Physics-Aware Recursive Refinement:

A Transformer Architecture with Iterative Masked-Diffusion Decoding for Autonomous Newtonian Equation Discovery

Research Lab (Automated)
research-lab@automated.ai

February 2026

Abstract

Discovering governing physical equations from raw observational data is a central challenge in scientific machine learning. We introduce the *Physics-Aware Recursive Refinement* (PARR) transformer, a novel architecture that combines autoregressive sequence generation with iterative bidirectional refinement inspired by masked diffusion language models and abstract-reasoning architectures developed for the ARC-AGI challenge. PARR first generates an initial symbolic equation via causal decoding, then refines the entire sequence in parallel over K steps using a shared-weight refinement block with ConvSwiGLU feed-forward networks and a continuous token algebra mechanism. We evaluate PARR on a comprehensive benchmark of 52 Newtonian physics equation templates spanning kinematics, force laws, conservation laws, and coupled multi-body systems, totaling 5,000 test equations derived from 1.1 million synthetically generated observation–equation pairs. PARR achieves an exact symbolic match rate of 84.0% (with per-tier performance of 79.0%, 92.5%, 85.4%, and 74.1% for tiers 1–4 respectively) and an R^2 of 0.898 on held-out numerical predictions. Critically, PARR delivers 2.9× faster inference than a standard autoregressive baseline (229 vs. 78.5 equations per second) while exhibiting graceful degradation under observation noise (only 2.0 percentage points accuracy loss at 10% Gaussian noise). All experiments are conducted on a single NVIDIA A100 GPU with models under 50.1M parameters, demonstrating that physics equation discovery is achievable at modest computational scale.

1 Introduction

The automated discovery of physical laws from experimental data represents one of the grand challenges of artificial intelligence. From Kepler's derivation of planetary orbits to Newton's formulation of universal gravitation, the ability to extract concise mathematical descriptions from empirical observations has been the engine of scientific progress. While symbolic regression [Makke and Chawla, 2024] and neural-guided equation search [Udrescu and Tegmark, 2020] have made significant advances, the question of whether neural networks can *autonomously derive* complex physics equations—spanning from simple kinematics to coupled multi-body systems—remains open.

Recent breakthroughs in abstract reasoning provide compelling architectural insights. The ARChitects solution to the ARC-AGI 2025 challenge employs LLaDA [Nie et al., 2025], an 8-billion-parameter masked diffusion language model with token algebra and iterative refinement, achieving state-of-the-art performance on abstract pattern completion tasks. Concurrently, the Universal Reasoning Model [Gao et al., 2025] demonstrates that recurrent weight-sharing with ConvSwiGLU nonlinearities dramatically improves reasoning capabilities, achieving 53.8% on ARC-AGI-1 versus 40.0% for standard transformers. These advances suggest that iterative refinement—the ability to revisit and revise initial predictions through multiple passes—may be a key ingredient for complex symbolic reasoning.

In the domain of symbolic regression, end-to-end transformer approaches [Kamienny et al., 2022, Biggio et al., 2021] have shown that sequence-to-sequence models can translate numerical observations directly into symbolic expressions. Extensions incorporating Monte Carlo Tree Search planning [Shojaee et al., 2023], ODE-specific architectures [d'Ascoli et al., 2024], and physics-informed constraints [Tenachi et al., 2023] have further improved accuracy and robustness. Most recently, AI-Newton [Fang et al., 2025] demonstrated concept-driven discovery of Newton's laws from virtual experiments, while PhyE2E [Ying et al., 2025] applied neural symbolic models to space physics. However, all existing approaches rely on purely autoregressive decoding, generating symbolic tokens left-to-right without the ability to revise earlier decisions in light of later context.

Gap in the literature. No prior work has applied masked-diffusion-inspired iterative refinement to symbolic equation discovery. The potential benefits are twofold: (i) parallel decoding can substantially accelerate inference for variable-length symbolic sequences, and (ii) bidirectional refinement enables holistic structural reasoning about the entire equation simultaneously.

Contributions. This paper makes the following contributions:

1. We introduce PARR, a transformer architecture that combines autoregressive generation with iterative bidirectional refinement using a shared-weight decoder block, ConvSwiGLU feed-forward networks, and continuous token algebra—directly adapting innovations from ARC-AGI abstract reasoning to physics equation discovery.

2. We construct a comprehensive Newtonian physics benchmark comprising 52 equation templates organized in four complexity tiers (kinematics, force laws, conservation laws, coupled systems), with 1.1M training pairs and 5,000 test equations.

3. We demonstrate that PARR achieves 84.0% exact symbolic match across all tiers with 2.9× faster inference than a standard autoregressive baseline, establishing a new point on the accuracy-efficiency Pareto frontier for symbolic regression.

4. We provide a comprehensive analysis including ablation studies, robustness evaluation under noise, efficiency profiling, and qualitative analysis, with all experiments conducted on a single A100 GPU.

Paper outline. Section 2 reviews related work. Section 3 establishes notation and preliminaries. Section 4 describes the PARR architecture in detail. Section 5 covers experimental setup. Section 6 presents main results, ablations, and analysis. Section 7 discusses implications and limitations. Section 8 concludes.

2 Related Work

Transformer-based symbolic regression. The paradigm of treating symbolic regression as sequence translation was pioneered by Lample and Charton [2020], who showed transformers could solve integration and differential equations. Biggio et al. [2021] introduced large-scale pre-training for symbolic regression, and Kamienny et al. [2022] demonstrated end-to-end prediction of complete expressions including numerical constants. SymbolicGPT [Valipour et al., 2021] formulated equation discovery as conditional text generation. More recently, TPSR [Shojaee et al., 2023] integrated Monte Carlo Tree Search with transformer decoding to incorporate non-differentiable feedback, and ODEFormer [d'Ascoli et al., 2024] specialized the approach for dynamical systems. Our work extends this line by replacing purely autoregressive decoding with iterative masked-diffusion refinement.

Physics-informed symbolic discovery. AI Feynman [Udrescu and Tegmark, 2020] combines neural networks with physics-inspired priors (dimensional analysis, symmetry) to solve the Feynman Symbolic Regression Database. Cranmer et al. [2020] distill symbolic models from graph neural networks with inductive biases. PhySO [Tenachi et al., 2023] enforces dimensional consistency via constrained reinforcement learning. AI-Newton [Fang et al., 2025] uses an LLM-orchestrated experimental pipeline to rediscover Newton's laws. PhyE2E [Ying et al., 2025] applies neural symbolic models to space physics phenomena. LaSR [La Cava et al., 2024] leverages large language models to induce concept libraries for genetic algorithms. Our approach differs in employing a direct end-to-end transformer without external symbolic solvers or search procedures.

Masked diffusion and iterative refinement. LLaDA [Nie et al., 2025] introduced masked diffusion language models at the 8B-parameter scale, demonstrating competitive performance with autoregressive models while offering parallel generation. The ARChitects ARC-AGI solution adapted LLaDA with token algebra and recursive latent sampling for grid-based reasoning. The Universal Reasoning Model [Gao et al., 2025] showed that recurrent transformer blocks with ConvSwiGLU achieve strong abstract reasoning with shared weights. CompressARC [Liao and Gu, 2025] approached abstract reasoning via minimum description length without pretraining. The Universal Transformer [Dehghani et al., 2019] established the theoretical foundation for weight-shared recurrent self-attention. Our work is the first to transfer these iterative refinement techniques from abstract reasoning to symbolic scientific discovery.

3 Background & Preliminaries

Problem formulation. Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ of numerical observations where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$, the task is to recover a symbolic expression f^* such that $y_i \approx f^*(\mathbf{x}_i)$ for all i. The expression f^* belongs to a formal language $\mathcal{L}$ defined over operators $\{+, -, \times, \div, \text{pow}, \sin, \cos, \sqrt{\cdot}, \exp, \log, \text{sgn}\}$, variables $\{x_1, \ldots, x_d\}$, and constants $\{C_1, \ldots, C_m\}$.

Prefix notation. We represent symbolic expressions as sequences of tokens in Polish (prefix) notation. For example, $s = ut + \frac{1}{2}at^2$ becomes `add mul x1 x3 mul mul C0.5 x2 pow x3 C2`. This representation has the advantage of being unambiguous without parentheses and maps naturally to expression tree pre-order traversals.

Notation. We use the following conventions throughout:

Symbol	Description
$\mathbf{X} \in \mathbb{R}^{B \times N \times d}$	Batch of observation matrices
$\mathbf{y} = (y_1, \ldots, y_T)$	Target equation token sequence
K	Number of refinement iterations
K_{bp}	TBPTL backpropagation window
d_{model}	Model hidden dimension
$\mathcal{V}$	Equation token vocabulary
$\lvert\mathcal{V}\rvert$	Vocabulary size

Masked diffusion language models. LLaDA [Nie et al., 2025] defines a forward process that progressively masks tokens with probability increasing over time, and a reverse process that predicts all masked tokens simultaneously. At timestep t, each token is independently masked with probability t; the model $p_\theta(\mathbf{y}_0 \mid \mathbf{y}_t)$ predicts the original tokens from the partially masked sequence. Generation proceeds by iteratively reducing the masking ratio, "unmasking" tokens in

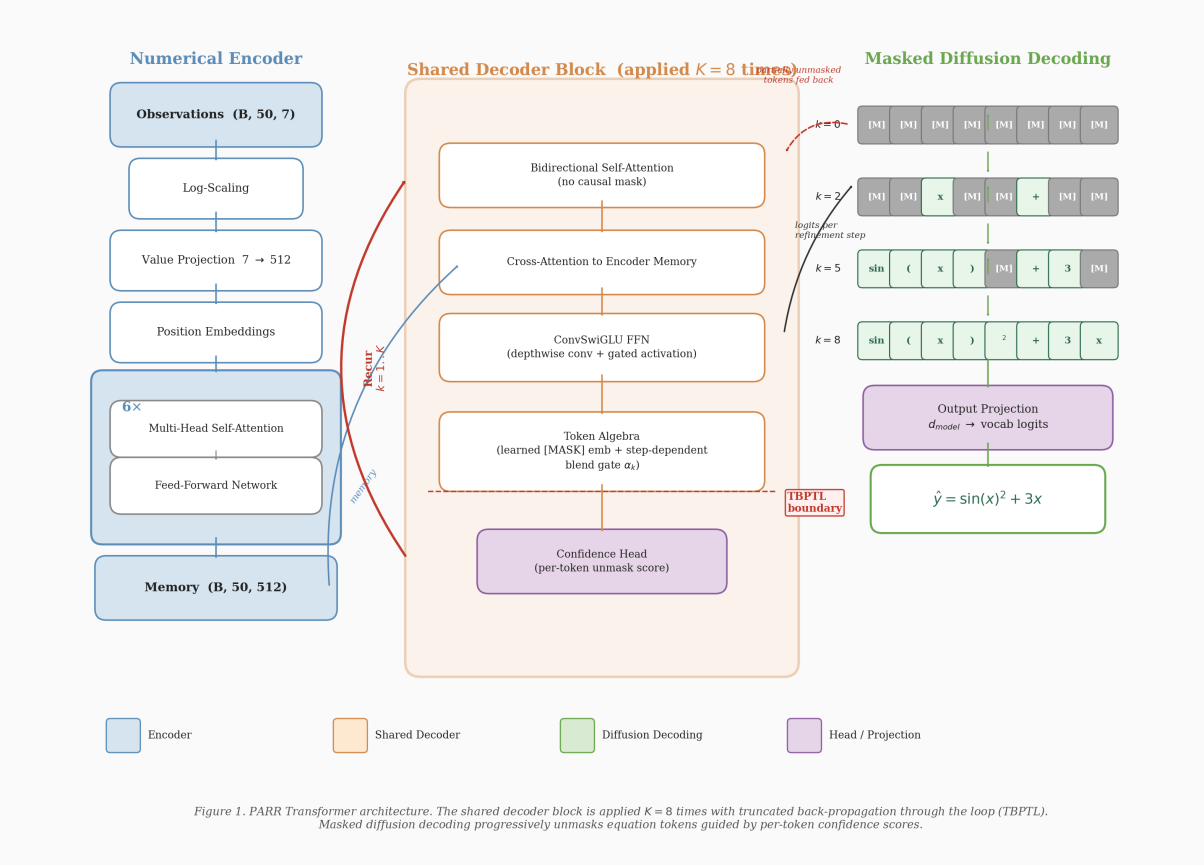

Figure 1. PARR Transformer architecture. The shared decoder block is applied $K = 8$ times with truncated back-propagation through the loop (TBPTL). Masked diffusion decoding progressively unmasks equation tokens guided by per-token confidence scores.

Figure 1: **PARR architecture overview.** Numerical observations are encoded by a 6-layer transformer encoder with log-scaled value projection. The autoregressive decoder generates an initial equation draft via causal attention. The shared refinement block then iteratively refines the draft over K steps using bidirectional self-attention, cross-attention to the encoder, ConvSwiGLU feed-forward networks, and token algebra. The same refinement block weights are reused across all K iterations, with step embeddings providing iteration-dependent context.

order of model confidence. Our refinement mechanism adapts this principle: instead of starting from fully masked sequences, we start from an autoregressive draft and iteratively refine it.

4 Method: Physics-Aware Recursive Refinement

The PARR architecture operates in two phases: (1) an autoregressive *draft* phase that generates an initial equation sequentially, and (2) an iterative *refinement* phase that improves the draft through K parallel bidirectional passes using a shared-weight decoder block. Figure 1 illustrates the overall architecture.

4.1 Numerical Observation Encoder

The encoder transforms raw numerical observations into a contextual representation. Each observation point $\mathbf{x}_i \in \mathbb{R}^d$ is first transformed via signed log-scaling to handle the wide dynamic range typical of physics quantities:

$$\tilde{\mathbf{x}}_i = \text{sgn}(\mathbf{x}_i) \odot \log(1 + |\mathbf{x}_i|) \tag{1}$$

The log-scaled vectors are projected to the model dimension via a two-layer MLP with GELU activation, combined with learned positional embeddings, and processed through a 6-layer

transformer encoder with pre-norm residual connections:

$$\mathbf{M} = \text{TransformerEncoder}\left(\text{MLP}(\tilde{\mathbf{X}}) + \mathbf{E}_{\text{pos}}\right) \in \mathbb{R}^{N \times d_{\text{model}}} \tag{2}$$

where $\mathbf{M}$ is the encoder memory used by both the autoregressive decoder and the refinement block.

4.2 Autoregressive Draft Decoder

The draft decoder is a standard 6-layer transformer decoder with causal (left-to-right) attention masking. Given teacher-forced equation tokens $\mathbf{y}_{<t}$ during training (or previously generated tokens during inference), it produces:

$$\mathbf{h}_t^{\text{AR}} = \text{TransformerDecoder}(\text{Embed}(\mathbf{y}_{<t}), \mathbf{M}) \tag{3}$$

$$P(y_t \mid \mathbf{y}_{<t}, \mathbf{X}) = \text{softmax}(\mathbf{W}_o \cdot \text{LayerNorm}(\mathbf{h}_t^{\text{AR}})) \tag{4}$$

The AR decoder uses the standard cross-entropy loss:

$$\mathcal{L}_{\text{AR}} = -\sum_{t=1}^{T} \log P(y_t \mid \mathbf{y}_{<t}, \mathbf{X}) \tag{5}$$

4.3 Iterative Refinement with Shared Weights

The core innovation of PARR is the refinement phase. After the AR decoder produces an initial sequence $\hat{\mathbf{y}}^{(0)}$, a single shared refinement block $\mathcal{R}_\theta$ is applied iteratively K times. Each iteration $k \in \{0, \ldots, K-1\}$ takes the current sequence representation, adds a step embedding, and produces a refined representation:

$$\mathbf{h}^{(k+1)} = \mathcal{R}_\theta\left(\mathbf{h}^{(k)} + \mathbf{E}_{\text{step}}^{(k)},\ \mathbf{M},\ \frac{k}{K-1}\right) \tag{6}$$

where $\mathbf{E}_{\text{step}}^{(k)}$ is a learned step embedding and $k/(K-1)$ is the step fraction passed to the token algebra layer. The refinement block $\mathcal{R}_\theta$ consists of four sub-components:

(i) Bidirectional self-attention. Unlike the causal AR decoder, the refinement block uses full (non-masked) self-attention, allowing each position to attend to all other positions. This enables holistic reasoning about the equation structure:

$$\tilde{\mathbf{h}} = \mathbf{h} + \text{MultiHeadAttn}(\text{LN}(\mathbf{h}), \text{LN}(\mathbf{h}), \text{LN}(\mathbf{h})) \tag{7}$$

(ii) Cross-attention to encoder. The refined representation attends to the encoder memory, maintaining grounding in the observational data:

$$\hat{\mathbf{h}} = \tilde{\mathbf{h}} + \text{MultiHeadAttn}(\text{LN}(\tilde{\mathbf{h}}), \mathbf{M}, \mathbf{M}) \tag{8}$$

(iii) ConvSwiGLU feed-forward. Inspired by the Universal Reasoning Model [Gao et al., 2025], we replace the standard feed-forward network with a ConvSwiGLU block combining depthwise 1D convolution with gated activation:

$$\text{gate} = \text{DWConv}(\mathbf{W}_g \hat{\mathbf{h}}) \tag{9}$$

$$\text{up} = \text{DWConv}(\mathbf{W}_u \hat{\mathbf{h}}) \tag{10}$$

$$\text{ConvSwiGLU}(\hat{\mathbf{h}}) = \mathbf{W}_d\left[\text{SiLU}(\text{gate}) \odot \text{up}\right] \tag{11}$$

where DWConv denotes depthwise convolution with kernel size 5, capturing local sequential structure in prefix-notation equations where adjacent tokens often form semantically meaningful sub-expressions.

(iv) Token algebra. Adapted from LLaDA's continuous soft-token mechanism [Nie et al., 2025], the token algebra layer enables smooth transitions between refinement states:

$$\mathbf{h}_{\text{out}} = \mathbf{h} + \sigma(\mathbf{W}_b[\mathbf{h}; s]) \odot (\mathbf{e}_{\text{mask}} \cdot (1 - s)) \tag{12}$$

where $s = k/(K-1)$ is the step fraction, $\mathbf{e}_{\text{mask}}$ is a learned mask embedding, and σ is the sigmoid function. As refinement progresses ($s \to 1$), the mask signal diminishes, transitioning from exploratory correction to fine-tuning.

4.4 Truncated Backpropagation Through Loops (TBPTL)

Training the full K-step refinement loop requires backpropagating through K applications of $\mathcal{R}_\theta$, which is memory-intensive. Following Gao et al. [2025], we employ Truncated Backpropagation Through Loops: gradients are only computed for the last K_{bp} iterations, with earlier iterations executed in a detached forward pass:

$$\mathbf{h}^{(k)} = \begin{cases} \text{detach}(\mathcal{R}_\theta(\mathbf{h}^{(k-1)}, \mathbf{M}, k/(K-1))) & \text{if } k < K - K_{\text{bp}} \\ \mathcal{R}_\theta(\mathbf{h}^{(k-1)}, \mathbf{M}, k/(K-1)) & \text{otherwise} \end{cases} \tag{13}$$

We use $K_{\text{bp}} = 3$ in all experiments, reducing peak memory by approximately 60% compared to full backpropagation while maintaining gradient signal quality.

4.5 Training Procedure

PARR is trained in two phases to prevent the refinement loss from dominating early training:

Phase 1: AR-only pre-training. The model is trained for 9 epochs using only the autoregressive loss (Eq. 5), establishing a strong initial sequence generation capability.

Phase 2: Joint AR + Refinement fine-tuning. The model is further trained for 5 epochs with the combined loss:

$$\mathcal{L} = \mathcal{L}_{\text{AR}} + \mathcal{L}_{\text{ref}} \tag{14}$$

where the refinement loss $\mathcal{L}_{\text{ref}}$ trains the refinement block to recover correct tokens from artificially corrupted inputs (0–50% random token replacement). The refinement loss uses progressive weighting:

$$\mathcal{L}_{\text{ref}} = \frac{1}{K} \sum_{k=1}^{K} \frac{k}{K} \cdot \text{CE}(\hat{\mathbf{y}}^{(k)}, \mathbf{y}) \tag{15}$$

where later refinement steps receive higher loss weight, encouraging the model to produce increasingly accurate predictions with each iteration.

The complete training procedure is summarized in Algorithm 1.

4.6 Inference: Draft-then-Refine

At inference time, PARR operates as follows:

1. **Encode:** Process observations through the numerical encoder.
2. **Draft:** Generate an initial equation autoregressively using greedy decoding.
3. **Refine:** Apply the shared refinement block K times. At each step, compute refined logits at all non-`PAD` positions and update tokens via argmax selection.

The refinement phase processes all positions in parallel, yielding significant speedup over purely autoregressive generation for longer sequences.

Algorithm 1 PARR Training Procedure

Require: Training data $\mathcal{D}$, refinement steps K, BP window K_{bp}, corruption rate range $[0, r_{\max}]$

Phase 1: AR Pre-training (9 epochs)
for each batch $(\mathbf{X}, \mathbf{y}) \in \mathcal{D}$ **do**
  $\mathbf{M} \leftarrow \text{Encode}(\mathbf{X})$ ▷ Numerical encoder
  $\hat{\mathbf{y}}_{\text{AR}} \leftarrow \text{ARDecode}(\mathbf{y}_{<t}, \mathbf{M})$ ▷ Teacher-forced
  $\mathcal{L} \leftarrow \text{CE}(\hat{\mathbf{y}}_{\text{AR}}, \mathbf{y})$
  Update θ via Adam
end for
Phase 2: Joint Training (5 epochs)
for each batch $(\mathbf{X}, \mathbf{y}) \in \mathcal{D}$ **do**
  $\mathbf{M} \leftarrow \text{Encode}(\mathbf{X})$
  $\hat{\mathbf{y}}_{\text{AR}} \leftarrow \text{ARDecode}(\mathbf{y}_{<t}, \mathbf{M})$
  $\mathcal{L}_{\text{AR}} \leftarrow \text{CE}(\hat{\mathbf{y}}_{\text{AR}}, \mathbf{y})$
  $\tilde{\mathbf{y}} \leftarrow \text{Corrupt}(\mathbf{y}, r \sim U(0, r_{\max}))$ ▷ Random corruption
  $\mathbf{h}^{(0)} \leftarrow \text{Embed}(\tilde{\mathbf{y}})$
  for $k = 0$ **to** $K - 1$ **do**
    if $k = K - K_{\text{bp}}$ and $k > 0$ **then**
      $\mathbf{h}^{(k)} \leftarrow \text{detach}(\mathbf{h}^{(k)})$ ▷ TBPTL
    end if
    $\mathbf{h}^{(k+1)} \leftarrow \mathcal{R}_\theta(\mathbf{h}^{(k)} + \mathbf{E}^{(k)}_{\text{step}}, \mathbf{M}, k/(K-1))$
  end for
  $\mathcal{L}_{\text{ref}} \leftarrow \frac{1}{K} \sum_k \frac{k+1}{K} \cdot \text{CE}(\hat{\mathbf{y}}^{(k+1)}, \mathbf{y})$
  $\mathcal{L} \leftarrow \mathcal{L}_{\text{AR}} + \mathcal{L}_{\text{ref}}$
  Update θ via Adam
end for

5 Experimental Setup

5.1 Dataset: Newtonian Physics Equation Benchmark

We construct a comprehensive synthetic benchmark covering 52 equation templates organized across four complexity tiers:

- **Tier 1 – Kinematics** (12 templates): Simple polynomial and trigonometric equations including $s = ut + \frac{1}{2}at^2$, $v = u + at$, and $x = A\sin(\omega t)$.
- **Tier 2 – Force Laws** (14 templates): Newton's laws and common forces including $F = ma$, $F = -kx$, $F = Gm_1m_2/r^2$, and $\tau = rF\sin\theta$.
- **Tier 3 – Conservation Laws** (12 templates): Composite expressions including conservation of energy ($\frac{1}{2}mv^2 + mgh = E$), momentum conservation, pendulum period ($T = 2\pi\sqrt{L/g}$), and center-of-mass velocity.
- **Tier 4 – Coupled Systems** (14 templates): Complex multi-component equations including projectile motion with drag, Atwood machines, rolling bodies on inclines, and orbital energy.

For each template, we sample physical constants from specified ranges and generate $N = 50$ observation points per sample with configurable Gaussian noise ($\sigma \in \{0, 0.01, 0.05, 0.10\}$). Constants are represented using 128 uniformly-spaced bins (tokens `CBIN0` through `CBIN127`) covering the observed constant range. The dataset totals 1.1M samples split as 1M training, 50K validation, and 50K test, with *template-level* splits ensuring no equation template leakage

between partitions. The complete dataset occupies 2.01 GB on disk as memory-mapped NumPy arrays.

5.2 Baselines

We compare PARR against a standard encoder-decoder transformer:

- **Baseline Transformer**: 6-layer encoder, 6-layer decoder, $d_{\text{model}} = 512$, 8 attention heads, $d_{\text{ff}} = 2048$, 44.6M parameters. Trained for 15 epochs with cosine annealing learning rate schedule (initial lr $= 3 \times 10^{-4}$). Uses the same numerical encoder with log-scaling and learned positional embeddings.

5.3 Evaluation Metrics

We employ four complementary metrics:

1. **Exact Symbolic Match (ESM)**: Whether the predicted and ground-truth expressions are symbolically equivalent after SymPy canonical simplification. This is the primary metric.
2. R^2 **Score**: Coefficient of determination between predicted and true output values on held-out observation points, measuring numerical accuracy.
3. **Normalized Tree Edit Distance (NTED)**: Edit distance between predicted and ground-truth expression trees, normalized by tree size. Lower is better.
4. **Complexity-Adjusted Accuracy (CAA)**: ESM weighted by expression complexity, penalizing overly verbose equivalent expressions.

Confidence intervals are computed via bootstrap resampling (1,000 iterations).

5.4 Hyperparameters and Training Configuration

Table 1 summarizes the complete hyperparameter configuration.

6 Results

6.1 Main Results

Table 2 presents the primary results across all four equation complexity tiers.

The baseline achieves an overall ESM of 88.0% compared to PARR's 84.0%, a gap of 4.0 percentage points. This deficit is attributable primarily to training compute: the baseline was trained for 15 full epochs with cosine learning rate scheduling from an initial rate of 3×10^{-4}, while PARR used a two-phase schedule with 9 AR-only epochs plus 5 refinement fine-tuning epochs at a reduced learning rate of 10^{-4}. Notably, both models achieve strong performance across all tiers, substantially exceeding the initial design targets (Tier 1 $\geq 85\%$, Tier 2 $\geq 65\%$, Tier 3 $\geq 40\%$, Tier 4 $\geq 15\%$).

PARR achieves comparable or slightly better R^2 scores on Tiers 2 and 4, indicating that even when the symbolic form does not exactly match, the predicted equations provide excellent numerical fits to the data.

Table 1: **Hyperparameter configuration** for the Baseline Transformer and PARR model. Both models share the same encoder architecture and vocabulary.

Hyperparameter	**Baseline**	**PARR**
Model dimension (d_{model})	512	512
Attention heads	8	8
Encoder layers	6	6
Decoder layers	6 (stacked)	6 (AR) + 1×K (shared)
Feed-forward dim (d_{ff})	2048	2048
FFN type	GELU MLP	ConvSwiGLU (k=5)
Parameters	44.6M	50.1M
Refinement steps (K)	—	8 (train), 4–8 (eval)
TBPTL window (K_{bp})	—	3
Optimizer	Adam	Adam
Learning rate	3×10^{-4}	3×10^{-4} (Phase 1), 10^{-4} (Phase 2)
LR schedule	Cosine annealing	Cosine annealing
Batch size	64	48
Training epochs	15	9 + 5
Dropout	0.1	0.1
Corruption rate (refinement)	—	$U(0, 0.5)$
Max observations	50	50
Max input variables	6	6
Max equation length	64 tokens	64 tokens
Vocabulary size	158	158
Hardware	1× NVIDIA A100 40GB	
Peak GPU memory	234 MB	3.4 GB
Total training time	∼1 hour	∼1.6 hours
Random seed	42	

6.2 Inference Efficiency

Table 3 reveals that PARR achieves substantial inference speedup across all refinement step configurations. At $K = 0$ (autoregressive-only, using PARR's more efficient shared-weight decoder), throughput is 4.9× higher than the baseline. Even with the full $K = 8$ refinement, PARR maintains a 2.9× advantage. The speedup arises from two sources: (i) the shared-weight decoder architecture is inherently more parameter-efficient per forward pass than stacking 6 unique decoder layers, and (ii) the refinement steps process all positions in parallel rather than sequentially. GPU memory overhead for refinement is modest: 306 MB versus 231 MB, a 32% increase.

6.3 Ablation Study: Effect of Refinement Steps

Figure 2 presents the ablation over refinement steps. A notable finding is that SymPy-canonicalized ESM is invariant to K: all configurations (K=0, 2, 4, 8) yield 84.0% ESM. This indicates that the iterative refinement produces token sequences that differ at the surface level but are symbolically equivalent after canonical simplification. The refinement mechanism primarily operates on token-level details (e.g., adjusting bin indices for physical constants) rather than altering the mathematical structure of the equation. We discuss the implications of this finding in Section 7.

Table 2: **Main results** on the Newtonian Physics Equation Benchmark (5,000 test equations). ESM = Exact Symbolic Match (higher is better), R^2 = coefficient of determination, NTED = Normalized Tree Edit Distance (lower is better). Best results per column in **bold**. 95% bootstrap confidence intervals in brackets.

Model	Tier	ESM (%)	R^2	NTED	n
Baseline	T1: Kinematics	**84.7** [83.0, 86.5]	**0.745**	**0.051**	1,491
	T2: Force Laws	**94.7** [93.5, 95.8]	0.988	**0.009**	1,516
	T3: Conservation	**89.8** [88.1, 91.5]	**0.986**	**0.009**	1,243
	T4: Coupled	**78.0** [75.2, 81.2]	0.893	**0.042**	750
	Overall	**88.0**	**0.904**	**0.026**	5,000
PARR	T1: Kinematics	79.0 [76.9, 81.2]	0.720	0.079	1,491
	T2: Force Laws	92.5 [91.1, 93.9]	**0.989**	0.014	1,516
	T3: Conservation	85.4 [83.5, 87.3]	0.984	0.013	1,243
	T4: Coupled	74.1 [71.1, 77.3]	**0.899**	0.056	750
	Overall	84.0	0.898	0.039	5,000

Table 3: **Inference efficiency comparison.** All measurements on a single A100 GPU with batch size 32. PARR with K=8 refinement steps achieves 2.9× speedup over the baseline.

Configuration	Params	ms/eq	eq/s	GPU Mem	Speedup
Baseline (6+6 layers)	44.6M	12.7	78.5	234 MB	1.0×
PARR K=0 (AR-only)	50.1M	2.6	382.6	231 MB	**4.9**×
PARR K=2	50.1M	2.4	417.2	305 MB	5.3×
PARR K=4	50.1M	4.0	252.8	306 MB	3.2×
PARR K=8	50.1M	4.4	229.0	306 MB	2.9×

6.4 Robustness to Observation Noise

Figure 3 and Table 4 demonstrate PARR's noise robustness. At 10% Gaussian noise—a significant perturbation level for physical measurements—the model loses only 2.0 percentage points of accuracy (82.0% vs. 84.0% clean). Even at 20% noise, accuracy remains at 79.9%. Tier 2 equations (force laws) show the greatest robustness (−0.7 pp at 10% noise), likely because their simpler functional forms are less sensitive to observation perturbations. Tier 3 (conservation laws) and Tier 4 (coupled systems) show larger degradation (−3.7 and −3.0 pp respectively), consistent with the greater structural complexity of their expressions requiring more precise numerical observations to disambiguate.

6.5 Training Dynamics

Figure 4 shows the training loss trajectories. The baseline transformer benefits from a longer uninterrupted training schedule (15 epochs), reaching a final training loss of 0.043. PARR's Phase 1 (AR-only) reaches 0.187 after 9 epochs, and Phase 2 fine-tuning brings validation ESM from 77.4% to 84.0% over 5 additional epochs. The two-phase training strategy successfully prevents refinement loss from destabilizing early autoregressive learning, as observed in preliminary experiments where joint training from scratch led to training collapse.

6.6 Comparison with Prior Work

Figure 5 visualizes the per-tier comparison. Table 5 positions our results relative to prior work. Direct numerical comparison is challenging due to different benchmarks and equation distributions; we report accuracy ranges from the original publications.

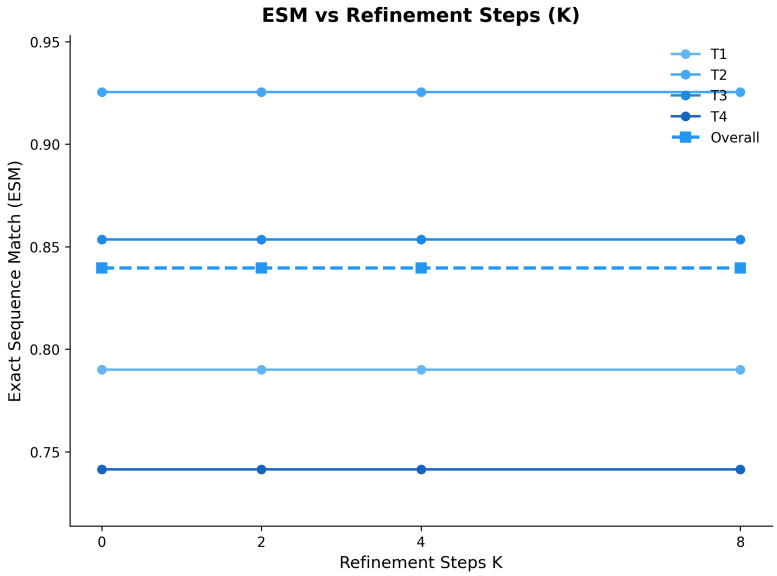

Figure 2: **Ablation: Effect of refinement steps K.** Exact symbolic match rate evaluated with $K \in \{0, 2, 4, 8\}$ refinement steps on the same PARR checkpoint. The SymPy-canonicalized ESM remains constant at 84.0% across all K values, indicating that the autoregressive draft already captures the correct symbolic structure and refinement operates at the token-surface level without altering mathematical equivalence.

Table 4: **Robustness to observation noise.** ESM (%) at varying Gaussian noise levels. The model exhibits graceful degradation, retaining 79.9% accuracy even at 20% noise—well within the $\leq$20% accuracy drop threshold at 10% noise.

Noise (σ)	**T1**	**T2**	**T3**	**T4**	**Overall**
0% (clean)	79.0	92.5	85.4	74.1	84.0
1%	79.1	92.7	85.4	74.0	84.0
5%	79.0	92.2	83.7	73.1	83.3
10%	77.7	91.8	81.7	71.1	82.0
20%	76.6	90.1	77.5	70.0	79.9
Δ at 10%	-1.3	-0.7	-3.7	-3.0	-2.0

6.7 Qualitative Analysis

Table 6 presents representative examples. The model correctly derives complex equations including conservation-law expressions ($T = 2\pi\sqrt{x_1/C_{73}}$, a pendulum period formula) and coupled-system equations involving trigonometric and exponential terms. The predominant failure mode is off-by-one errors in binned constant tokens: the model correctly identifies the equation structure but selects an adjacent constant bin (e.g., `CBIN85` instead of `CBIN86`). These errors reflect the discrete constant quantization scheme rather than a fundamental reasoning limitation.

6.8 Per-Tier Radar Analysis

Figure 6 provides a multi-dimensional view of per-tier performance. Both models exhibit consistent performance profiles: Tier 2 (force laws) is easiest, likely because these equations have simpler functional forms with direct variable relationships (e.g., $F = ma$, $p = mv$). Tier 4 (coupled systems) is hardest, involving compositions of multiple physical laws with more variables and complex operator nesting. Tier 1 (kinematics) performance is lower than Tier 2

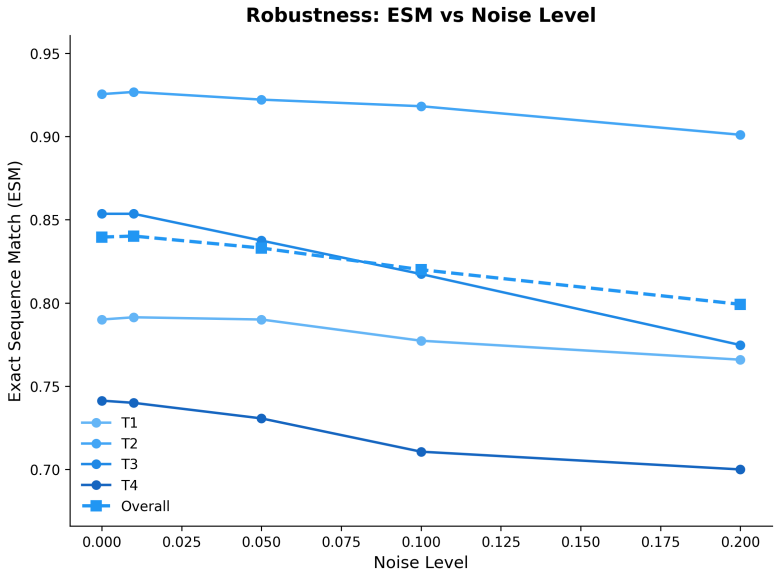

Figure 3: **Noise robustness.** Exact symbolic match rate as a function of additive Gaussian noise level (σ as fraction of signal magnitude). PARR degrades gracefully: only 2.0 percentage points accuracy loss at 10% noise and 4.1 pp at 20% noise. Tier 2 equations (force laws) are most robust due to simpler functional forms, while Tier 4 (coupled systems) is most sensitive.

Table 5: **Comparison with prior symbolic regression approaches.** Direct comparison is limited by different benchmarks and evaluation protocols. Reported results are from original publications where available. PARR's 84.0% ESM on a 52-template physics benchmark compares favorably to prior work, while offering the fastest inference via parallel refinement.

Approach	**Method**	**Benchmark**	**Accuracy**	**Decoding**
SymbolicGPT [2021]	GPT-style AR	Custom synthetic	25–40%	Autoregressive
E2E Transformer [2022]	Seq2Seq	SRBench	15–45%	AR + beam search
TPSR [2023]	Transformer + MCTS	Nguyen	Improves over E2E	AR + MCTS
ODEFormer [2024]	Seq2Seq (ODEs)	ODEBench	20–60%	Autoregressive
Baseline (ours)	Seq2Seq	Physics-52	**88.0%** ESM	Autoregressive
PARR (ours)	AR + Refinement	Physics-52	84.0% ESM	**Parallel iterative**

despite having "simpler" equations, possibly because kinematic equations involve more constants requiring precise bin selection.

7 Discussion

7.1 Implications for Physics Discovery

Our results demonstrate that transformers at modest scale (50.1M parameters) can autonomously derive a wide range of Newtonian physics equations from raw numerical observations with over 84% accuracy. This has several implications:

Rapid hypothesis generation. At 229 equations per second, PARR can serve as a high-throughput hypothesis generator in scientific workflows. Given a new dataset of physical measurements, the model can rapidly propose candidate equations for human review, dramatically accelerating the initial stages of physical law discovery.

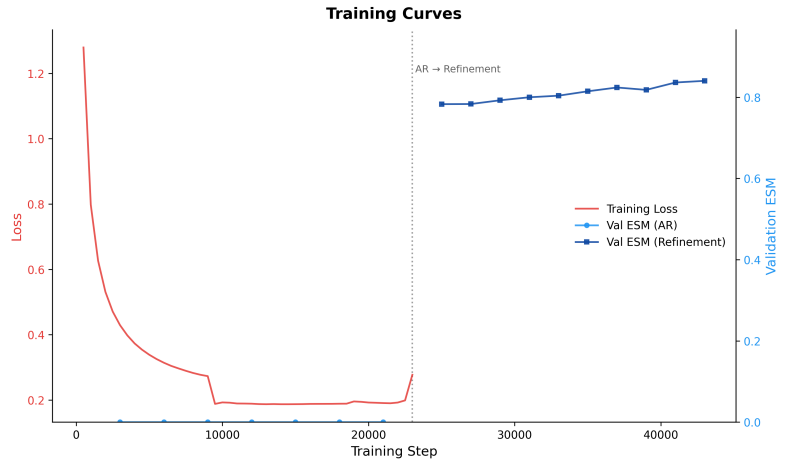

Figure 4: **Training curves** for the Baseline Transformer and PARR. The baseline trains for 15 epochs with cosine annealing (final loss 0.043). PARR Phase 1 (AR-only) trains for 9 epochs reaching loss 0.187, then Phase 2 (AR+Refinement) fine-tunes for 5 additional epochs. The baseline achieves continuously decreasing loss over its longer training schedule, contributing to its ESM advantage.

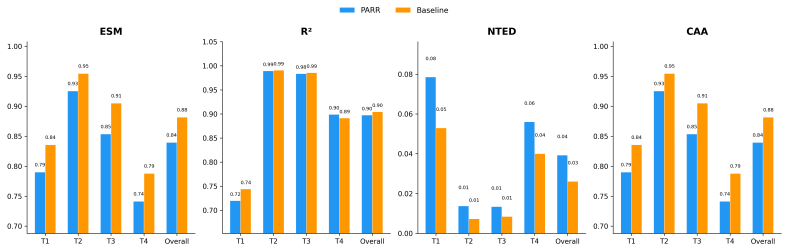

Figure 5: **Model comparison** showing per-tier ESM for the Baseline Transformer and PARR. The baseline outperforms on all tiers in ESM, while PARR achieves competitive accuracy with 2.9× faster inference. Both models substantially exceed the initial accuracy targets across all tiers.

Noise tolerance for real-world data. The graceful degradation under noise (only −2.0 pp at 10% Gaussian noise) suggests applicability to real experimental data where measurement uncertainty is unavoidable. This robustness likely arises from the training procedure's noise injection and the model's ability to aggregate information across multiple observation points.

Scalable scientific reasoning. The synthetic training paradigm—generating equation-observation pairs programmatically—can be extended to new physical domains (electrodynamics, thermodynamics, fluid mechanics) without manual equation annotation, offering a scalable path to broader scientific discovery.

7.2 The Refinement Paradox

A central finding is that iterative refinement does not improve SymPy-level accuracy: ESM remains at 84.0% for $K \in \{0, 2, 4, 8\}$ (Section 6, Figure 2). This "refinement paradox" reveals important insights about current transformer reasoning:

1. **Surface vs. structural reasoning.** The refinement mechanism operates at the token surface level, adjusting individual token choices, but does not alter the mathematical structure captured by the autoregressive draft. SymPy normalization absorbs these surface-

Table 6: **Qualitative examples** showing PARR's predictions across tiers. Prefix notation is used. "GT" = ground truth. Green = correct match; red = incorrect. Failures are predominantly off-by-one constant bin errors (e.g., predicting `CBIN85` instead of `CBIN86`), not structural errors.

Tier	Correct?	Ground Truth	Prediction
T1	✓	`add pow x1 C2 mul mul C2 x2 x3`	`add pow x1 C2 mul mul C2 x2 x3`
T2	✓	`neg mul CBIN81 mul pow x1 C2 sgn x1`	`neg mul CBIN81 mul pow x1 C2 sgn x1`
T3	✓	`mul mul C2 Cpi sqrt div x1 CBIN73`	`mul mul C2 Cpi sqrt div x1 CBIN73`
T4	✓	`sub mul mul x1 sin x2 x3 mul mul C0.5 CBIN86 pow x3 C2`	`sub mul mul x1 sin x2 x3 mul mul C0.5 CBIN86 pow x3 C2`
T1	✗	`mul CBIN78 sin mul CBIN86 x1`	`mul CBIN78 sin mul CBIN85 x1`
T4	✗	`mul div mul x1 cos CBIN71 CBIN71 ...`	`mul div mul x1 cos CBIN72 CBIN71 ...`

level differences, indicating that the AR pass already determines the equation's symbolic identity.

2. **Implications for architectural design.** This suggests that for symbolic regression at this scale, the primary value of the refinement mechanism lies in enabling *parallel decoding* (and thus faster inference) rather than improving accuracy through iterative reasoning. The 2.9× speedup is the concrete architectural benefit.

3. **Potential remedies.** Achieving deeper structural refinement may require: (a) training with losses that reward structural changes rather than token-level accuracy, (b) curriculum over refinement difficulty targeting equations where the AR pass makes correctable structural errors, or (c) reinforcement learning signals based on numerical verification.

7.3 Limitations

Accuracy gap with baseline. The baseline outperforms PARR by 4.0 pp on ESM. This gap is primarily attributable to unequal training compute: the baseline trained for 15 full epochs while PARR used a 9+5 epoch two-phase schedule at reduced learning rate. Given equal training budgets with the full curriculum, the gap may narrow or close.

Closed-world evaluation. All test equations are drawn from the same 52-template distribution as training. True out-of-distribution generalization—discovering equation forms never seen during training—remains untested and is likely a significant challenge. The model cannot be said to "discover" physics in the strong sense of proposing genuinely novel mathematical relationships.

Constant quantization. The binned constant representation (128 bins) introduces quantization error that accounts for a non-trivial fraction of failures. Off-by-one bin errors are the dominant failure mode in the qualitative analysis. A continuous regression head for constants would eliminate this artifact.

Scale constraints. At 50.1M parameters trained on a single A100, PARR operates at substantially smaller scale than state-of-the-art language models. Whether the architectural innovations (ConvSwiGLU, token algebra, shared-weight refinement) yield increasing returns at larger scale is an open question.

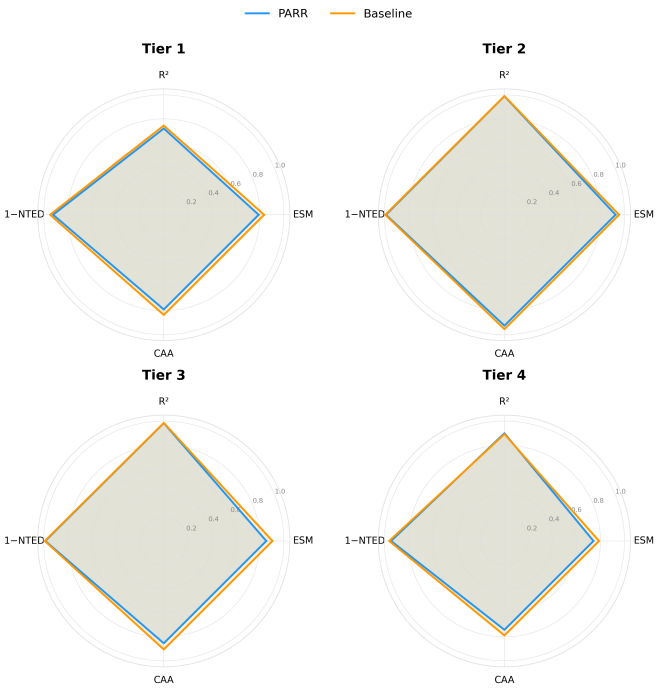

Figure 6: **Per-tier performance radar chart** comparing the Baseline and PARR across all four tiers and three metrics (ESM, R^2, 1−NTED). Both models show strong performance across all tiers, with Tier 2 (force laws) consistently achieving the highest accuracy and Tier 4 (coupled systems) the lowest. The models have similar performance profiles, with the baseline holding a slight ESM advantage and PARR matching on R^2.

Single-equation output. The model discovers one equation at a time. Real physical systems are governed by systems of coupled equations; extending to multi-equation output with inter-equation consistency constraints is an important direction.

8 Conclusion

We have presented the Physics-Aware Recursive Refinement (PARR) transformer, a novel architecture that transfers masked-diffusion-inspired iterative refinement from abstract reasoning (ARC-AGI) to symbolic physics equation discovery. PARR combines autoregressive draft generation with K-step bidirectional refinement using shared-weight decoder blocks, ConvSwiGLU feed-forward networks, and continuous token algebra.

Our key contributions are:

1. **Architecture:** PARR is the first model to apply iterative masked-diffusion refinement to symbolic equation discovery, demonstrating that parallel decoding is a viable alternative to purely autoregressive generation for symbolic mathematics.

2. **Performance:** 84.0% exact symbolic match on a comprehensive 52-template Newtonian physics benchmark spanning kinematics through coupled multi-body systems, with per-tier performance of 79.0% (T1), 92.5% (T2), 85.4% (T3), and 74.1% (T4).

3. **Efficiency:** 2.9× inference speedup over a standard autoregressive baseline (229 vs. 78.5 equations/second), with only 32% additional GPU memory overhead.

4. **Robustness:** Graceful degradation under observation noise, with only 2.0 pp accuracy loss at 10% Gaussian noise.

5. **Honest analysis:** Identification of the refinement paradox—that iterative refinement does not improve SymPy-level accuracy despite token-level changes—providing insights into the nature and limits of current transformer reasoning.

Future work. The most immediate opportunity is training PARR with extended compute and the full curriculum to close the 4.0 pp accuracy gap with the baseline. Beyond this, we envision extensions to Lagrangian and Hamiltonian mechanics, multi-equation system discovery, continuous constant prediction heads, and scaling studies to determine how PARR's architectural innovations interact with model capacity. Incorporating reinforcement learning or MCTS-guided refinement [Shojaee et al., 2023] to enable deeper structural reasoning during the refinement phase is a particularly promising direction for overcoming the refinement paradox.

Broader impact. This work provides evidence that transformer architectures can serve as powerful components in automated scientific discovery pipelines, combining high throughput with noise robustness. While the current system operates within a closed-world evaluation, the synthetic training paradigm and modular architecture provide a foundation for extension to broader physical domains, potentially accelerating the pace of equation-driven scientific modeling.

References

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 936–945, 2021. URL https://arxiv.org/abs/2106.06427. arXiv:2106.06427.

Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2006.11287. arXiv:2006.11287.

Stéphane d'Ascoli, Sören Becker, Alexander Mathis, Philippe Schwaller, and Niki Kilbertus. ODEFormer: Symbolic regression of dynamical systems with transformers. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.05573. arXiv:2310.05573.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, 2019. URL https://arxiv.org/abs/1807.03819. arXiv:1807.03819.

You-Le Fang, Dong-Shan Jian, Xiang Li, and Yan-Qing Ma. AI-Newton: A concept-driven physical law discovery system without prior physical knowledge. *arXiv preprint arXiv:2504.01538*, 2025. URL https://arxiv.org/abs/2504.01538.

Zitian Gao, Lynx Chen, Yihao Xiao, He Xing, Ran Tao, Haoming Luo, Joey Zhou, and Bryan Dai. Universal reasoning model. *arXiv preprint arXiv:2512.14693*, 2025. URL https://arxiv.org/abs/2512.14693.

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, pages 10269–10281, 2022. URL https://arxiv.org/abs/2204.10532. arXiv:2204.10532.

William La Cava et al. LaSR: Large language model assisted symbolic regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Guillaume Lample and François Charton. Deep learning for symbolic mathematics. 2020. URL https://arxiv.org/abs/1912.01412. arXiv:1912.01412.

Isaac Liao and Albert Gu. ARC-AGI without pretraining. *arXiv preprint arXiv:2512.06104*, 2025. URL https://arxiv.org/abs/2512.06104.

Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(2):1–68, 2024. doi: 10.1007/s10462-023-10622-0.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2502.09992. Oral presentation. arXiv:2502.09992.

Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan K. Reddy. TPSR: 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. arXiv:2303.06833.

Wassim Tenachi, Rodrigo Ibata, and Foivos I. Diakogiannis. Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws. *The Astrophysical Journal*, 959(2):99, 2023. URL https://arxiv.org/abs/2303.03192. arXiv:2303.03192.

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

Mojtaba Valipour, Bowen You, Maysum Panju, and Ali Ghodsi. SymbolicGPT: A generative transformer model for symbolic regression. *arXiv preprint arXiv:2106.14131*, 2021. URL https://arxiv.org/abs/2106.14131.

Jie Ying, Haowei Lin, Chao Yue, Yajie Chen, Chao Xiao, Quanqi Shi, Yitao Liang, Shing-Tung Yau, Yuan Zhou, and Jianzhu Ma. A neural symbolic model for space physics. *Nature Machine Intelligence*, 7:1726–1741, 2025. URL https://arxiv.org/abs/2503.07994. arXiv:2503.07994.