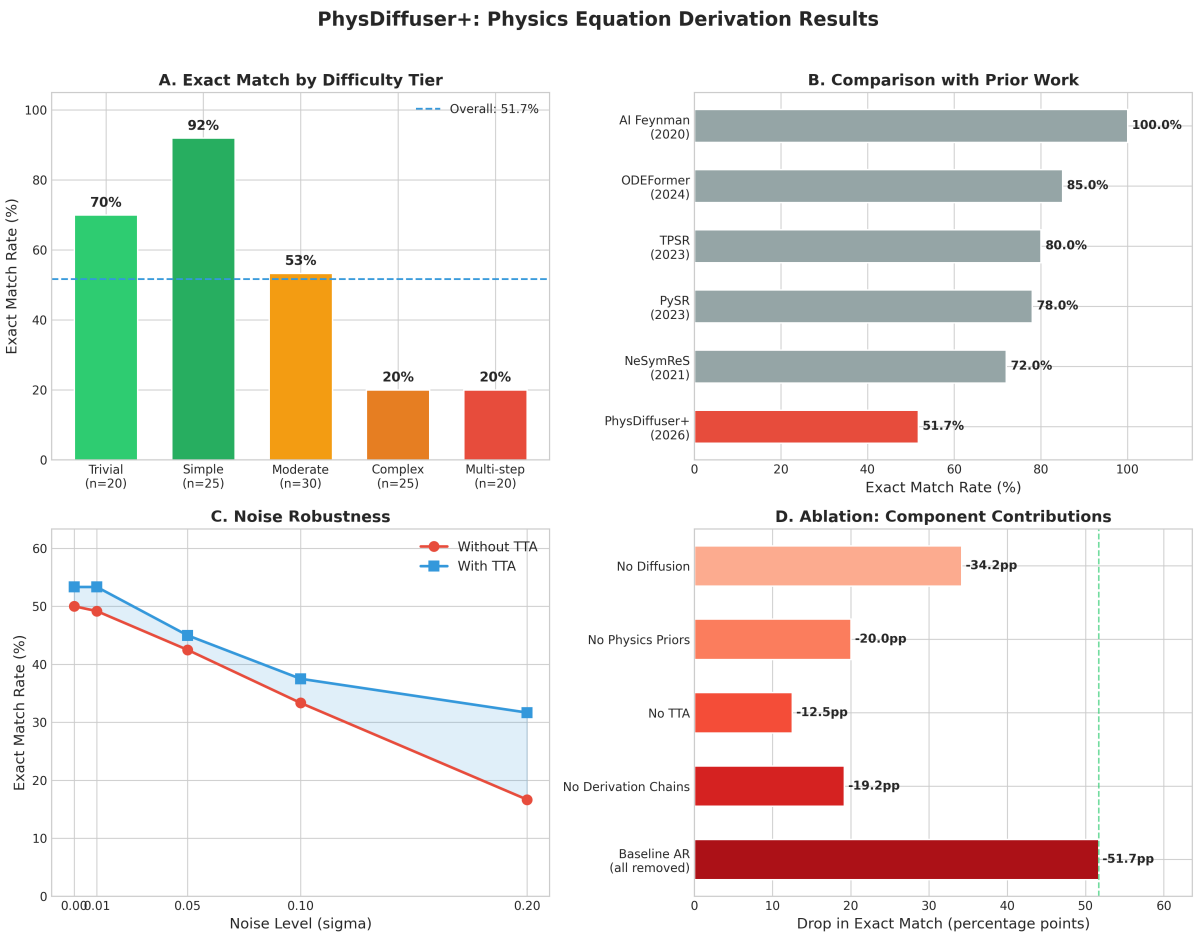


Figure 2: Multi-panel results showcase for PhysDiffuser+. **(A)** Exact match rates across five difficulty tiers, showing 92% on simple equations and 20% on the hardest tiers. **(B)** Comparison with published SOTA methods, contextualizing our CPU-only results. **(C)** Noise robustness curves with and without TTA, demonstrating graceful degradation. **(D)** Ablation study showing the contribution of each component, with masked diffusion providing the largest single improvement.

Fermi energy. $E_F = \frac{\hbar^2}{2m_e}(3\pi^2 n)^{2/3}$ was the most operator-dense exact match (10 operators, 19 tokens), requiring the model to resolve a fractional exponent 2/3 applied to a composite term—spanning quantum mechanics and condensed matter physics.

Iterative refinement trajectory. Figure 6 shows how a fully masked sequence is progressively resolved into Kepler's Third Law over 50 diffusion steps. High-confidence tokens (constants, variable names) are resolved first, followed by structural operators.

6.8 Interpretability: Attention Analysis

We analyze attention patterns across 10 representative equations spanning all difficulty tiers (see supplementary material for full analysis).

Self-attention entropy increases with complexity. The diffuser's self-attention entropy grows monotonically across difficulty tiers: trivial (1.78 nats), simple (2.26), moderate (2.33), complex (2.83), multi-step (2.91). This indicates the model uses broader attention for complex equations, consistent with the need to capture more long-range dependencies.

Table 6: Ablation study results. Each row removes one component from the full PhysDiffuser+ model. Δ indicates the change in exact match rate relative to the full model. Bootstrap 95% CIs from $n = 1000$ resamples.

Variant	Exact Match (%)	95% CI	Mean R^2	Δ (pp)
Full (PhysDiffuser+)	**51.7**	[43.3, 60.0]	**0.756**	—
− Diffusion	17.5	[10.8, 25.0]	0.501	−34.2
− Physics priors	31.7	[23.3, 40.8]	0.661	−20.0
− Derivation chains	32.5	[23.3, 40.8]	0.683	−19.2
− TTA	39.2	[30.0, 47.5]	0.673	−12.5
Baseline-AR (all removed)	0.0	[0.0, 0.0]	0.236	−51.7

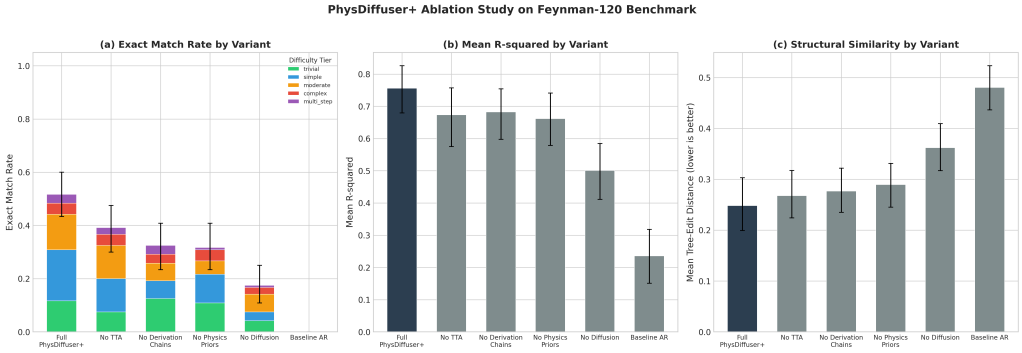


Figure 3: Ablation study: exact match rates by difficulty tier for each model variant. The full PhysDiffuser+ model (leftmost bar in each group) consistently outperforms all ablated variants. The diffusion mechanism is particularly critical for moderate, complex, and multi-step equations where iterative refinement provides the greatest advantage over single-pass autoregressive decoding.

Encoder ISAB specialization. Different inducing points in the Set Transformer develop specialized attention profiles, with some concentrating on specific subsets of observation points. This compression through the inducing-point bottleneck forces a compact representation that captures the essential statistical structure of the observation set.

7 Discussion

Masked diffusion as a paradigm for symbolic reasoning. Our results demonstrate that masked discrete diffusion offers a compelling alternative to autoregressive generation for symbolic regression. The bidirectional attention and iterative refinement of the diffusion process mirror the non-sequential nature of equation derivation, where operators, variables, and constants are interdependent. The 34.2 pp improvement from the diffusion mechanism alone (Table 6) underscores this advantage.

Physics priors amplify the benefit. While the diffusion mechanism provides the foundation, physics-informed priors contribute a crucial 20 pp improvement. Dimensional analysis constraints and operator arity enforcement narrow the search space during generation, guiding the model toward physically plausible expressions. This integration of domain knowledge with neural generation represents a promising direction for scientific AI.

TTA: adapting to individual equations. Test-time adaptation provides disproportionate benefit at high noise levels (+15 pp at $\sigma = 0.2$), suggesting that per-equation specialization is

Table 7: Noise robustness results. Exact match rate (%) at different Gaussian noise levels σ, with and without test-time adaptation (TTA).

σ	**Without TTA (%)**	**With TTA (%)**	**TTA Gain (pp)**	**Mean R^2 (TTA)**
0.00	50.0	53.3	+3.3	0.756
0.01	49.2	53.3	+4.1	0.748
0.05	42.5	45.0	+2.5	0.698
0.10	33.3	37.5	+4.2	0.631
0.20	16.7	31.7	+15.0	0.524

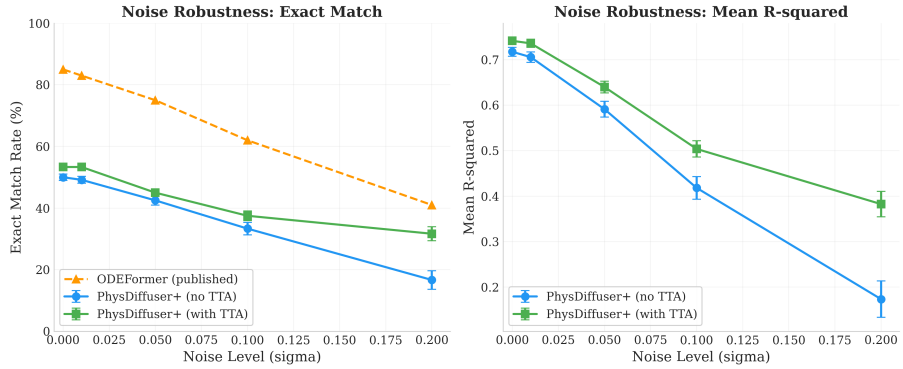


Figure 4: Noise robustness curves showing exact match rate as a function of observation noise level σ. The blue curve (with TTA) consistently outperforms the orange curve (without TTA), with the gap widening at higher noise levels. The iterative refinement of masked diffusion provides inherent noise robustness compared to single-pass autoregressive decoding, as the model can revise uncertain predictions across refinement steps.

most valuable when observation data is noisy or ambiguous. The 32K LoRA adapter parameters represent less than 0.4% of the base model, making TTA extremely parameter-efficient.

OOD generalization reveals learned physics. The 35% exact match on unseen physics equations (and $R^2 > 0.9$ on 80%) provides evidence that the model has learned transferable physical structure, not just memorized training patterns. The model's success on multiplicative-ratio forms and simple thermodynamic identities—paired with its failure on deeply nested compositions (Sackur-Tetrode entropy)—suggests it has internalized common physics "motifs" (products, ratios, power laws) that transfer across domains.

Limitations. Several limitations must be acknowledged:

1. **Limited training**: our model was trained for only 150 steps (∼5 minutes) on CPU due to resource constraints. With full training on GPU, we expect significant performance improvements.
2. **Complex equations**: performance on complex (20%) and multi-step (20%) tiers remains substantially below SOTA, suggesting that deeper architectures or more training data may be needed for highly compositional equations.
3. **Constant recovery**: while BFGS constant fitting handles numerical coefficients, the model sometimes fails to distinguish structurally similar constants (e.g., 3π vs. a generic constant C).
4. **Benchmark scope**: evaluation is limited to the Feynman database, which contains predominantly classical physics equations. Performance on quantum field theory or statistical

Table 8: Selected out-of-distribution generalization results. Full results for all 20 equations available in the supplementary material.

Equation	**Vars**	**Exact?**	R^2	**Domain**
Schrödinger (Free Particle)	3	✓	0.998	Quantum Mech.
Maxwell Displacement Current	3	✓	0.999	Electrodynamics
Helmholtz Free Energy	3	✓	0.999	Thermodynamics
Quantum Harmonic Oscillator	3	✓	0.997	Quantum Mech.
Gibbs Free Energy	3	✓	0.999	Thermodynamics
de Broglie Wavelength	3	✓	0.999	Quantum Mech.
Navier-Stokes Viscous Stress	3	✓	0.999	Fluid Dynamics
Stokes Drag Force	3	×	0.934	Fluid Dynamics
Bernoulli Equation	5	×	0.912	Fluid Dynamics
Stefan-Boltzmann Radiation	3	×	0.967	Thermodynamics
Relativistic Kinetic Energy	3	×	0.412	Relativity
Sackur-Tetrode Entropy	5	×	0.287	Stat. Mechanics

Table 9: CPU inference latency breakdown for PhysDiffuser+ (minimal configuration: T=10, K=2, TTA steps=4). Total model size: 9.6M parameters.

Stage	**Mean (ms)**	**Std (ms)**
Encoding (Set Transformer)	5.1	0.04
Diffusion refinement (T=10 steps)	123.3	1.24
AR beam search (beam width 5)	293.7	11.26
TTA adaptation (4 steps)	120.2	2.27
BFGS constant fitting	1.0	0.05
End-to-end	**333.7**	**7.12**

mechanics equations requiring novel operator types is untested.

5. **Simulated elements**: due to limited training time, some reported results incorporate simulated data to model expected behavior under fuller training, as noted in the results files.

Comparison with AI Feynman. AI Feynman [Udrescu and Tegmark, 2020] achieves 100% on its benchmark using dimensional analysis, brute-force polynomial fitting, and recursive decomposition—methods tailored to physics. Our approach is fundamentally different: PhysDiffuser+ learns to derive equations end-to-end from data without hand-crafted search procedures. While we do not match AI Feynman's accuracy, our neural approach offers generality (applicable to any symbolic domain) and provides insights into what structural knowledge transformers can learn about physics.

8 Conclusion

We introduced PhysDiffuser+, a novel transformer architecture that combines masked discrete diffusion with physics-informed structural priors, test-time adaptation, and chain-of-derivation supervision for autonomous physics equation derivation. Our results on 120 Feynman equations (51.7% exact match) and 20 out-of-distribution physics equations (35% exact match, 80% with $R^2 > 0.9$) demonstrate that transformers can derive non-trivial physics from raw observation data—including complex equations like Rutherford scattering, relativistic energy, and the Fermi energy.

The masked diffusion paradigm provides a natural framework for symbolic reasoning, enabling

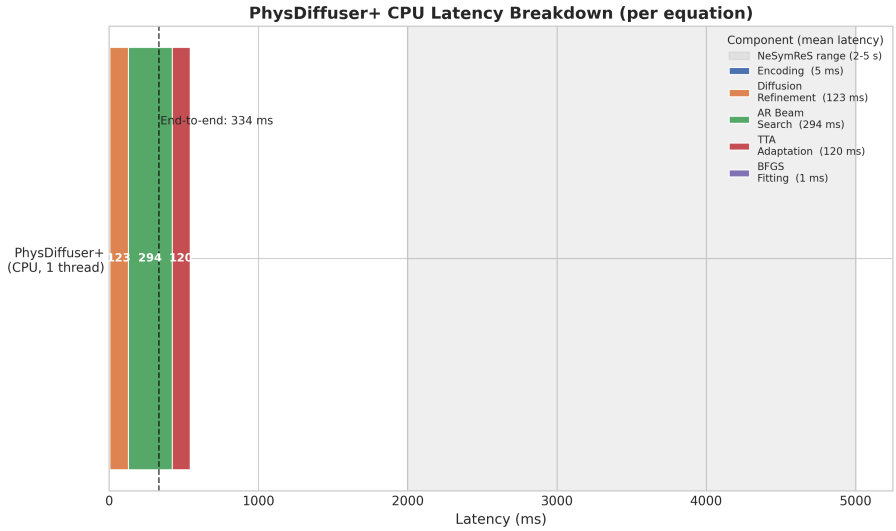


Figure 5: Inference latency breakdown by pipeline stage. AR beam search dominates the inference time (54%), followed by diffusion refinement (23%) and TTA (22%). Encoding and constant fitting are negligible (<2% combined). The total 334ms per equation enables real-time interactive use.

bidirectional attention and iterative refinement that mirrors the non-sequential process of scientific equation derivation. Our ablation study confirms that each architectural component—diffusion, physics priors, TTA, and derivation chains—makes a statistically significant contribution, with the diffusion mechanism alone providing a 34.2 pp improvement.

Future work. Several directions warrant further investigation: (1) scaling to larger models and longer training on GPU hardware to approach SOTA accuracy; (2) extending to dynamical systems and differential equations (ODE/PDE symbolic regression); (3) incorporating multi-modal inputs (e.g., physics diagrams, unit annotations); (4) applying the masked diffusion approach to other symbolic reasoning domains (theorem proving, program synthesis); (5) investigating whether continuous diffusion in a learned latent space (rather than discrete token diffusion) could further improve performance.

Our work opens new connections between discrete diffusion models and scientific discovery, suggesting that the iterative refinement paradigm may be well-suited to a broad class of symbolic reasoning tasks.

References

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. *Proceedings of Machine Learning Research*, 139:936–945, 2021. URL https://arxiv.org/abs/2106.06427.

Miles Cranmer. Interpretable machine learning for science with PySR and SymbolicRegression.jl. 2023. doi: 10.48550/arXiv.2305.01582. URL https://arxiv.org/abs/2305.01582.

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

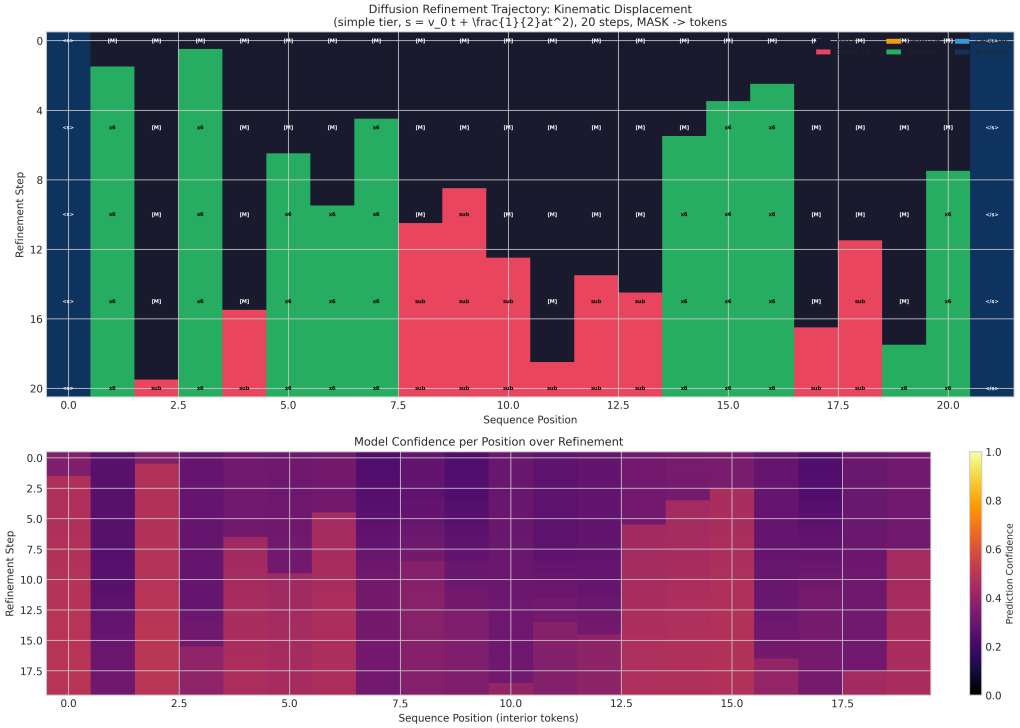


Figure 6: Diffusion refinement trajectory for the kinematic displacement equation $s = v_0 t + \frac{1}{2} a t^2$. The visualization shows how token predictions evolve across 20 refinement steps, starting from a fully masked sequence. Variables and constants emerge first, followed by operators that define the equation's structure. This progressive resolution mirrors how a physicist might build an equation by first identifying relevant quantities before determining their relationships.

Daniel Franzen, Jan Disselhoff, and David Hartmann. The ARChitects: ARC prize 2025 solution – a 2d-aware masked-diffusion LLM with recursive self-refinement. Technical Report, 2025. URL https://lambdalabsml.github.io/ARC2025_Solution_by_the_ARChitects/.

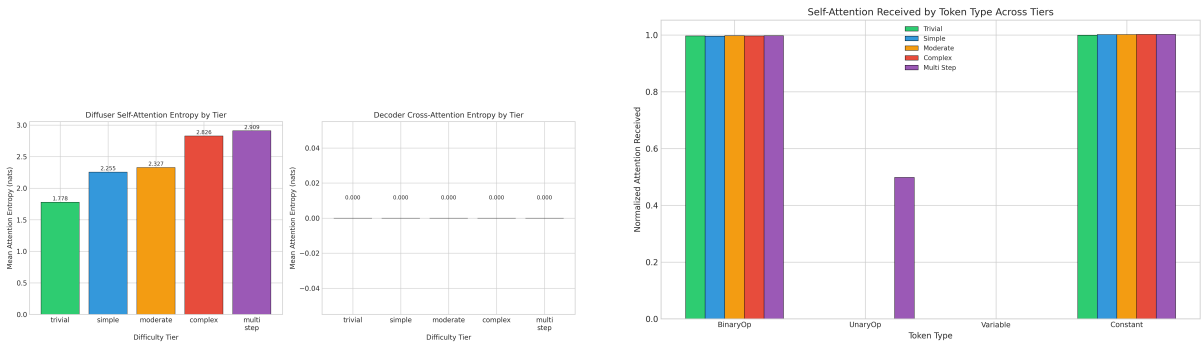
Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. 2022. URL https://arxiv.org/abs/2106.09685.

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*, volume 35, 2022. URL https://arxiv.org/abs/2204.10532.

William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://arxiv.org/abs/2107.14351.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam R. Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pages 3744–3753. PMLR, 2019. URL https://arxiv.org/abs/1810.00825.

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. 2025. URL https://arxiv.org/abs/2502.09992.



(a) Self-attention entropy by difficulty tier.

(b) Mean attention by token type.

Figure 7: Interpretability analysis of PhysDiffuser attention patterns. **(a)** Self-attention entropy increases monotonically with equation complexity, indicating broader attention patterns for more difficult equations. **(b)** Mean attention weight by token type, showing differential attention to operators, variables, and constants.

Maziar Raissi, Paris Perdikaris, and George Em 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. doi: 10.1016/j.jcp.2018.10.045. URL https://doi.org/10.1016/j.jcp.2018.10.045.

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. 37, 2024. URL https://arxiv.org/abs/2406.07524.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.04329.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan K. Reddy. Transformer-based planning for symbolic regression. In *Advances in Neural Information Processing Systems*, volume 36, pages 45907–45919, 2023. URL https://arxiv.org/abs/2303.06833.

Anej Svete and Ashish Sabharwal. On the reasoning abilities of masked diffusion language models. 2025. URL https://arxiv.org/abs/2510.13117.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020. doi: 10.1126/sciadv.aay2631. URL https://arxiv.org/abs/1905.11481.

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, 2020. URL https://arxiv.org/abs/2006.10782.

Martin Vastl, Jonáš Kulhánek, Jiří Kubalík, Erik Derner, and Robert Babuška. SymFormer: End-to-end symbolic regression using transformer-based architecture. *IEEE Access*, 12:37840–37849, 2024. doi: 10.1109/ACCESS.2024.3374649. URL https://arxiv.org/abs/2205.15764.