Learning to Compute Products of Polynomials with Transformers

Yuxuan Song, Bin Wang, and **Changbo Chen**

Chongqing Institute of Green and Intelligent Technology,
Chinese Academy of Sciences, Chongqing, China
chenchangbo@cigit.ac.cn

SCML 2026, RISC, Hagenberg, Austria
July 7, 2026

Outline

Outline

Learning symbolic computation with Transformers

Basic operations (understanding the learning)

- Integer addition/multiplication (Cho et al. 24; McLeish et al. 24; Cho et al. 25).

Complex computations (end-to-end learning for accelerating)

- Symbolic integration (Lample & Charton 20); Lyapunov function (Alfarano et al. 24);
- Gröbner bases (Keta et al. 24; Malhou et al. 26).

Heuristics requiring complex computations (data barrier)

- CAD variable ordering (R. Jing, Y. Zhao and C.).
  This afternoon session (17:00-18:15, Room: Gemeindesaal).

This talk: learning the basic aiming for the complex.

The problem to prove

- $\forall x \in k^n, F(x) = 0 \Rightarrow p(x) = 0.$
- This is equivalent to proving that $V(F) \subseteq V(p)$.

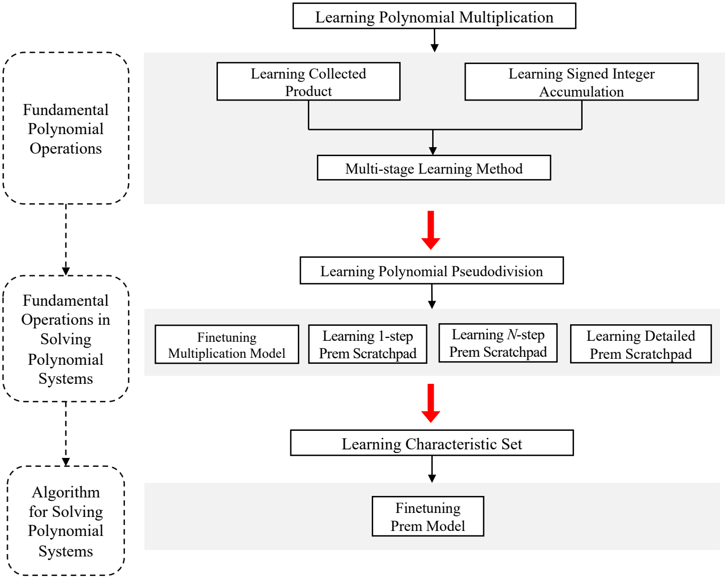
The problem to prove

- $\forall x \in k^n, F(x) = 0 \Rightarrow p(x) = 0$.
- This is equivalent to proving that $V(F) \subseteq V(p)$.

From proving to solving (Wen-Tsün Wu's method, 1970s)

- Compute a reduced triangular set C such that $C \subseteq \langle F \rangle$ and $\mathrm{prem}(F, C) = 0$ (C is called a Ritt-Wu characteristic set of F).
- Note that $\mathrm{prem}(f, C) = h_C^e f - \sum_{i=1}^{|C|} q_i C_i$, for any $f \in F$.
- Thus $V(F) \setminus V(h_C) = V(C) \setminus V(h_C)$.
- Assume that $h_C \neq 0$ holds, then it's equivalent to show that $V(C) \subseteq V(p)$.
- Then it's equivalent to show that $\mathrm{prem}(p, C) = 0$ holds, as $\mathrm{prem}(p, C) = h_C^{e'} p - \sum_{i=1}^{|C|} q_i' C_i$.

Overview of the framework



Outline

Polynomial multiplication

Polynomial

- Let $\mathbf{x}^\alpha = x_1^{\alpha_1} \cdots x_n^{\alpha_n}$, where $\alpha_i \in \mathbb{N}$.
- Let $f = \sum_{\alpha} a_\alpha \mathbf{x}^\alpha,\ a_\alpha \in \mathbb{Z}^*$.
- Let $g = \sum_{\beta} b_\beta \mathbf{x}^\beta,\ b_\beta \in \mathbb{Z}^*$.
- Two terms $a_\alpha \mathbf{x}^\alpha$ and $b_\beta \mathbf{x}^\beta$ are called like terms if $\alpha = \beta$.

Multiplication

- The product $fg = \sum_{\gamma} \left(\sum_{\gamma = \alpha + \beta} a_\alpha b_\beta \right) \mathbf{x}^\gamma$.

Transformer

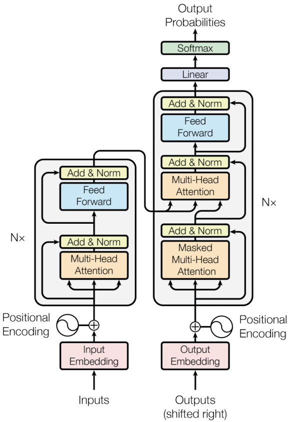


Figure 1: The Transformer - model architecture.

Use Transformer in a direct way

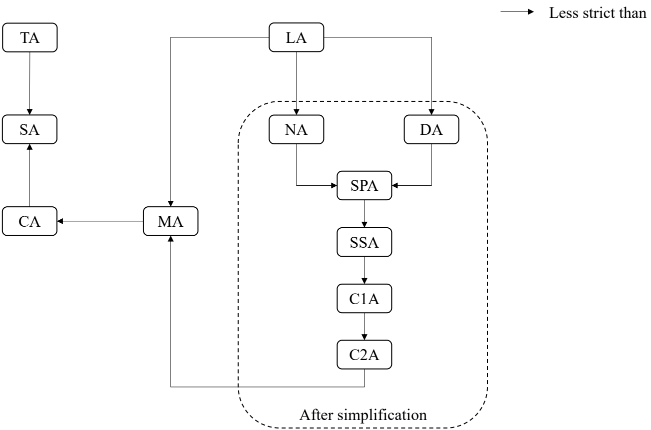
- Prepare a dataset of (f, g, fg).
- Tokenize both input and output.
- Train with teacher-forcing mode.
- Predict with auto-regressive mode.

Potential problem

- Evaluation metric?
- Learning integer arithmetic?

- The basic setting in this work: embedding dimension 512, hidden dimension 2048, 8 heads, 6 encoders/decoders.
- NVIDIA® GeForce RTX™ 5080 (16GB) and 4090 (24GB).

Evaluation metrics



- SA: exact as a sequence
- CA: exact up to sorting
- MA: mathematically equivalent

An example illustrating different metrics

Label	$116 * x_1^9 * x_2^3 - 7 * x_1^2$	
Metric	Satisfaction	Violation
SA	$116 * x_1^9 * x_2^3 - 7 * x_1^2$	$-7 * x_1^2 + 116 * x_1^9 * x_2^3$
CA	$-7 * x_1^2 + 116 * x_1^9 * x_2^3$	$26 * x_1^9 * x_2^3 - 7 * x_1^2 + 90 * x_1^9 * x_2^3$
MA	$26 * x_1^9 * x_2^3 - 7 * x_1^2 + 90 * x_1^9 * x_2^3$	$113 * x_1^9 * x_2^3 - 7 * x_1^2$
C2A	$113 * x_1^9 * x_2^3 - 7 * x_1^2$	$143 * x_1^9 * x_2^3 - 7 * x_1^2$
C1A	$143 * x_1^9 * x_2^3 - 7 * x_1^2$	$14 * x_1^9 * x_2^3 - 7 * x_1^2$
SSA	$14 * x_1^9 * x_2^3 - 7 * x_1^2$	$14 * x_1^9 * x_2^3 + 7 * x_1^2$
SPA	$14 * x_1^9 * x_2^3 + 7 * x_1^2$	$14 * x_1^8 * x_2^3 + 7 * x_1$
NA	$14 * x_1^8 * x_2^3 + 7 * x_1$	$-14 * x_1^8 * x_2^3 + 7 * x_1 + x_2$
DA	$14 * x_1^8 * x_2^4 + 7 * x_1^2$	$4 * x_1^3 * x_2^3 + 7 * x_1$
LA	$4 * x_1^3 * x_2^3 + 7 * x_1$	$4 * x_1^{x_2} * x_2^3 + 8*$

Tokenization scheme

Univariate polynomial	
Scheme	$-x^{14} - 31 * x^6 - 9 * x - 106$
ImUs	[$-$, x, $\wedge$, 14, $-$, 31, $*$, x, $\wedge$, 6, $-$, 9, $*$, x, $-$, 106]
ImDs	[$-$, x, $\wedge$, 1, 4, $-$, 3, 1, $*$, x, $\wedge$, 6, $-$, 9, $*$, x, $-$, 1, 0, 6]
ExUs	[$-$, 1, $*$, x, $\wedge$, 14, $-$, 31, $*$, x, $\wedge$, 6, $-$, 9, $*$, x, $\wedge$, 1, $-$, 106]
ExDs	[$-$, 1, $*$, x, $\wedge$, 1, 4, $-$, 31, $*$, x, $\wedge$, 6, $-$, 9, $*$, x, $\wedge$, 1, $-$, 1, 0, 6]
Multivariate polynomial	
Scheme	$36 * x_0^9 * x_1^3 + x_1 * x_2^{12}$
ImUs	[36, $*$, $x0$, $\wedge$, 9, $*$, $x1$, $\wedge$, 3, $+$, $x1$, $*$, $x2$, $\wedge$, 12]
ImDs	[3, 6, $*$, $x0$, $\wedge$, 9, $*$, $x1$, $\wedge$, 3, $+$, $x1$, $*$, $x2$, $\wedge$, 1, 2]
ExUs	[36, $*$, $x0$, $\wedge$, 9, $*$, $x1$, $\wedge$, 3, $+$, 1, $*$, $x1$, $\wedge$, 1,$*$, $x2$, $\wedge$, 12]
ExDs	[3, 6, $*$, $x0$, $\wedge$, 9, $*$, $x1$, $\wedge$, 3, $+$, 1, $*$, $x1$, $\wedge$, 1,$*$, $x2$, $\wedge$, 1, 2]

- **Implicit (Im)**: default expression in common mathematical software like Maple.
- **Explicit (Ex)**: explicit expression with visible unit coefficients and exponents.
- **Unsigned (Us)**: the whole unsigned part of every number is regarded as a token.
- **Digits-split (Ds)**: unsigned part of every number is split into tokens digit by digit.

The datasets

Dataset	Size	v	d	t / o	e	c	Stage	L^{in}_{max}	L^{out}_{max}
v1d1C1	100,000	1	1	N/A	N/A	$-9..9$	Direct	19	20
							Collect	19	24
							Simplification	24	20
v1d3C1	100,000	1	3	N/A	N/A	$-9..9$	Direct	43	52
							Collect	43	84
							Simplification	84	52
v1d7C1	100,000	1	7	N/A	N/A	$-9..9$	Direct	91	121
							Collect	91	281
							Simplification	281	121
v1d15C1	100,000	1	15	N/A	N/A	$-9..9$	Direct	101	270
							Collect	101	969
							Simplification	969	270
v2d2C1	100,000	2	2	N/A	N/A	$-9..9$	Direct	81	139
							Collect	81	214
							Simplification	214	139
v3d2C1	100,000	3	2	N/A	N/A	$-9..9$	Direct	141	378
							Collect	141	610
							Simplification	610	378
v2t2E1C1	100,000	2	N/A	2	$0..9$	$-9..9$	Direct	43	54
							Collect	43	54
							Simplification	54	54
v3t3E1C1	100,000	3	N/A	3	$0..9$	$-9..9$	Direct	87	164
							Collect	87	164
							Simplification	164	164
C4o9	1,000,000	N/A	N/A	$2..9$	N/A	$-9999..9999$	Decoder-only	112	

Initial try

Dataset	Encoding	TA	SA	LA	DA	NA	SPA	SSA	C1A	C2A	MA	CA
v1d1C1	ImUs	99.98	99.95	**100**	**100**	**100**	**100**	**100**	99.98	99.95	99.95	99.95
	ImDs	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
	ExUs	99.99	99.97	**100**	**100**	**100**	**100**	**100**	99.97	99.97	99.97	99.97
	ExDs	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
v1d3C1	ImUs	92.22	23.84	**100**	99.97	**100**	98.58	93.93	49.83	24.89	23.84	23.84
	ImDs	**94.49**	**70.93**	**100**	**100**	**100**	**99.40**	**96.17**	**82.23**	**71.86**	**70.93**	**70.93**
	ExUs	92.17	4.24	**100**	99.94	**100**	97.62	87.53	14.90	4.65	4.24	4.24
	ExDs	85.46	29.54	**100**	99.98	**100**	98.60	92.93	52.78	30.75	29.54	29.54
v1d7C1	ImUs	64.92	0	99.94	99.61	99.94	90.68	45.53	0	0	0	0
	ImDs	56.70	0	**100**	99.99	**100**	**94.36**	**85.88**	**9.48**	**0.05**	0	0
	ExUs	**69.83**	0	99.97	99.82	99.97	89.34	0.16	0	0	0	0
	ExDs	6.9478	0	99.99	99.89	99.99	90.20	0.20	0	0	0	0
v2t2E1C1	ImUs	98.16	94.66	**100**	99.94	**100**	99.71	99.66	99.44	99.42	99.42	98.57
	ImDs	98.05	96.22	99.99	99.95	99.99	99.84	**99.82**	**99.74**	**99.73**	**99.73**	**98.83**
	ExUs	**98.79**	**96.47**	99.99	99.98	99.99	**99.86**	99.79	98.91	98.84	98.84	98.11
	ExDs	97.33	93.88	**100**	99.40	**100**	98.31	98.27	97.03	96.86	96.86	96.05
v3t3E1C1	ImUs	96.03	65.54	**100**	99.91	**100**	98.53	98.50	98.27	98.27	98.27	97.85
	ImDs	95.54	76.55	**100**	**99.96**	**100**	**99.61**	**99.56**	**99.49**	**99.48**	**99.48**	**99.07**
	ExUs	**97.56**	72.88	**100**	99.93	**100**	99.32	94.94	94.42	94.41	94.40	94.00
	ExDs	95.85	**77.07**	**100**	99.94	**100**	99.42	97.33	97.25	97.24	97.23	96.83

- Difficulty arises for univariate dense polynomials.

Why happened to v1d7C1?

Monomial	Maximum number of like terms	Accuracy
x^{14}	1	45.6164
x^{13}	2	28.2015
x^{12}	3	10.5933
x^{11}	4	3.5427
x^{10}	5	1.5304
x^{9}	6	0.9756
x^{8}	7	0.9055
x^{7}	8	0.7839
x^{6}	7	0.9053
x^{5}	6	0.8549
x^{4}	5	1.0261
x^{3}	4	3.2931
x^{2}	3	10.6157
x^{1}	2	28.0414
x^{0}	1	47.9760

Learning polynomial multiplication

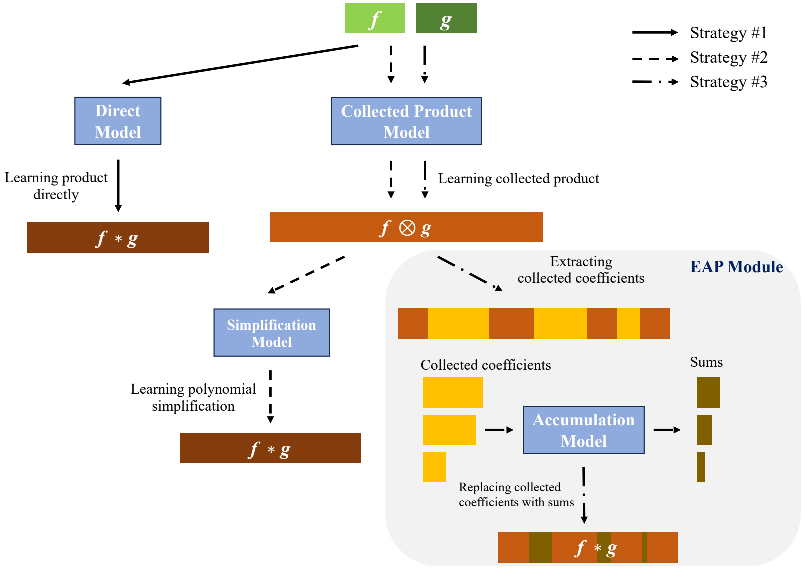


Figure: Three different strategies for learning the multiplication.

Collected product

$$
\begin{aligned}
f &= 5 * x^3 + 3 * x^2 - 4 * x - 8, \\
g &= 2 * x^3 - 6 * x^2 - 4 * x - 5, \\
f \otimes g &= 10 * x^6 \, (-30 + 6) * x^5 \, (-20 - 18 - 8) * x^4 \\
& \quad (-25 - 12 + 24 - 16) * x^3 \, (-15 + 16 + 48) * x^2 \\
& \quad (+20 + 32) * x + 40.
\end{aligned}
$$

Accumulation model $(-36 - 68 + 97 + 8)$

Tokens	-	3	6	-	6	8	+	9	7	+	8	=	0	+	>	6	3	-	>	4	0	1	-	>	7	-	>	1	+
PosID1	1	3	2	1	3	2	1	3	2	1	2	0	2	1	0	2	3	1	0	2	3	4	1	0	2	1	0	2	1
PosID2	1	1	1	2	2	2	3	3	3	4	4	1	1	1	2	2	2	2	3	3	3	3	3	4	4	4	5	5	5

Position coupling + Scratchpad.

Comparison of our method and [CCBY25]

Cho *et al.*	**Tokens**	28+96+77=0>82>421>102
	PosID1	321321321 21231 2341234
	PosID2	111222331 12223 3334444
Our method	**Tokens**	+28+96+77=0+>82+>421+>102+
	PosID1	132132132 021 0231 02341 02341
	PosID2	111222333 111 2222 33333 44444

Figure: Comparison of our method and the method in [CCBY25].

- We extend [CCBY25] to include signed integers.
- Cho et al. treats "+" as a separator, sharing the same PosID1 as the delimiter ">".
- We treat the sign as a meaningful digit in an operand with its own PosID1.

Outline

Performance of three strategies

Type	Dataset	S	TA	SA	MA	CA
Univariate	v1d1C1	1	**100**	**100**	**100**	**100**
		2	99.99	99.97	99.97	99.97
		3	98.84	98.84	**100**	**100**
	v1d3C1	1	85.46	29.54	29.54	29.54
		2	**98.41**	94.36	94.37	94.37
		3	95.09	**95.06**	**99.91**	**99.91**
	v1d7C1	1	6.94	0	0	0
		2	6.26	0	0	0
		3	**90.24**	**90.14**	**99.57**	**99.57**
Multi-Dense	v2d2C1	1	17.69	0.03	0.03	0.03
		2	97.07	90.62	90.62	90.62
		3	**99.77**	**99.56**	**99.56**	**99.56**
	v3d2C1	1	3.92	0	0	0
		2	0.61	0	0	0
		3	**99.12**	**98.87**	**98.87**	**98.87**
Multi-Sparse	v2t2E1C1	1	97.33	93.88	96.86	96.05
		2	97.50	94.48	97.92	97.07
		3	**98.42**	**95.35**	**97.95**	**97.95**
	v3t3E1C1	1	95.85	77.07	97.23	96.83
		2	96.39	80.17	98.92	98.51
		3	**96.86**	**80.75**	**99.27**	**99.27**

LLM: Qwen-3.5-4B and DeepSeek-Math-7B-Base

- Qwen-3.5-4B: No deep thinking (Mode 1); with deep thinking (Mode 2); DeepSeek-Math-7B-Base: (Mode 3).
- Randomly choose 100 examples from each testing dataset.
- Token limit: Mode 1 (1024); Mode 2 (35000); Mode 3 (1024).

Type	Dataset	Mode	MA	CA	Total Time (s)
Univariate	v1d1C1	1	33	33	39.44
		2	100	100	1998.01
		3	27	27	24.91
	v1d3C1	1	0	0	107.33
		2	91	91	13809.70
		3	0	0	93.27
	v1d7C1	1	0	0	197.14
		2	68	68	34610.37
		3	0	0	439.15
Multi-Dense	v2d2C1	1	0	0	441.52
		2	46	46	43596.81
		3	0	0	696.62
	v3d2C1	1	0	0	612.20
		2	0	0	60244.55
		3	0	0	1082.19
Multi-Sparse	v2t2E1C1	1	0	0	97.03
		2	95	95	15398.13
		3	2	2	175.49
	v3t3E1C1	1	0	0	565.28
		2	96	96	18697.16
		3	0	0	1393.77

Application on learning characteristic sets

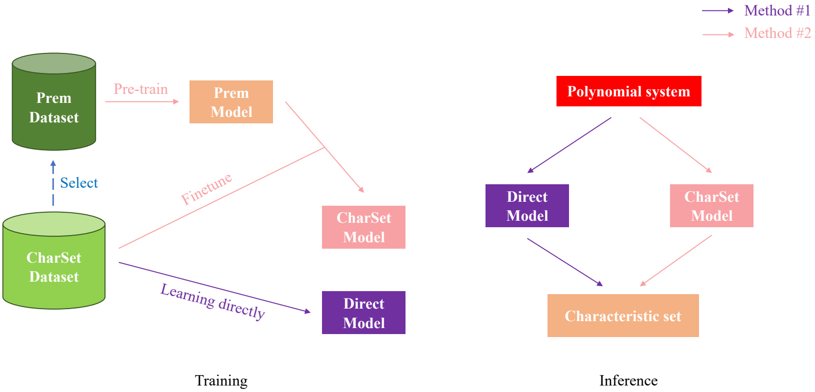


Figure: Learning schemes for characteristic sets of polynomial equations.

Experimental results

Table: Evaluation on learning polynomial pseudo division.

Dataset	M	TA	SA	LA	DA	NA	SPA	SSA	C1A	C2A	MA	CA
v1d3C1	1	47.43	2.26	**100**	98.49	**100**	97.42	24.95	2.95	2.39	2.26	2.26
	2	**98.90**	**97.30**	**100**	**99.96**	**100**	**99.96**	**99.60**	**97.68**	**97.40**	**97.30**	**97.30**
v1d7C1	1	11.87	0	**100**	98.52	**100**	91.66	1.74	0	0	0	0
	2	**32.47**	**1.62**	**100**	**98.76**	**100**	**92.97**	**90.82**	**2.75**	**1.64**	**1.62**	**1.62**
v3t3E1C1L3	1	84.74	65.85	**100**	98.04	**99.99**	90.18	85.55	81.29	80.77	80.71	80.49
	2	**90.28**	**77.64**	99.98	**98.92**	99.97	**93.35**	**90.44**	**87.83**	**87.56**	**87.50**	**87.25**

Table: Evaluation on learning characteristic set.

Dataset	M	Pre-training	TA	SA	LA	DA	NA	SPA	C1A	MA	CA
v3t3e2C1L3	1	N/A	60.47	42.12	100	96.54	85.73	78.83	45.36	45.31	42.12
	2	v3t3E1C1L3	**77.50**	**62.32**	**100**	**97.63**	**92.69**	**89.28**	**68.30**	**67.21**	**62.32**
v3t3e2F7	1	N/A	82.05	53.45	100	96.27	84.65	79.30	54.09	54.09	53.45
	2	v3t3E1F7	**84.55**	**59.31**	**100**	**97.23**	**86.85**	**83.93**	**59.99**	**59.99**	**59.31**
v3t3E1F7S2	2	v3t3E1F7	98.99	98.80	100	100	99.15	98.80	98.80	98.80	98.80

- $M = 1$: direct model; $M = 2$: fine-tuned model.
- Multi-step fine-tuning: $v3t3E1C1L3$ ($multi$) $\rightarrow$ $v3t3E1C1L3$ ($prem$) $\rightarrow$ $v3t3E1F7$ ($prem$) $\rightarrow$ $v3t3e2F7/v3t3E1F7S2$ ($charset$).

Outline

Conclusion and future work

Taking home message

- Learning basic operations helps on learning complex ones.
- Position coupling, scratchpad, and pre-training & fine-tuning can greatly improve the performance of Transformer models on learning symbolic computation.

Limitation and future work

- A one-stage model for effectively learning polynomial multiplication is still missing.
- A scratchpad-based pseudo division (characteristic set) learning method is working in progress.
- Bring scalability for symbolic computation via a specialized LLM.

References I

Hanseul Cho, Jaeyoung Cha, Srinadh Bhojanapalli, and Chulhee Yun, *Arithmetic transformers can length-generalize in both operand length and count*, Proceedings of the 13th International Conference on Learning Representations, 2025.