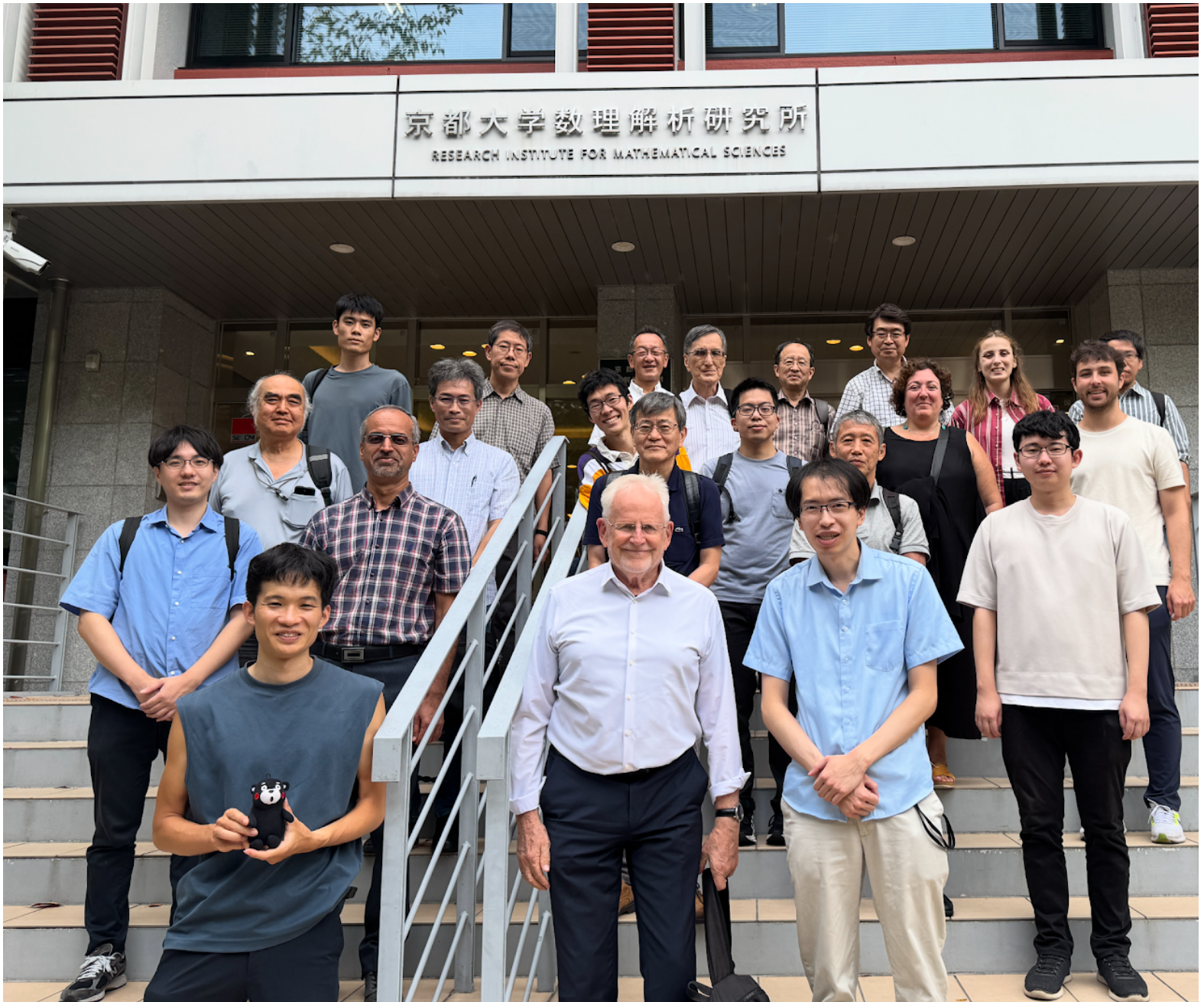
Efficient Dataset Generation for Bases of Zero-Dimensional Ideals

Yuki Ishihara* and Kazuhiro Yokoyama+

*Nihion University, +Rikkyo University, Japan

SCML 2026 @ RISC Linz

Gröbner 2025 at Kyoto University



Overview

Overview of Slides

Transformer for Computer Algebra

- Transformer is a Deep Learning Model, which is used in ChatGPT (Generative Pre-trained Transformer)
- Transformer is used to learn Symbolic Computation

+ GCD (Charton, 2024)
+ Integral (Lample and Charton, 2020)
+ Matrix Computation (Charton, 2022)
+ Gröbner/Border Basis (Kera et al., 2024/2025)
+ Lyapunov function (Alfarano et al., 2024)

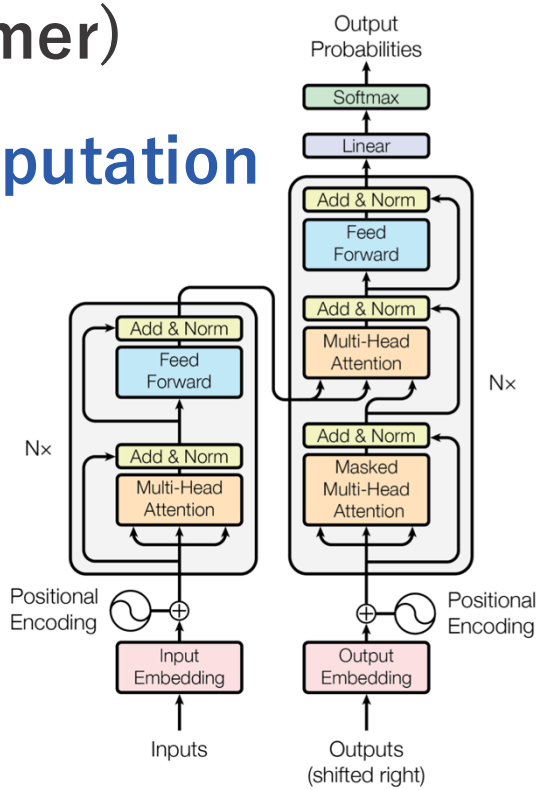


Figure 1: The Transformer - model architecture.

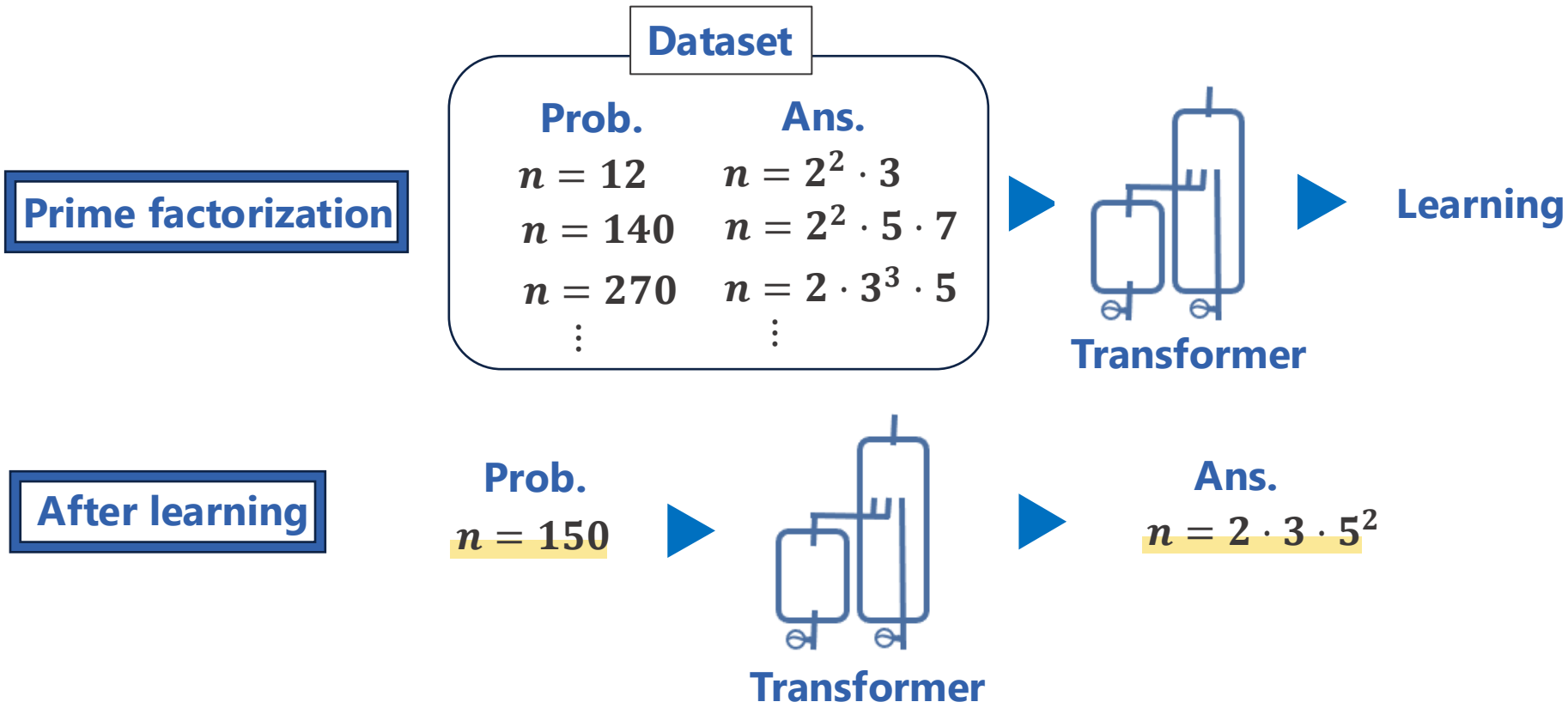
"Attention is all you need"
(Vaswani et al, 2017)

➔ Transformers have the potential

to speed up computations in computer algebra.

Learning of Transformer

1. Prepare many pairs "Problem" and "Answer" (e.g. 1 million).
2. Feed them to the Transformer and train them.
3. Based on the learning, Transformer answers new one.



Issues in Learning Symbolic Computations

・Unlike language learning, datasets for Symbolic Computations tend to be insufficient ⇨ Need to generate datasets

The Paradox of Dataset Generation

To accelerate computationally intensive tasks through learning, a large dataset are required, but generating the dataset takes an enormous amount of time.

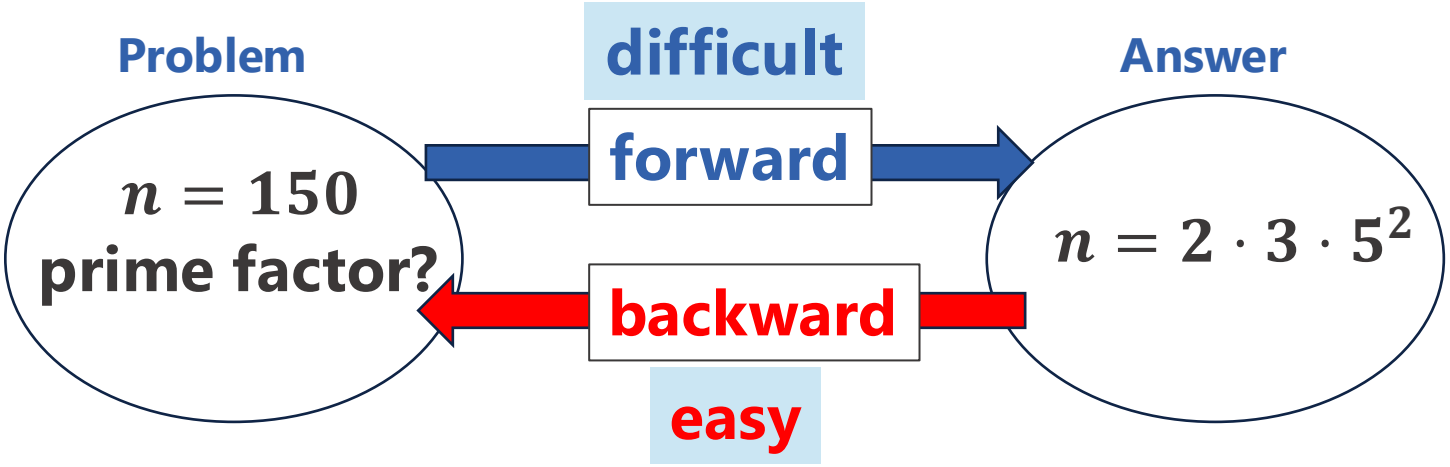
To generate one million data, where one datum take one minute to compute, it requires “one million minutes” = “694 days”.

⇨ Even if you could speed it up to one second by the learning, it would take one second and two years.

Avoiding the Paradox

Not create “answers” from “problems” (forward generation)

But create “problems” from “answers” (backward generation)



This technique used in

- Integral Computation (Lample and Charton, 2020)
- Gröbner (Border) Basis Computation (Kera et al, 2024/2025)

Main Theorem in Kera et al, 2024

Theorem 4.6. in [Kera et al. 2024]

I: **0-dimensional ideal in** $k[x_1, \dots, x_n]$.

$G = (g_1, \dots, g_t)^T$: a Gröbner basis of I.

$F = (f_1, \dots, f_s)^T = AG$ where A is $s \times t$ matrix on $k[x_1, \dots, x_n]$

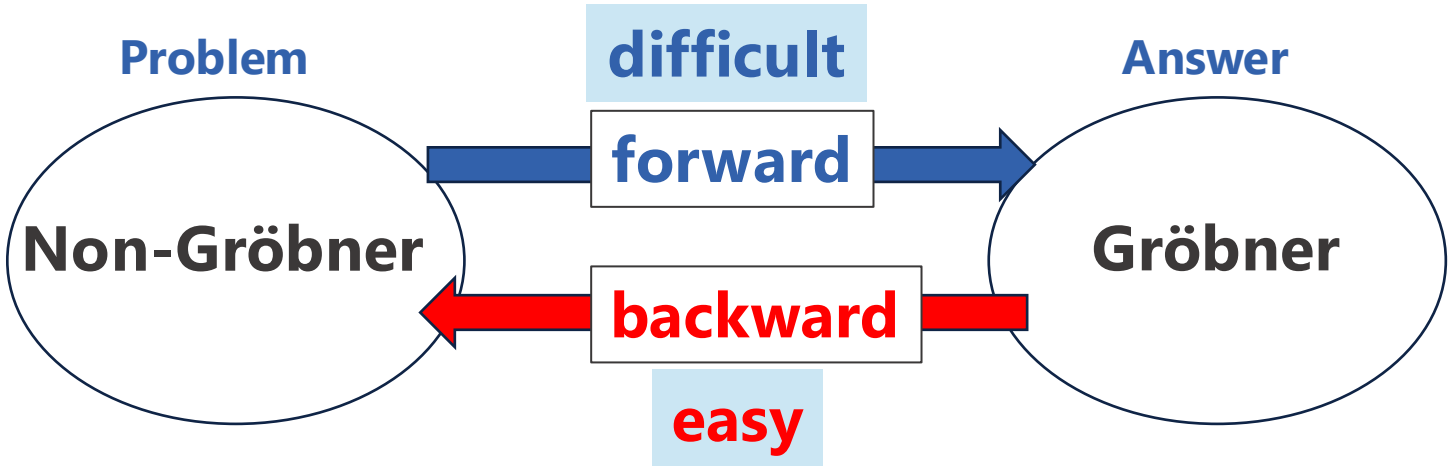
A **has the left-inverse matrix** $\Longrightarrow \langle F \rangle = \langle G \rangle$

By multiplying the Gröbner basis G from the left with a matrix, a large number of non-Gröbner bases $F = AG$ can be generated.

Backward Generation for Gröbner Bases

Not create “Gröbner” from “Non-Gröbner” (forward)

But create “Non-Gröbner” from “Gröbner” (backward)



$\{3x + 4y - 3, 9xy + 13y^2\}$ $\{3x + 4y - 3, y^2 + 9y\}$

AG ← G

Experiment “Timing of Dataset Generation”

Total runtime [sec] for 1000 samples.

Method	$n = 2$	$n = 3$	$n = 4$	$n = 5$
F. (STD)	**4.20**	216.3	740.1†	1411.1‡
F. (SLIMGB)	4.29	183.4	697.5†	1322.7‡
F. (STDFGLM)	7.22	8.29	21.0	164.3
B. (ours)	5.23	**5.46**	**7.05**	**7.91**

Exponential growth

Quadratic growth

† (13%+) and ‡ (25%+) timeouts included (with 5-sec budget for each sample)

- Backward can generate datasets faster than forward.
- Forward is exponential but backward is quadratic.

“Accuracy Rate” of Gröbner Bases by ML

	$n=2$	$n=3$	$n=4$	$n=5$
$\mathbb{Q}$	93.7	88.7	90.8	86.5
$\mathbb{F}_7$	72.3	78.1	71.3	84.3
$\mathbb{F}_{31}$	46.8	50.2	51.1	28.6

- For $\mathbb{Q}[x, y]$, the accuracy rate is over 90%
- The accuracy rate of learning varies significantly depending on the coefficient field, with $\mathbb{Q}$ performing better than finite fields $\mathbb{F}_p$

Main Theorem in Kera et al, 2024

Theorem 4.6. in [Kera et al. 2024]

I: 0-dimensional ideal in $k[x_1, \dots, x_n]$.

$G = (g_1, \dots, g_t)^T$: a Gröbner basis of I.

$F = (f_1, \dots, f_s)^T = AG$ where A is $s \times t$ matrix on $k[x_1, \dots, x_n]$

A has the left-inverse matrix $\Longrightarrow \langle F \rangle = \langle G \rangle$

Problem of Theorem 4.6

- A must have the left-inverse matrix (limited)
- t must be equal or less than s

We generalize this theorem to non-left-inverse matrix and the case s>max{t,n} with high probability

Overview of Slides

Random Polynomial

Generate Random Polynomial

Fix terms of polynomials and Choose radome coefficients, then get random polynomials within

- D: Upper degree bound
- (-C,C): Coeficients range

Example: Generate 2-degree 1-variable polynomial $ax^2 + bx + c$ with coefficients a, b, c chosen randomly in $(-10, 10)$:

- $x^2 + x + 1$
- $-2x^2 - 3x + 2$
- $5x^2 - 9x + 10$

We can also generate multi-variable polynomial:

$$ax^2 + by^2 + cxy + dx + ey + f$$

for $a, b, c, d, e, f \in (-C, C)$

Random Generation for bases of Ideals

Random Generation for bases of Ideals

For an ideal $I = \langle g_1, \ldots, g_t \rangle$ of $K[x_1, \ldots, x_n]$,
how to generate a random basis $\{f_1, \ldots, f_s\}$ of I?

$\Longrightarrow$ An efficient Generation gives us a lot of dataset for machine learning

Random Polynomial in Ideal

$$\langle g_1, \ldots, g_t \rangle = \left\{ \sum_{i=1}^{t} h_i g_i \mid h_i \in K[x_1, \ldots, x_n] \right\}$$

$\Longrightarrow$ $f \in I$ is written as $f = h_1 g_1 + \cdots + h_t g_t$

$\Longrightarrow$ Choose polynomials $h_1, \ldots, h_t$ randomly then get a random polynomial in I

Example 1

For $I = \langle x, y \rangle$, elements of I are expressed as

$$h_1 x + h_2 y$$

for $h_1, h_2 \in K[x, y]$. Then, choose h_1 and h_2 randomly

$$(x^2 + 2y + 3)x + (-xy + 2y)y$$

is a random polynomial in I.

- Random polynomials $f_1, \ldots, f_s$ in I may be a basis of I
- In this talk, if $s > n$, then $f_1, \ldots, f_s$ is a basis of I with almost 1 probability when I is zero-dimensional ideal

Setting and Conjecture

Setting

In $K[x_1, \ldots, x_n]$,

- $G = \{g_1, \ldots, g_t\}$, $\langle G \rangle$: zero-dimeosional.
- (s, t)-matrix $A \in K[x_1, \ldots, x_n]^{s \times t}$
- $AG = \{f_1, \ldots, f_s\}$

Consider $A \in K[x_1, \ldots, x_n]^{s \times t}$ s.t. $\langle AG \rangle = \langle G \rangle$.
$\Longrightarrow$ Can generate non-Gröbner basis AG from Gröbner basis G

Conjecture

- When $s < n$, $\langle AG \rangle \neq \langle G \rangle$.
- When $s = n$, $\langle AG \rangle \neq \langle G \rangle$ with almost 1 probability
- When $s > n$. $\langle AG \rangle = \langle G \rangle$ with almost 1 probability

Here, Coefficients in A are chosen randomly in $(-C, C)$

Partial Solution for the Conjecture

(1) When $s = n$, if $\langle G \rangle$ is radical and G is Gröbner basis, then the conjecture is solved in [1]

(2) When $s > n$, we solved the conjecture i.e. we can efficiently generate random basis AG of I with high probability

Abstract of Proof of (2)

- Consider coefficients in random polynomials as parameters and use the techniques in Comprehensive Gröbner basis
- Consider parametric (primary) decomposition $\langle \tilde{A}G \rangle = \langle G \rangle \cap J$ where $\tilde{A}$ is a parametric polynomial matrix
- Prove $\sigma_q(J) = K[x_1, \ldots, x_n]$ for q in a Zariski open set O where σ_q is the specialization map and $\langle AG \rangle = \langle G \rangle$ for generic $A = \sigma_q(\tilde{A})$.

[1] Kera et.al "Transformer Oracles for Border Basis Algorithms" (NeurIPS 2025)

Example 2

- Let $G = \{x, y\}$ and $A = \begin{pmatrix} a_1x + a_2y + a_3 & a_4x + a_5y + a_6 \\ a_7x + a_8y + a_9 & a_{10}x + a_{11}y + a_{12} \end{pmatrix}$.
  Then for random $a_1, \ldots, a_{12} \in (-C, C)$, $\langle AG \rangle \neq \langle G \rangle$ with almost 1 probability. For example,

$$\langle (x + y + 1)x + (2x - y)y,\ yx + 2y \rangle \neq \langle x, y \rangle$$

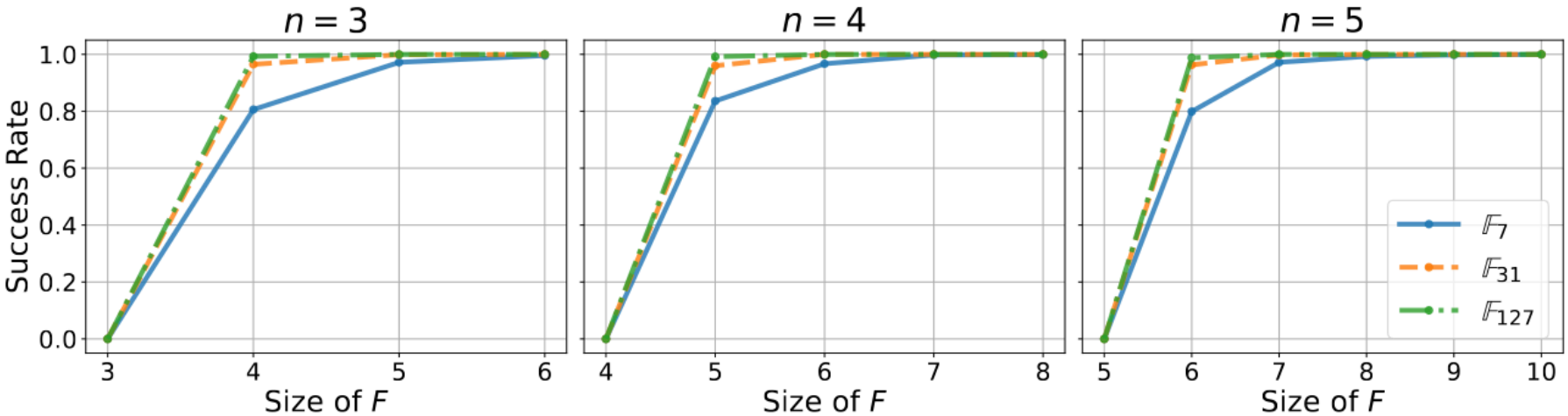
- Let $G = \{x, y\}$ and $A = \begin{pmatrix} a_1x + a_2y + a_3 & a_4x + a_5y + a_6 \\ a_7x + a_8y + a_9 & a_{10}x + a_{11}y + a_{12} \\ a_{13}x + a_{14}y + a_{15} & a_{16}x + a_{17}y + a_{18} \end{pmatrix}$

$$\begin{aligned} AG = \{&(a_1x + a_2y + a_3)x + (a_4x + a_5y + a_6)y, \\ &(a_7x + a_8y + a_9)x + (a_{10}x + a_{11}y + a_{12})y, \\ &((a_{13}x + a_{14}y + a_{15})x + (a_{16}x + a_{17}y + a_{18})y\} \end{aligned}$$

  Then for random $a_1, \ldots, a_{18} \in (-C, C)$, $\langle AG \rangle = \langle G \rangle$. For example,

$$\langle (x+y+1)x+(2x-y)y,\ yx+2y,\ (3x+2y-1)x+(x-y+2)y \rangle = \langle x, y \rangle$$

Experimental Results



- For each $n = 3, 4, 5$ and finite fileds $\mathbb{F}_7, \mathbb{F}_{31}, \mathbb{F}_{127}$, 1000 samples of AG were geenrated randomly
- If $|AG| = n$ then, the probability of $\langle AG \rangle = \langle G \rangle$ is 0
- If $|AG| > n$ then, the probability of $\langle AG \rangle = \langle G \rangle$ becomes close to 1 when the cardinality of $\mathbb{F}_p$ becomes large

[1] Kera et.al "Transformer Oracles for Border Basis Algorithms" (NeurIPS 2025)

Overview of Slides

Conclusion and Future Work

Conclusion

- Need an Efficient Dataset Generation to learn symbolic computations by Machine Learning
- Prove that it is easy to generate (with high probability) a random basis $\{f_1, \ldots, f_s\}$ of a 0-dimensional ideal I if $s > n$

Future Work

- Analyze the distribution of random bases
- Generalize the result to positive dimensional ideals

Thank you for your attention!