
SoS1: O1 and R1-Like Reasoning LLMs are Sum-of-Square Solvers

Anonymous Authors¹

Abstract

Large Language Models (LLMs) have achieved human-level proficiency across diverse tasks, but their ability to perform rigorous mathematical problem solving remains an open challenge. In this work, we investigate a fundamental yet computationally intractable problem: determining whether a given multivariate polynomial is nonnegative. This problem, closely related to Hilbert’s Seventeenth Problem, plays a crucial role in global polynomial optimization and has applications in various fields. First, we introduce **SoS-1K**, a meticulously curated dataset of approximately 1,000 polynomials, along with expert-designed reasoning instructions based on five progressively challenging criteria. Evaluating multiple state-of-the-art LLMs, we find that without structured guidance, all models perform only slightly above the random guess baseline (50%). However, high-quality reasoning instructions significantly improve accuracy—boosting performance up to 85.6%. Furthermore, our 7B model, SoS-7B, fine-tuned on SoS-1K for just 4 hours, outperforms the 671B DeepSeek-V3 and GPT-4o-mini in accuracy while only requiring 1.8% and 5% of the computation time needed for letters, respectively. Our findings highlight the potential of LLMs to push the boundaries of mathematical reasoning and tackle NP-hard problems.

¹Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1 Introduction

With Large Language Models (LLMs) reaching human-level proficiency across a diverse range of tasks (Brown et al., 2020; Singhal et al., 2023; Cai et al., 2023; Yoshikawa et al., 2023), their ability to reason has emerged as a central topic of interest (Wei et al., 2022; Huang & Chang, 2022). Among these, mathematical reasoning stands out as one of the most rigorous and demanding (Kant, 1908; Hendrycks et al., 2021; Ahn et al., 2024; Liu et al., 2024). As a result, the ability of LLMs to solve research-level mathematical problems is not only a critical benchmark for evaluating their reasoning capabilities but also has the potential to transform mathematical research and practice.

Demonstrated by the success of OpenAI o1 (OpenAI, 2024) and DeepSeek-R1 (Guo et al., 2025), test-time scaling has emerged as a promising technique for enhancing LLMs’ performance in mathematical reasoning (Snell et al., 2024a; Welleck et al., 2024). This approach involves prompting LLMs to generate more reasoning steps, either sequentially (Snell et al., 2024b; Hou et al., 2025; Lee et al., 2025) or in parallel (Brown et al., 2024; Xin et al., 2024), to increase the accuracy of their final answers. However, the community’s primary focus has been limited to relatively simple levels of mathematics, ranging from high school and Olympiad-level problems to early undergraduate topics (of America, 2024; Hendrycks et al., 2021). Whether the promise of test-time scaling extends to research-level mathematics remains an open question.

In this paper, we investigate a fundamental yet formally well-posed problem in mathematics: determining whether a given multivariate polynomial is nonnegative. This question is closely related to Hilbert’s Seventeenth Problem, which was posed by David Hilbert in 1900 as part of his famous 23 problems presented at the International Congress of

Mathematicians (ICM) (Hilbert, 1893), and it remains central to global polynomial optimization. Many key challenges in applied and computational mathematics can be reframed as deciding the nonnegativity of certain polynomials including control theory (Parrilo, 2000), quantum computation (Doherty et al., 2002), polynomial games (Gvozdenović & Laurent, 2007), tensor methods (Zhu & Cartis, 2024; Ahmadi et al., 2023a) and combinatorial optimization (Gvozdenović & Laurent, 2007).

Testing whether a general polynomial is nonnegative is provably NP-hard, even for polynomials of relatively low degrees or with a small number of variables. People usually seek special cases of polynomials where the challenging nonnegativity constraints can be replaced with more manageable conditions. For instance, the sum of squares (SoS) condition, a mathematical technique in polynomial optimization where a polynomial is expressed as a sum of squared polynomials, provides a sufficient criterion for polynomial nonnegativity. Classical solvers, such as SoSTOOLS, YALMIP, and Gloptipoly have been developed to verify these SoS conditions (Prajna et al., 2002). However, a significant limitation of these approaches lies in the typically large size of the resulting semidefinite programming (SDP) problem. Specifically, for a polynomial with n variables and a degree of $2d$, the SDP’s dimension is given by $N = \binom{n+2d}{2d}$, making it challenging to scale this approach to larger problems.

To evaluate if state-of-the-art (SOTA) reasoning LLMs like Openai o1 and DeepSeek-R1 can solve large-scale SoS programming problems, we introduce **SoS-1K**, a meticulously curated dataset of approximately 1,000 polynomials—along with expert-designed, SoS-specialized, reasoning-guiding instructions based on five progressively challenging criteria: polynomial degree, nonnegativity of the leading search direction, identification of special structures, assessment of square-form expressions, and matrix decomposition into the quadratic form of monomials. Our comprehensive evaluation of multiple SOTA LLMs, including DeepSeek-R1, DeepSeek-V3, GPT-4o, OpenAI o1-mini, Qwen2.5 series, and QwQ-32B-Preview, demonstrate the following interesting findings:

- When presented with a plain question, all SOTA LLMs bluntly fail to solve SoS with most models achieving

only around 60% accuracy, just slightly above the random guess baseline of 50%.¹

- When prompted with high-quality reasoning traces, we consistently observe a significant accuracy boost across all models, up to 21%. And models perform better with higher-quality reasoning traces.
- Reasoning-focused LLMs generally outperform general-purpose LLMs, regardless of prompt quality.
- Higher-capacity models require fewer thinking tokens to make correct predictions, while lower-capacity models need more reasoning steps to achieve their optimal performance.

We further demonstrate that supervised fine-tuning (SFT) of a pre-trained 7B model on SoS-1K for just 4 hours using 2 A100 GPUs significantly improves accuracy from 54% to 70% with significantly faster response times. Specifically, SoS-7B requires only 1.8% and 5% of the computation time needed for DeepSeek-V3 and GPT-4o-mini, respectively. The resulting SoS-7B surpasses much larger models, including the 671B DeepSeek-V3 and GPT-4o-mini. More interestingly, when prompted with high-quality reasoning, the models demonstrate an understanding of research-level questions. For instance, Qwen2.5-14B-Instruct-1M leverages the Motzkin polynomial to generate new, previously unseen counterexamples to Hilbert’s 17th problem (Motzkin, 1967). Such examples are nontrivial, as the first counterexample to Hilbert’s 17th problem was discovered 27 years after Hilbert originally posed it (Hilbert, 1893). These findings suggest that LLMs exhibit reasoning patterns, expanding the boundaries of solving NP-hard problems.

Our work serves as a pilot study on leveraging reasoning LLMs for solving SoS problems, paving the way for tackling large-scale research-level questions in mathematics using AI.

1.1 Related Work

The study of SoS and nonnegative polynomials has a rich history spanning over 120 years, with numerous scholars

¹Since SoS is a binary classification problem, random guessing yields 50% accuracy.

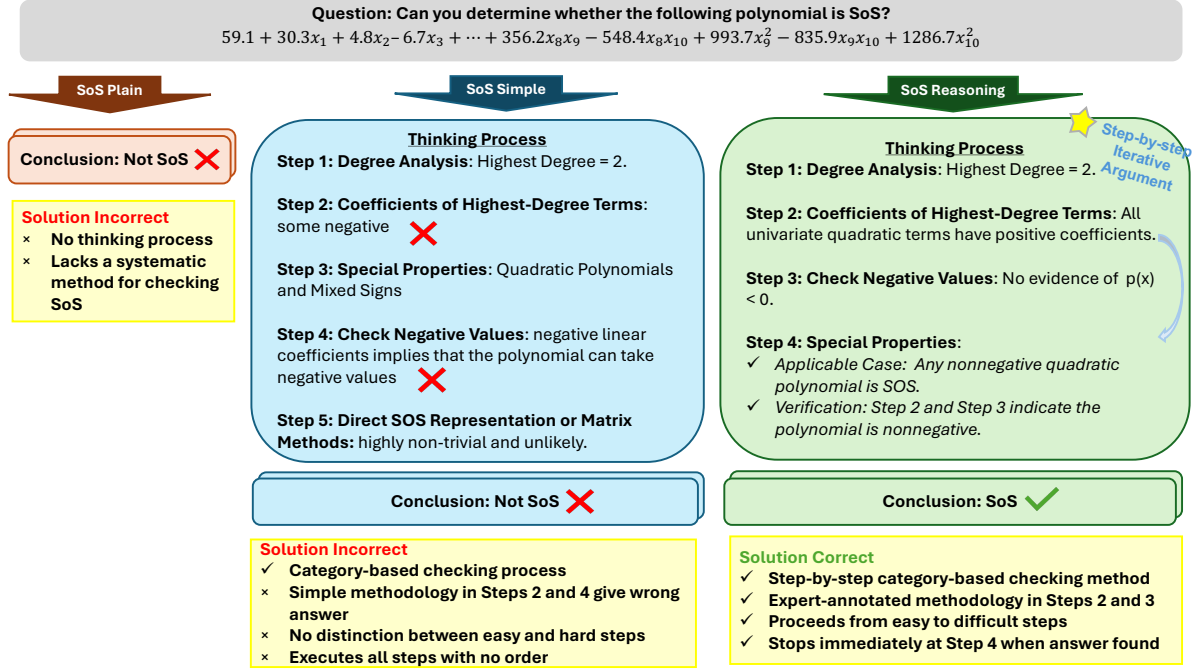


Figure 1: Demonstration of SoS Plain (left), SoS Simple (mid), and SoS Reasoning (right).

contributing to this field. Classical methods for characterizing an SoS polynomial involve expressing the polynomial as a quadratic form of monomials and then reformulating the problem as a semidefinite program (SDP) (Lasserre, 2000; 2001). Classic techniques, such as the Lasserre hierarchy, date back to 2001 (Lasserre, 2001). The development of numerical solvers began in 2009, with tools like GloptiPoly and SoSTOOLS introduced in (Henrion et al., 2009; Papachristodoulou et al., 2013). More details on the literature review for SoS can be found in Appendix A.

The first attempt to tackle SoS-related mathematical difficulty via AI was presented in (Alfarano et al., 2024). The authors trained Transformers to address a long-standing open problem in mathematics: discovering a Lyapunov function that guarantees the global stability of a dynamical system. The global stability of polynomial Lyapunov systems is closely tied to SoS framework. Their approach was tested on relatively small systems (with at most five equations for polynomial systems) and demonstrated promising results compared to state-of-the-art conventional solvers.

2 SoS-1K Dataset

In this section, we first provide the definition of SoS polynomials. We describe the expert-annotated reasoning traces based on five criteria of increasing difficulty in Section 2.1, which lead to the generation of the SoS-1K dataset, detailed in Section 2.2.

Definition 2.1. SoS polynomial (Ahmadi et al., 2023a; Ahmadi & Parrilo, 2013; Kojima, 2003) A $2d$ -degree multivariate polynomial $q(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ where $x = [x_1, \dots, x_n] \in \mathbb{R}^n$ is a sum of squares (SoS) if there exist polynomials $\tilde{q}_1, \dots, \tilde{q}_r : \mathbb{R}^n \rightarrow \mathbb{R}$, for some $r \in \mathbb{N}$, such that $q(\mathbf{x}) = \sum_{j=1}^r \tilde{q}_j(\mathbf{x})^2$ for all $\mathbf{x} \in \mathbb{R}^n$ (?)Def. 1]ahmadi2023higher.

Further details on the theory for SoS polynomials can be found in Appendix B.

2.1 Expert-Designed Reasoning Instructions for SoS

Instructions play a crucial role in guiding LLMs toward better reasoning and problem-solving (Zhang et al., 2023). The way an LLM processes a problem can be significantly improved with carefully crafted instructions that provide

structure, constraints, and logical flow.

To evaluate the capacity of SOTA LLMs on SoS reasoning, we create three sets of reasoning-guiding instructions with increasing quality that can be applied across multiple SoS problem types: (1) plain question (SoS Plain); (2) simple SoS instruction (SoS Simple); (3) reasoning-guiding SoS instruction (SoS Reasoning).

SoS Plain simply asks LLMs: “Please analyze if this polynomial can be expressed as a Sum of Squares (SOS)”.

SoS Simple classifies SoS polynomials into five distinct groups, each defined by a concise, one-line criterion. The full instruction set for SoS Simple contains 78 words and 647 characters, with complete details provided in Appendix C.

SoS Reasoning is a structured five-step framework designed to identify SoS polynomials. Unlike SoS Simple, which provides only basic classification criteria, SoS Reasoning encourages the model follow a step by step mathematical verification process. The framework introduces progressively more detailed reasoning steps to guide the model in verifying whether a polynomial is SoS. Specifically, we provide a logical reasoning trace based on proofs and theorems, offering necessary and sufficient conditions for identifying SoS polynomials. A large number of positive and negative examples accompany each set of theorems, helping the model recognize special structures, symmetries, and mathematical forms inherent to SoS polynomials. Additionally, SoS Reasoning introduces intermediate steps and incorporates key and challenging reasoning processes, such as the \mathbf{Q} matrix and the squared form p_s , enabling deeper reasoning and iterative refinement.

Below is an illustration of SoS Reasoning. The full version is provided in Appendix D.

Step 1. Check the Degree: An SoS polynomial must have an even highest degree.

Step 2. Check for Non-negativity: SoS polynomials are nonnegative for all real inputs. We verify this by examining the constant term, the coefficients of the leading term, and performing a grid-based numerical check.

Step 3. Check for Well-known Special Cases: Any non-

negative quadratic polynomial and any nonnegative quartic polynomial in one or two variables is SoS.

Step 4. Check for Square Form: By Definition 2.1, an SoS polynomial can be expressed as: $p_s(\mathbf{x}) = \sum_i q_i(\mathbf{x})^2$, where each $q_i(\mathbf{x})$ is a polynomial.

Step 5. Check for Matrix Decomposition: Based on Theorem B.2, we express the polynomial as $p(\mathbf{x}) = \mathbf{y}^{*\top} \mathbf{Q} \mathbf{y}^*$, where \mathbf{Q} is a symmetric matrix². We then check whether \mathbf{Q} is positive semidefinite.

2.2 Construction of SoS-1K Test Set

Building on the above expert design, we construct SoS-1K, a dataset comprising five subsets of polynomials, each corresponding to polynomials filtered out at steps 1–5. Appendix E provides a comprehensive summary of the SoS-1K test set and its subsets. Since most LLMs struggle with very long polynomials, we have ensured that the majority remain within a length of 4,000. Approximately half of the polynomials are SoS, while the other half are not. The number of variables and the polynomial degree both range from 2 to 10. For all test subsets (except Set 1), we provide two corresponding sets: one containing SoS polynomials and another containing non-SoS polynomials. Each polynomial is labeled as either SoS or non-SoS, with accompanying justifications and difficulty levels. For certain polynomial classes, we provide theoretical proofs confirming their SoS status, while the remaining polynomials are labeled based on results from standard solvers.

3 Evaluation of LLMs on SoS

In this section, we compare the performance of SOTA LLMs using SoS Plain, SoS Simple, and SoS Reasoning on a subset of SoS-1K of approximately 340 randomly chosen from all sub-classes of test problems (see Table 3). These test samples are drawn such that the number of samples in each subtest is approximately equal. We test across sub-classes and report results per test subclasses. The models evaluated include reasoning-purpose models like DeepSeek-R1, OpenAI o1-mini, and QwQ-32B-Preview, as well as general-purpose models such as DeepSeek-V3, GPT-4o, and

² \mathbf{Q} and \mathbf{y}^* are provided in Appendix D

Model	Accuracy on Valid Samples			Accuracy on Total Samples			Response Time (s)		
Instruction Type	SoS Plain	SoS Simple	SoS Reasoning	SoS Plain	SoS Simple	SoS Reasoning	SoS Plain	SoS Simple	SoS Reasoning
General-purpose LLMs									
Qwen2.5-7B-Instruct	55%	61%	76%	52%	59%	62%	22.4	31.2	68.5
Qwen2.5-7B-Instruct-1M	54%	64%	75%	54%	64%	63%	5.6	8.4	35.2
Qwen2.5-14B-Instruct	55%	66%	74%	52%	66%	69%	12.9	23.1	48.3
Qwen2.5-14B-Instruct-1M	56%	60%	74%	56%	59%	67%	12.7	20.7	52.7
Qwen2.5-32B-Instruct	56%	58%	67%	55%	58%	62%	13.0	18.0	37.4
Qwen3-8B-nothinking	56.5%	61.2%	69.2%	56.5 %	60.9%	67.4 %	14.1	33.75	40.2
Qwen3-32B-nothinking	63%	63.5%	72.8%	63 %	63.5 %	72.6%	22.4	39.11	44
DeepSeek-V3	54%	60%	70%	54%	60%	69%	29.6	39.8	95.0
GPT-4o-mini	59%	67%	72%	59%	67%	69%	10.8	15.4	53.1
GPT-4o	60%	61%	75%	59%	61%	75%	14.6	16.2	27.8
Reasoning-purpose LLMs									
QwQ-32B-Preview	64%	71%	79%	44%	54%	52%	105.7	101.8	100.0
Qwen3-8B-thinking	70%	73.9%	75%	70%	73.2%	75%	238	288.5	222
OpenAI o1-mini	58%	61%	77%	57%	61%	76%	8.3	18.1	34.9
DeepSeek-R1	62%	62%	81%	55%	55%	56%	514.5	565.6	492.5
Qwen3-32B-thinking	78.3%	79%	85.6%	77.3%	78.5%	85.6%	276	310	185.5
Average	57%	63%	75%	54%	60%	65%	68.2	78.0	95.0

Table 1: Accuracy comparison across various SOTA LLMs on a subset of SoS-1K with 340 samples. Results are divided into “Valid Samples” and “Total Samples”, as we found that LLMs sometimes suffer from timeout issues.

Qwen2.5 series.

The full results for each model on each test set are summarized in Table 1. We summarize our main observations below:

OB1. All LLMs fail when presented with a plain question. When given SoS-plain, all LLMs exhibit poor accuracy, ranging from 50% to 60%, with QwQ-32B-Preview being the sole exception, achieving 64% valid accuracy. This result suggests that, despite being trained on vast amounts of mathematical data, SOTA LLMs struggle to solve SoS problems without explicit prompting.

OB2. LLMs have a significant performance boost when prompted with high-quality reasoning-guiding instructions. We consistently observe a substantial accuracy improvement across all models. With SoS Simple, QwQ-32B-Preview achieves 71% accuracy, while DeepSeek-R1 with the highest-level SoS Reasoning reaches the highest accuracy (81%). It suggests that while LLMs may possess the underlying knowledge to solve SoS problems, they require clear and structured instructions to effectively retrieve and

apply it.

Furthermore, it is worth mentioning that SoS Reasoning plays a crucial role in achieving these improvements. The improvement from SoS Simple over the baseline (SoS Plain) is relatively small, averaging 5% increase in the accuracy of valid samples and 6% increase in the accuracy of valid samples, whereas SoS Reasoning improves performance by 17% and 11%, respectively over the same baseline.

OB3. Reasoning-purpose LLMs benefit more from high-quality instructions than general-purpose LLMs. Overall, reasoning-focused LLMs such as DeepSeek-R1, OpenAI o1-mini, and QwQ-32B-Preview achieve a higher average accuracy (79.0%) compared to general-purpose LLMs (72.9%). This result suggests that stronger reasoning capabilities contribute to improved performance in solving SoS problems.

OB4. Many LLMs struggle to consistently provide valid answers to SoS questions. Most LLMs fail to consistently provide valid answers, often encountering timeout issues. Nevertheless, GPT-4o, DeepSeek-V3, and o1-mini demon-

strate robustness in this regard, consistently producing effective and reliable answers.

4 Further Analysis

In this section, we outline several research questions that the authors find particularly intriguing.

Q1: Does the model follow a truly mathematical step by step verification process?

We find that LLMs are able to generate answers that are both logically and mathematically correct, step-by-step, following our SoS Reasoning instruction. For instance, we demonstrate o1-mini’s response in Appendix F where we can observe that the responses are logically and mathematically correct, and the model stops naturally once it derives an answer, rather than blindly going through all possible steps.

Q2: Can LLMs effectively retrieve critical information from long-context polynomials?

Unlike standard text input, polynomials are complex algebraic expressions consisting of variables, coefficients, exponents, and terms. Thus, it is crucial for LLMs to effectively interpret and extract critical information from such structured formats. Our analysis reveals that while QwQ-32B-Preview struggles with questions exceeding 4K tokens in length, most SOTA LLMs can successfully extract the necessary coefficients from 4K-length polynomials for evaluation, producing correct answers.

Q3: At which of the Steps 1 through 5 in SoS Reasoning does the accuracy improve?

In Figure 2, we illustrate the accuracy improvement across different test sets for the o1-mini model under SoS Plain, SoS Simple, and SoS Reasoning.³ We observe that the simplest test set, Test Set 1 (corresponding to Step 1), unsurprisingly achieves 100% accuracy across all prompting methods. For the more challenging test sets, Test Sets 2a, 3.1a, 5.1a–5.4a, we observe a continuous improvement from SoS Plain to SoS Simple and further to SoS Reasoning. This improvement is attributed to Steps 2 and 5 in SoS Reasoning, where a series of mathematical verification methods for

non-negativity are introduced, including constant coefficient check, grid evaluation, leading order and dominant terms comparison, finding minima, matrix decomposition, and finding symmetry and translation.

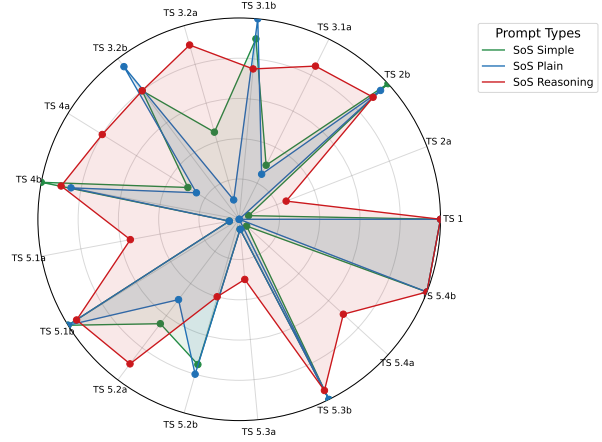


Figure 2: Accuracy of different test sets using o1-mini.

Q4. Will LLMs become lazy (take shortcuts) during reasoning?

Yes, another interesting phenomenon observed under the SoS Reasoning prompt is that the model tends to be lazy in Step 5. Specifically, instead of fully executing Step 5, it often avoids matrix decomposition or semidefinite programming (SDP) due to complexity and instead guesses an answer based on prior steps. This behavior is particularly prevalent for long inputs and complex polynomials, such as those in Test Set 5.4a. For simpler problems, reasoning models such as o1-mini (which had the shortest runtime of 17s) and larger models like QwQ-32B-Preview tend to take shortcuts, skipping Step 5 and inferring the answer from earlier, simpler steps. In contrast, DeepSeek-V3 does not take shortcuts and instead spends significantly more time solving all steps properly (40s).

Q5: How does reasoning length affect accuracy? Figure 3 shows that higher-capacity models generally require fewer thinking tokens to make correct predictions, whereas lower-capacity models need more reasoning steps to reach optimal performance. For instance, DeepSeek-R1 and o1-mini achieve the highest number of correct predictions with a 1K-2K response length, whereas the Qwen2.5 series require 3K–4K tokens to produce correct answers.

³Similar patterns are observed for other models.

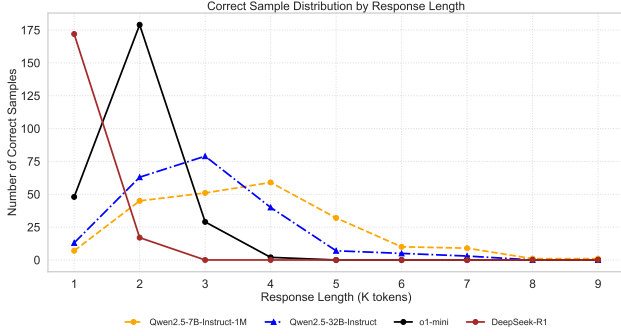


Figure 3: Number of correct samples with various response lengths.

Q6: Do SOTA LLMs have any limitations? Though we demonstrate that SoS Reasoning effectively improves accuracy, it is subject to the following limitations. Firstly, for long input cases, invalid samples occur. For example, in DeepSeek-R1, only 234 out of 340 samples were valid. Secondly, when handling complex problems, "taking shortcuts" may save time; however, stopping prematurely at difficult steps and guessing an answer can negatively impact prediction accuracy. Thirdly, while these LLMs excel on small-sized polynomials (achieving accuracy close to 90%), they struggle in cases where the quadratic form of the polynomial involves a low-rank matrix decomposition.

5 Performing SFT on SoS-1K

We further conduct supervised fine-tuning (SFT) with Qwen2.5-7B-Instruct-1M on SoS-1K using LLaMA-Factory (Zheng et al., 2024). The training process was performed on 2× NVIDIA A100 GPUs for 4 hours. The resulting model, SoS-7B, establishes a SOTA total accuracy of 70%, outperforming 671B DeepSeek-V3 (69%), while requiring only 1.8 seconds response time compared to DeepSeek-V3’s 100 seconds. While Qwen3-32B-thinking achieves a higher accuracy (85.6%), it is acceptable as our model is only trained with 1K dataset and enjoys a much faster response time, i.e., 1.8s vs 34.9s.

6 Model’s Understanding of SoS and Nonnegativity

One might ask: Did the model merely learn to classify, or has it truly developed the ability to “think” and “construct”

Table 2: Accuracy Comparison on SoS Reasoning Benchmark, where “—” denotes the undisclosed model size. Accuracy is measured on full evaluation samples.

Model	Size	Acc. (%)
<i>Closed Source</i>		
GPT-4o	—	75
o1-mini	—	76
<i>Open Source</i>		
Qwen2.5-7B-Instruct-1M	7B	63
Qwen2.5-32B-Instruct	32B	62
QwQ-32B-Preview	32B	52
DeepSeek-V3	671B	69
DeepSeek-R1	671B	56
Qwen3-8B-thinking	8B	75
Qwen3-32B-thinking	32B	85.6
SoS-7B (Ours)	7B	70

new proofs and examples? When faced with research questions in SoS or polynomial optimization, can the model generate mathematically meaningful insights?

To explore this, we designed a series of research-driven questions (Appendix H) to test the model’s ability to understand the mathematical concepts behind Sum of Squares (SoS) and nonnegativity properties. Finding nonnegative polynomials that are not Sum of Squares (NNSoS) is a fundamental and ongoing research problem in real algebraic geometry and polynomial optimization (Ahmadi et al., 2023b; 2024; Ahmadi & Zhang, 2022). This problem is closely connected to Hilbert’s 17th problem, semidefinite programming (SDP), and positivity certificates in polynomial optimization.

We test the model’s ability to create and analyze unseen mathematical examples. We asked the following question to Qwen-7B-1M and Qwen-14B-1M: *Can you provide a new NNSoS that was never found in the literature?*

Interestingly, when prompted with SoS plain, Qwen-14B-1M can only give examples that are well-known in the literature and Qwen-7B-1M returned an incorrect example:

$$p_a(\mathbf{x}) = x_1^4 + x_2^4 + 1 - x_1^2 - x_2^2 - x_1^2 x_2^2.$$

Although this example is incorrect, it is nontrivial, as classic

solvers such as YALMIP also fail to extract global optimality⁴. The reason this example is challenging is that it has four global minima at $(1, 1)$, $(1, -1)$, $(-1, 1)$, and $(-1, -1)$, with a global minimum value of 0. The number of global minima exceeds the rank of the moment matrix, making it difficult to extract global optimality certificates.

However, when the same research question was posed to the model with SoS reasoning, the model correctly identified that p_a is not a valid solution to our question. This improvement can be attributed to Step 4 of SoS Reasoning (Appendix D), where the trained model recognized that $p_a(\mathbf{x})$ is a nonnegative quartic polynomial in two variables and, therefore, cannot be NNSoS.

Moreover, using SoS reasoning, Qwen-14B-1M derived a new valid example for NNSoS,

$$q_a(\mathbf{x}) = x_1^4 x_2^2 x_3^2 + x_1^2 x_2^4 x_3^2 + x_3^4 + 1 - 3x_1^2 x_2^2 x_3^2.$$

We cross-checked this polynomial using the classic solver YALMIP and confirmed that q_a is indeed a NNSoS. The trained model’s approach to constructing this example is particularly interesting. It began with the well-known example of NNSoS such as $p_m(\mathbf{x}) = x_1^4 x_2^2 + x_1^2 x_2^4 + 1 - 3x_1^2 x_2^2$. Then, the model then introduced a new variable and slightly modified the coefficients to generate q_a . This demonstrates that the trained model not only recognizes existing patterns in polynomial optimization but also generalizes and constructs novel cases, providing valuable mathematical insights.

7 Conclusion and Discussion

This paper investigates the capacity of LLMs on a research-level mathematical problem: determining whether a given multivariate polynomial is SoS. This problem, closely related to Hilbert’s Seventeenth Problem, plays a crucial role in various fields. We first introduce SoS-1K, a dataset of approximately 1,000 polynomials, along with a set of SoS-specific reasoning-guiding instructions. Our results show that with our expert-designed, current SOTA LLMs is able to achieve up to 81% accuracy. Furthermore, an in-depth analysis of model responses reveals several intriguing insights. This study highlights the potential of AI in tackling

⁴YALMIP returns status 0, indicating that although the SDP is solvable, rank conditions cannot ensure global optimality.

large-scale open problems in mathematics, paving the way for future advancements.

8 Limitation and Potential Risks

One limitation of this study is that we constrain the context length of SoS question within 4K, as some LLMs tend to fail with longer sequence length. Due to this reason, most SoS polynomials are within the capacity of traditional solvers. In the future, we will extend our dataset, targeting more challenging questions with a scale that is beyond traditional SoS solvers. One potential risk of this study is that LLMs have the chance to make wrong decisions, which might be misleading, and therefore we need to use traditional solvers to verify.

References

- Ahmadi, A. A. and Majumdar, A. Dsos and sdsos optimization: more tractable alternatives to sum of squares and semidefinite optimization. *SIAM Journal on Applied Algebra and Geometry*, 3(2):193–230, 2019.
- Ahmadi, A. A. and Parrilo, P. A. A complete characterization of the gap between convexity and sos-convexity. *SIAM Journal on Optimization*, 23(2):811–833, 2013.
- Ahmadi, A. A. and Zhang, J. On the complexity of finding a local minimizer of a quadratic function over a polytope. *Mathematical Programming*, 195(1-2):783–792, 2022.
- Ahmadi, A. A., Olshevsky, A., Parrilo, P. A., and Tsitsiklis, J. N. Np-hardness of deciding convexity of quartic polynomials and related problems. *Mathematical programming*, 137:453–476, 2013.
- Ahmadi, A. A., Chaudhry, A., and Zhang, J. Higher-order newton methods with polynomial work per iteration. *arXiv preprint arXiv:2311.06374*, 2023a.
- Ahmadi, A. A., Dibek, C., and Hall, G. Sums of separable and quadratic polynomials. *Mathematics of Operations Research*, 48(3):1316–1343, 2023b.
- Ahmadi, A. A., Blekherman, G., and Parrilo, P. A. Convex ternary quartics are sos-convex. *arXiv preprint arXiv:2404.14440*, 2024.

- Ahn, J., Verma, R., Lou, R., Liu, D., Zhang, R., and Yin, W. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*, 2024.
- Alfarano, A., Charton, F., and Hayat, A. Global Iyapunov functions: a long-standing open problem in mathematics, with symbolic transformers. *arXiv preprint arXiv:2410.08304*, 2024.
- Brown, B., Juravsky, J., Ehrlich, R., Clark, R., Le, Q. V., Ré, C., and Mirhoseini, A. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.
- Cai, T., Wang, X., Ma, T., Chen, X., and Zhou, D. Large language models as tool makers. *arXiv preprint arXiv:2305.17126*, 2023.
- Camps, O. and Sznaiar, M. The interplay between big-data and sparsity in systems identification. In Laumond, J.-P., Mansard, N., and Lasserre, J.-B. (eds.), *Geometric and Numerical Foundations of Movements*, number 117 in Springer Tracts in Advanced Robotics, pp. 133–159. Springer, New York, 2017.
- Doherty, A. C., Parrilo, P. A., and Spedalieri, F. M. Distinguishing separable and entangled states. *Physical Review Letters*, 88(18):187904, 2002.
- Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Gvozdenović, N. and Laurent, M. Semidefinite bounds for the stability number of a graph via sums of squares of polynomials. *Mathematical Programming*, 110:145–173, 2007.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- Henrion, D., Lasserre, J.-B., and Löfberg, J. Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, 24(4-5):761–779, 2009.
- Hilbert, D. Über ternäre definite formen. *Journal*, 1893.
- Hou, Z., Lv, X., Lu, R., Zhang, J., Li, Y., Yao, Z., Li, J., Tang, J., and Dong, Y. Advancing language model reasoning through reinforcement learning and inference scaling, 2025. URL <https://arxiv.org/abs/2501.11651>.
- Huang, J. and Chang, K. C.-C. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.
- Kant, I. Critique of pure reason. 1781. *Modern Classical Philosophers*, Cambridge, MA: Houghton Mifflin, pp. 370–456, 1908.
- Kojima, M. *Sums of squares relaxations of polynomial semidefinite programs*. Inst. of Technology, 2003.
- Lasserre, J., Toh, K.-C., and Yang, S. A bounded degree sos hierarchy for polynomial optimization. *EURO Journal on Computational Optimization*, 5(1-2):87–117, 2017.
- Lasserre, J. B. Optimisation globale et théorie des moments. *Comptes Rendus de l’Académie des Sciences-Series I-Mathematics*, 331(11):929–934, 2000.
- Lasserre, J. B. Global optimization with polynomials and the problem of moments. *SIAM Journal on optimization*, 11(3):796–817, 2001.
- Lasserre, J.-B. Convergent sdp-relaxations in polynomial optimization with sparsity. *SIAM Journal on Optimization*, 17(3):822–843, 2006.
- Lee, K.-H., Fischer, I., Wu, Y.-H., Marwood, D., Baluja, S., Schuurmans, D., and Chen, X. Evolving deeper llm thinking, 2025. URL <https://arxiv.org/abs/2501.09891>.

- Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- Luo, Z.-Q. and Zhang, S. A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints. *SIAM Journal on Optimization*, 20(4):1716–1736, 2010.
- Molzahn, D. K. and Hiskens, I. K. Sparsity-exploiting moment-based relaxations of the optimal power flow problem. *IEEE Transactions on Power Systems*, 30:3168–3180, 2015.
- Motzkin, T. S. The arithmetic-geometric inequality. *Inequalities (Proc. Sympos. Wright-Patterson Air Force Base, Ohio, 1965)*, 205:54, 1967.
- Murty, K. G. and Kabadi, S. N. Some np-complete problems in quadratic and nonlinear programming. *Mathematical Programming: Series A and B*, 39(2):117–129, 1987.
- of America, M. A. Aime, February 2024. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions/.
- OpenAI. Learning to reason with llms, September 2024. URL <https://openai.com/index/learning-to-reason-with-llms/>.
- Papachristodoulou, A., Anderson, J., Valmorbidia, G., Prajna, S., Seiler, P., Parrilo, P., Peet, M. M., and Jagt, D. Sostools version 4.00 sum of squares optimization toolbox for matlab. *arXiv preprint arXiv:1310.4716*, 2013.
- Parrilo, P. A. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, California Institute of Technology, 2000.
- Parrilo, P. A. and Sturmfels, B. Minimizing polynomial functions. *arXiv preprint math/0103170*, 2001.
- Prajna, S., Papachristodoulou, A., and Parrilo, P. A. Introducing sostools: A general purpose sum of squares programming solver. In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002., volume 1, pp. 741–746. IEEE, 2002.
- Singhal, K., Azizi, S., Tu, T., Mahdavi, S. S., Wei, J., Chung, H. W., Scales, N., Tanwani, A., Cole-Lewis, H., Pfohl, S., et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.
- Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024a.
- Snell, C., Lee, J., Xu, K., and Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024b. URL <https://arxiv.org/abs/2408.03314>.
- Waki, H., Kim, S., Kojima, M., and Muramatsu, M. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM Journal on Optimization*, 17(1):218–242, 2006.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Weisser, T., Lasserre, J. B., and Toh, K.-C. Sparse-bsos: a bounded degree sos hierarchy for large scale polynomial optimization with sparsity. *Mathematical Programming Computation*, 10:1–32, 2018.
- Welleck, S., Bertsch, A., Finlayson, M., Schoelkopf, H., Xie, A., Neubig, G., Kulikov, I., and Harchaoui, Z. From decoding to meta-generation: Inference-time algorithms for large language models. *arXiv preprint arXiv:2406.16838*, 2024.
- Xin, H., Guo, D., Shao, Z., Ren, Z., Zhu, Q., Liu, B., Ruan, C., Li, W., and Liang, X. Deepseek-prover: Advancing theorem proving in llms through large-scale synthetic data. *arXiv preprint arXiv:2405.14333*, 2024.
- Yoshikawa, N., Skreta, M., Darvish, K., Arellano-Rubach, S., Ji, Z., Bjørn Kristensen, L., Li, A. Z., Zhao, Y., Xu, H., Kuramshin, A., et al. Large language models for chemistry robotics. *Autonomous Robots*, 47(8):1057–1086, 2023.

- Zhang, S., Dong, L., Li, X., Zhang, S., Sun, X., Wang, S., Li, J., Hu, R., Zhang, T., Wu, F., et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.
- Zheng, Y., Fantuzzi, G., and Papachristodoulou, A. Sparse sum-of-squares (sos) optimization: A bridge between dsos/sdsos and sos optimization for sparse polynomials. In *2019 American Control Conference (ACC)*, pp. 5513–5518. IEEE, 2019.
- Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., and Ma, Y. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL <http://arxiv.org/abs/2403.13372>.
- Zhu, W. and Cartis, C. Global convergence of high-order regularization methods with sums-of-squares taylor models. *arXiv preprint arXiv:2404.03035*, 2024.

A Literature Review for SoS

The problem of determining whether a multivariate polynomial is nonnegative is inherently linked to the task of finding its global minimum—a fundamental challenge in the optimization community (Ahmadi et al., 2023a; Zhu & Cartis, 2024; Lasserre, 2000; Parrilo & Sturmfels, 2001). Testing whether a general polynomial is nonnegative is provably NP-hard, even for polynomials of relatively low degrees or with a small number of variables. For instance, it has been shown that finding the global minimum of general even-degree polynomials of degree at least four is NP-hard (Ahmadi et al., 2023a; 2013; Ahmadi & Zhang, 2022; Murty & Kabadi, 1987).

Due to the computational intractability of the general problem, we seek special cases of polynomials where the challenging nonnegativity constraints can be replaced with more manageable conditions. The sum of squares (SoS) condition, a mathematical technique in polynomial optimization where a polynomial is expressed as a sum of squared polynomials, provides a sufficient criterion for polynomial

nonnegativity. The SoS property is particularly useful because it allows the nonnegativity problem to be reformulated as a semidefinite programming (SDP) problem, for which efficient algorithms, such as interior-point methods, exist. In certain special cases, nonnegativity and SoS are equivalent; for example, any nonnegative quadratic polynomial or any nonnegative even-degree univariate polynomial can always be expressed as a sum of squares (Hilbert, 1893; Ahmadi & Parrilo, 2013; Ahmadi et al., 2023b). For more complex polynomials, the Lasserre hierarchy provides a systematic way to approximate nonnegativity using a sequence of SoS relaxations (Lasserre, 2001). This method constructs a sequence of SDP problems that yield increasingly tighter approximations to nonnegativity.

Many large-scale problems exhibit structured sparsity patterns, enabling the application of a sparsity-adapted hierarchy of SDP relaxations (Camps & Sznaiar, 2017; Lasserre, 2006; Molzahn & Hiskens, 2015; Waki et al., 2006). Additional techniques for addressing large-scale problems include Structured DSoS and SDSos programming, as well as Bounded Degree SoS (BSoS) (Ahmadi & Majumdar, 2019; Lasserre et al., 2017; Waki et al., 2006; Weisser et al., 2018; Zheng et al., 2019). These approaches take advantage of the structure of the problem (sparsity) to generate smaller SDPs. There are also methods to reformulate the original optimization problem to reduce the size of the optimization. For instance, the optimization of a multivariate fourth-order (quartic) homogeneous polynomial under quadratic constraints can be relaxed into a quadratic SDP (Luo & Zhang, 2010). In contrast to the SoS approach, which gives a matrix variable of size at least $N \times N$, the quadratic SDP system has a size of $n \times n$ only. The resulting quadratic SDP can be well approximated in polynomial time in some cases, but it remains NP-hard. Yet, these methods primarily depend on the specific structure of the problem, and generally, the scalability of characterizing polynomial nonnegativity remains a significant challenge in the literature.

B Mathematical Background for SoS

Definition 2.1 implies that the degree of an SoS polynomial q has to be even and that the maximum degree of each \tilde{q}_j is $\frac{\deg[q]}{2}$. Therefore, we denote the degree of the SoS

polynomial as $p' = 2d$ where d is a positive integer.

Definition B.1. (Total degree polynomial space) Let $n > 0$ be the dimension and $\mathbf{x} = [x_1, \dots, x_n]^T$ be the variables. We denote $\mathcal{P}_{p'}[\mathbf{x}]$ as the general representation of the polynomial spaces, where p' represents the highest degree each entry can take. The associated multi-index set as $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_n] \in [\mathbb{Z}[0, p']^n$ where each α_r is an integer between 0 to p' inclusively and the indices satisfy $\sum_{r=1}^n \alpha_r \leq p'$. The number of monomial bases are $N := \binom{n+p'}{p'}$.

Theorem B.2. (From (Parrilo, 2000)) For a variable $\mathbf{x} = [x_1, \dots, x_n]^T$ and an even integer $p' = 2d$, let $\phi_d(\mathbf{x})$ be the vector of all monomials of degree at most d in x_j for $1 \leq j \leq n$. A polynomial $\hat{p} : \mathbb{R}^n \rightarrow \mathbb{R}$ of degree p' is SoS if and only if there exists a symmetric matrix \mathbf{Q} such that (i) $\hat{p}(\mathbf{x}) = \phi_d(\mathbf{x})^T \mathbf{Q} \phi_d(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^n$, (ii) $\mathbf{Q} \succeq 0$.

C SoS Simple

Step 1: Examine if the highest degree is odd or even.

Step 2: For the even highest degree d , examine the coefficients of highest-degree terms. Check for any negative values.

Step 3: Consider these special properties:

- Properties of quadratic polynomials
- Properties of quartic polynomials in 1-2 variables.
- Properties of quartic homogeneous polynomials in 1-3 variables.
- Properties of even-degree univariate polynomials.

Step 4: Try direct sum of squares representation.

Step 5: Consider matrix methods if needed.

D SoS Reasoning

Step 1. Degree: An SoS polynomial must have an even degree (i.e., its highest-degree term must have an even exponent). Any odd-degree polynomial cannot be expressed as a sum of squared polynomials. This is the simplest criterion and should always be checked first.

If the highest-degree univariate term (i.e., x_1^d, \dots, x_n^d) has a negative coefficient, then the polynomial is not SoS. Otherwise, we cannot determine whether it is SoS and proceed to the next step.

Example 1: $p(\mathbf{x}) = x_1^4 - x_2^4 + x_3^4 + x_1^2 x_2^2$. Since the highest-degree univariate term has a negative coefficient (namely, $-x_2^4$), by letting $x_2 \rightarrow \infty$, it is clear that $p(x)$ becomes negative. Therefore, it is not SoS.

Example 2: $p(\mathbf{x}) = x_1^4 + x_2^4 + x_3^4 - 2x_1^2 x_2^2 + x_1 x_2$. All the highest-degree univariate terms have non-negative coefficients (i.e., x_1^4, x_2^4, x_3^4). Thus, we cannot determine whether it is SoS, and we move to the next step.

Example 3: $p(\mathbf{x}) = x_1^4 + x_2^4 - 2x_1^2 x_2^2$. This polynomial is SoS because it can be rewritten as: $p(\mathbf{x}) = (x_1^2 - x_2^2)^2$. Note that a negative coefficient in the highest-degree cross term is allowed. For instance, in this case, we have the negative coefficient cross term $-2x_1^2 x_2^2$. However, the highest-degree univariate terms are positive (i.e., x_1^4, x_2^4).

Test Set Construction: Test Set 1 is constructed such that the highest-degree term is odd, thereby ensuring that the polynomials are not SoS.

Step 2. Non-negativity: SoS polynomials are nonnegative for all real inputs. For example, if a polynomial $p(\mathbf{x})$ has a negative constant term, then $p(0) < 0$, proving it is not SoS. Similarly, if a horizontally translated and scaled polynomial $q(\mathbf{x}) = cp(\mathbf{x} + \mathbf{d})$ (for any $c \in \mathbb{R}$ and $\mathbf{d} \in \mathbb{R}^n$) satisfies $q(0) < 0$, then $p(\mathbf{x})$ cannot be SoS.

To determine whether a polynomial is nonnegative, please use the following approaches:

Constant coefficient check: If the constant coefficient is negative, then $p(0) < 0$. For instance, $p(x) = x^4 + x^3 - 1$, $p(\mathbf{x}) = x_1^2 + x_1^2 x_2^2 + x_2^4 - 0.1$. are no SoS polynomials.

Grid evaluation: Try finding the minimum value of the polynomial over a selected evaluation grid. It is crucial to perform this step. Substitute multiple values of x , such as $(1, 0, 0, \dots)$, $(0, 1, 0, \dots)$, $(0, 0, 1, \dots)$, etc., to check whether the polynomial evaluates to a negative value.

Leading order and dominant terms: Analyze the highest-degree terms and explore symmetries among cross terms. Evaluate the magnitude of negative coefficients relative to

positive coefficients.

Finding minima: Attempt to find the local or global minimum of the polynomial to determine if it is negative.

Finding Symmetry and Translation: Example 1: Consider a horizontally translated and scaled polynomial: $p(\mathbf{x}) = 1.8x_1^2 + 10.8x_1 + 1.2x_2^2 + 4.8x_2 + 20.82$. Rewriting, $p(\mathbf{x}) = 1.8(x_1 + 3)^2 + 1.2(x_2 + 2)^2 - 0.18$. Since $p(-3, -2) < 0$, the polynomial is still not SoS.

Step 3. Square Form: An SoS polynomial $p(\mathbf{x})$ can be written as $p(\mathbf{x}) = \sum_i q_i(\mathbf{x})^2$, where each $q_i(\mathbf{x})$ is a polynomial. Examples and counterexamples are provided in Test Set 2, as this is the most common method for checking SoS.

Example: Consider $p(\mathbf{x}) = (x_1 - x_1x_2)^2 + (x_2^2 - x_1^4)^2$, which is an SoS polynomial. However, polynomials are sometimes given in their expanded form. For instance, the same polynomial can be written as: $p(\mathbf{x}) = -2x_1^2x_2 + x_1^2 + x_1^8 - 2x_1^4x_2^2 + x_1^2x_2^2 + x_2^4$. On the other hand, consider $p(\mathbf{x}) = (x_1 - x_1x_2)^2 + (x_2^2 - x_1^4)^2 - 20$. This polynomial is not SoS. To determine whether an expanded polynomial can be expressed in SoS form with a negative constant, one should analyze the symmetries of the terms and the structure of the cross terms.

Test Set Construction: Examples and counterexamples are provided in Test Set 2.

Step 4. Special Structures and Cases:

- a) Any nonnegative quadratic polynomial is a sum of squares (SoS).

Examples: $p(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1x_2$, $p(\mathbf{x}) = x_1^2 + x_2^2 + 4x_3^2 - 3x_2x_3$. These are SoS.

Counterexamples: $p(\mathbf{x}) = x_1^2 + x_2^2 - 2x_1x_2 - 1$, $p(\mathbf{x}) = x_1^2 + x_2^2 + 4x_3^2 - 5x_2x_3$.

- b) Any nonnegative quartic polynomial in one or two variables is SoS.

Example: $p(\mathbf{x}) = x_1^4 + 2x_1^2x_2 - 2x_1^2 + x_2^2 - 2x_2 + 1 = (x_1^2 + x_2 - 1)^2$.

Counterexample: $p(\mathbf{x}) = x_1^4 + 2x_1^2x_2 - 2x_1^2 + x_2^2 - 2x_2 = (x_1^2 + x_2 - 1)^2 - 1$.

- c) Any nonnegative quartic homogeneous polynomial in one, two, or three variables is SoS.

- d) Any nonnegative even-degree univariate polynomial is SoS.

Example: $p(x) = x^6 + 3x^4 + 2x^2$, which is nonnegative and SoS.

Counterexample: $p(x) = x^6 + 3x^4 + 2x$, which takes negative values and is not SoS.

- e) Any nonnegative polynomial with a quadratic term and quartic regularization is SoS.

Therefore, if a polynomial meets one of the above criteria and is nonnegative, it is an SoS polynomial. Nonnegativity can be verified by determining the global minimum, checking the descent direction, or performing a grid search.

Test Set Construction: Examples are provided in Test Set 3.1, 3.2, and 4, while counterexamples are constructed as polynomials that take negative values.

Step 5. Matrix Decomposition and Check for Symmetric Positive Definite Q: If the above checks fail, we can use the following theoretical reasoning:

- a) For an even degree $2d$ polynomial in $[x_1, \dots, x_n]$, construct a monomial basis using canonical ordering:

$$\mathbf{y}^* := (x_1^*, \dots, x_n^*, (x_1^*)^2, \dots, x_1^*x_n^*, x_2^*x_3^*, \dots, (x_1^*)^{2d}, \dots, (x_n^*)^{2d}).$$

This vector \mathbf{y}^* has length $\binom{n+2d}{2d}$.

- b) Express the polynomial as $p(\mathbf{x}) = \mathbf{y}^{*\top} \mathbf{Q} \mathbf{y}^*$, where \mathbf{Q} is a symmetric matrix of size $\binom{n+2d}{2d} \times \binom{n+2d}{2d}$. Note that this representation is not unique; there are multiple valid forms of \mathbf{Q} .
- c) Check whether \mathbf{Q} is positive definite. This can be done by finding its smallest eigenvalue. If such a \mathbf{Q} exists, then $p(x)$ is a sum of squares (SoS). Otherwise, $p(x)$ is very likely not SoS.

If all the above tests fail, we can try **Semidefinite Programming (SDP)**, which is the test used by existing solvers (e.g., YALMIP) to verify whether a polynomial is SoS. For example, to determine if

$$p(\mathbf{x}) = x_1^4 - 4x_1^3x_2 + 7x_1^2x_2^2 - 4x_1x_2^3 - 4x_1x_2 + x_2^4$$

is SoS. We convert the problem to the following. We solve the SDP

$$\gamma^* = \min p = y_{40} - 4y_{31} + 7y_{22} - 4y_{13} - 4y_{11} + y_{04}$$

subject to the constraint

$$\begin{bmatrix} 1 & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} \end{bmatrix} \succeq 0.$$

If $\gamma^* \geq 0$, then p is SoS; otherwise, it is not.

Test Set Construction: Examples and counterexamples are provided in Test Set 5.1-5.4. The examples are constructed using different positive semidefinite (PSD) matrices \mathbf{Q} , such as sparse PSD \mathbf{Q} , low-rank PSD \mathbf{Q} , and ill-conditioned PSD \mathbf{Q} . The counterexamples are generated using an indefinite \mathbf{Q} and the corresponding polynomials are likely to not be SoS. However, to conclusively prove that a polynomial is not SoS, all possible \mathbf{Q} must be examined. In our test set, we cross-checked the results with classic solvers to confirm the "Not SoS" classification for Test Sets 5.1-5.4.

E Details for Test Subsets

We provide the details of test subsets in Table 3.

F Sample Model’s Response under SoS Reasoning Prompt

F.1 o1-mini response for a simple polynomial under SoS Reasoning Prompt

To determine whether the given polynomial is a sum of squares (SoS), we will follow the provided steps:

Step 1: Check the Highest Degree

The highest degree of a polynomial is the largest total degree among its terms. When expanding these squared terms, the highest degree terms are of the form x_i^3 (e.g., x_1^3, x_2^3 , etc.), which are degree 3 terms.

Since 3 is an odd degree, according to Step 1: If the highest degree of a polynomial is odd, it is not SoS. Therefore, the

polynomial cannot be expressed as a sum of squares.

F.2 o1-mini response for a difficult polynomial under SoS Reasoning Prompt

To determine whether the given polynomial is a sum of squares (SoS), we follow the outlined steps meticulously.

Step 1: Check the Highest Degree

The highest total degree among the terms is 4, which is even. Since the highest degree is even, we proceed to the next step.

Step 2: Check Highest Degree Univariate Terms for Negative Coefficients

Univariate Terms of Degree 4: $7.9x_1^4$ (Coefficient: 7.9) $4.1x_2^4$ (Coefficient: 4.1) $0.91x_3^4$ (Coefficient: 0.91) $2.0x_4^4$ (Coefficient: 2.0) $0.92x_5^4$ (Coefficient: 0.92)

Conclusion: All highest degree univariate terms have positive coefficients. We cannot determine the SoS property yet and move to the next step.

Step 3: Check for Negative Values

Testing Specific Points: Point (1, 1, 1, 0, 0), Evaluated $p(1, 1, 1, 0, 0) = 66.89$ (Positive). Point (0, 0, 0, 1, 1), Evaluated $p(0, 0, 0, 1, 1) = 6.52$ (Positive). Point (1, 1, 1, 1, 0), Evaluated $p(1, 1, 1, 1, 0) = 2.037$ (Positive).

Conclusion: At tested points, the polynomial remains non-negative. While this doesn’t guarantee it’s SoS, it suggests the possibility.

Step 4: Special Cases Analysis: The polynomial doesn’t clearly fit into any of the special cases (a to f) outlined for immediate determination.

Step 5: Advanced Methods - Matrix Decomposition and SDP: Given the complexity of the polynomial, performing matrix decomposition or semidefinite programming (SDP) manually is impractical. However, based on the evaluations in Step 3 and the positive coefficients in Step 2, there’s a strong indication that the polynomial does not take negative values.

Final Conclusion: Based on the evaluations and the absence of negative terms in critical components, the poly-

SoS1: O1 and R1-Like Reasoning LLMs are Sum-of-Square Solvers

Polynomial Type	length<4000	length>4000	Total	Is it SoS?	Difficulty
Test Set 1: Odd Degree Polynomial	150	50	200	NO	Easy
Test Set 2a: SoS (Expanded Form)	69	51	120	YES	Hard
Test Set 2b: Negative (Expanded Form)	23	40	63	NO	Hard
Test Set 2.1a: SoS (Squared Form)	105	15	120	YES	Easy
Test Set 2.1b: Negative (Squared Form)	38	25	63	NO	Easy
Test Set 3.1a: Nonnegative Quadratic Quartic	100	0	100	YES	Medium
Test Set 3.1b: Negative Quadratic Quartic	100	0	100	NO	Medium
Test Set 3.2a: Nonnegative Quartic with 2 variables	100	0	100	YES	Medium
Test Set 3.2b: Negative Quartic	100	0	100	NO	Medium
Test Set 4a: Nonnegative Quadratic Quartic	100	0	100	YES	Medium
Test Set 4b: Negative Quartic	100	0	100	NO	Medium
Test Set 5.1a: PSD \mathbf{Q}	80	16	96	YES	Hard
Test Set 5.1b: Non-PD \mathbf{Q}	80	16	96	NO	Hard
Test Set 5.2a: PSD Spare \mathbf{Q} (Sparsity 0.1)	56	16	72	YES	Hard
Test Set 5.2b: Non-PD Spare \mathbf{Q} (Sparsity 0.1)	56	16	72	NO	Hard
Test Set 5.3a: PSD Low Rank \mathbf{Q} (rank 3)	42	18	60	YES	Hard
Test Set 5.3b: Non-PD Low Rank \mathbf{Q} (rank 3)	28	12	40	NO	Hard
Test Set 5.4a: PSD Ill-Conditioned \mathbf{Q} ($\lambda = 1 - 10^{12}$)	20	15	35	YES	Hard
Test Set 5.4b: Non-PD Ill-Conditioned \mathbf{Q}	40	30	70	NO	Hard

Table 3: Summary of SoS-1K Test Sets.

mial is a sum of squares.

G Further Results for Performance of SOTA LLMs

H Research Level SoS and Nonnegativity Questions

1. Can you comment on the sum of squares (SoS) and nonnegativity properties of the following polynomials?

- $p(\mathbf{x}) = x_1^4 x_2^2 + x_1^2 x_2^4 + 1 - 3x_1^2 x_2^2$,
- $p(\mathbf{x}) = x_1^6 + x_2^6 + x_3^6 - x_1^4 x_2^2 - x_1^4 x_3^2 - x_2^4 x_1^2 - x_2^4 x_3^2 - x_3^4 x_1^2 - x_3^4 x_2^2 + 3x_1^2 x_2^2 x_3^2$,
- $p(\mathbf{x}) = (x_1^4 + x_2^4 + x_3^4) + 2(x_1^2 + x_2^2 + x_3^2) + 8(x_1 x_2 + x_1 x_3 + x_2 x_3) + \frac{9}{4}$.

2. Can you construct a general formula for nonnegative polynomials that are not SoS?

3. Can you provide a new nonnegative polynomial that is not SoS and has never been found in the literature?

Note that the first three polynomials are well-known examples of nonnegative but not sum-of-squares (SoS) polynomials. The first is the Motzkin polynomial, a classic example of a nonnegative polynomial that is not SoS. The second is the Robinson polynomial, a counterexample to Hilbert’s 17th problem. The last polynomial is a recent result from Ahmadi’s work (2023) [Ahmadi2023].

Finding nonnegative polynomials that are not sum of squares (SoS) is a significant ongoing research problem in real algebraic geometry and polynomial optimization (Ahmadi

Model	# Total Samples	# Valid Samples		
Instruction Type		SoS Plain	SoS Simple	SoS Reasoning
DeepSeek-R1	340	300	302	234
QWQ-32b	340	233	259	225
o1-mini	340	338	340	337
Qwen2.5-7b	340	323	332	277
GPT-4o	340	336	338	337
Qwen2.5-7b-1m	340	340	339	286
Qwen2.5-14b	340	325	340	316
Qwen2.5-14b-1m	340	340	336	309
GPT-4o-mini	340	339	339	327
DeepSeek-V1	340	340	340	332
Qwen2.5-32b	340	334	339	315
Average	340	323	328	300

Table 4: # Total Samples and Valid Samples of Different Models.

et al., 2023b; 2024; Ahmadi & Zhang, 2022). It connects to Hilbert’s 17th problem, semidefinite programming (SDP), and positivity certificates in polynomial optimization.