S1: Simple test-time scaling

대전 4반 DL 스터디

유승현

목차

- 논문 정보
  - 저자
  - 요약
- 배경 지식
  - Test-Time Scaling vs Train-Time Scaling
- 핵심 내용
  - s1K
  - Budget Forcing
  - 추가 접근

저자

Niklas Muenninghoff

1st year PhD at Stanford

Bloom: 176B 라는 프로젝트에 참가했음

요약

고품질의 데이터와 효과적인 추론 제어 = 성능향상

배경 지식

- Test-Time Scaling vs Train-Time Scaling

Test-Time Scaling vs Train-Time Scaling

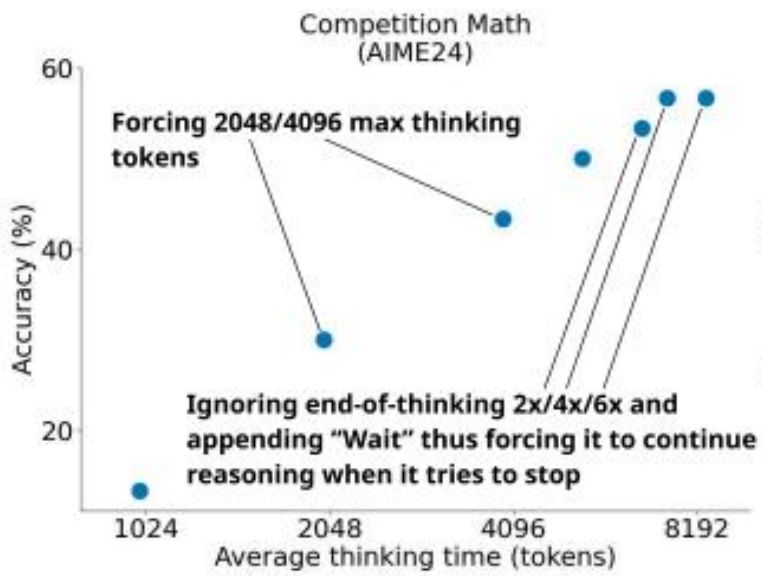
둘 다 Model의 성능을 높이는 기법

어떻게 성능을 올리는지 그 방법의 차이

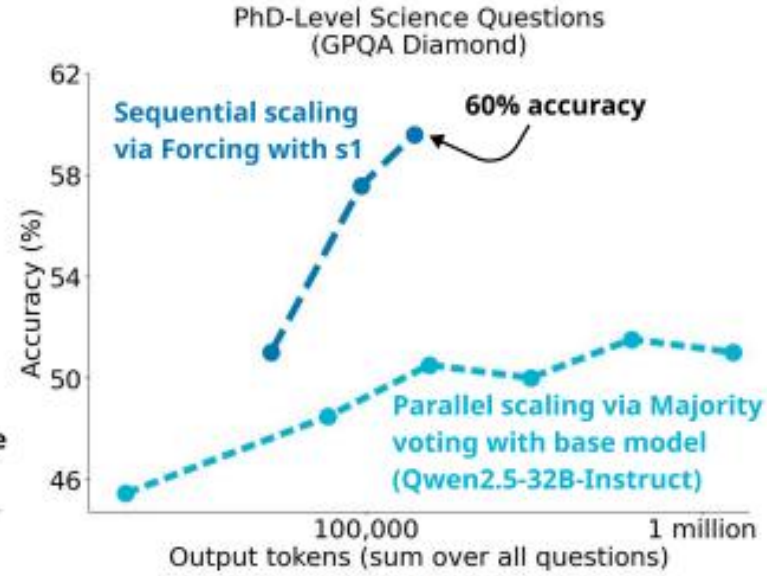
Train-time scaling : 기존의 방법. 더 많은 데이터로 더 많은 학습을 통한 성능 향상

Test-time scaling : o1을 통해 주목받는 방식. 추론 과정을 통한 성능 향상

Test-Time Scaling



(a) Sequential scaling via budget forcing



(b) Parallel scaling via majority voting

두 가지 대표적인 방법

1. 병렬 방식 (Parallel Scaling)

>> 여러 독립적인 출력을 생성한 후 가장 적절한 답을 선택하는 방법

>> 다수결 방식, 후처리 알고리즘 등으로 최종 답을 결정

>- 계산량이 증가하고, 반드시 최적의 답을 도출하는 것은 아님 => 한계가 존재

2. 순차 방식 (Sequential Scaling)

>> 하나의 답을 생성하는 과정에 이전 사고 과정을 활용하여 더 나은 답을 찾는 방식

핵심 내용

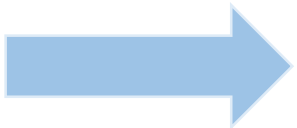
1. 고품질 소규모 데이터셋 (s1K)

2. Budget Forcing

3. 추가 접근

s1K

Source	Description	#Samples	Avg. thinking length
NuminaMATH (LI et al., 2024)	Math problems from online websites	30660	4.1K
MATH (Hendrycks et al., 2021)	Math problems from competitions	11999	2.9K
OlympicArena (Huang et al., 2024a)	Astronomy, Biology, Chemistry, Computer Science, Geography, Math, and Physics olympiad questions	4250	3.2K
OmniMath (Gao et al., 2024a)	Math problems from competitions	4238	4.4K
AGIEval (Zhong et al., 2023; Ling et al., 2017; Hendrycks et al., 2021; Liu et al., 2020; Zhong et al., 2019; Wang et al., 2021)	English, Law, Logic and Math problems from the SAT, LSAT and other exams	2385	1.2K
xword	Crossword puzzles	999	0.7K
OlympiadBench (He et al., 2024)	Math and Physics olympiad questions	896	3.9K
AIME (1983-2021)	American Invitational Mathematics Examination	890	4.7K
TheoremQA (Chen et al., 2023)	Computer Science, Finance, Math, and Physics university-level questions relating to theorems	747	2.1K
USACO (Shi et al., 2024)	Code problems from the USA Computing Olympiad	519	3.6K
JEEBench (Arora et al., 2023)	Chemistry, Math, and Physics problems used in the university entrance examination of the Indian Institute of Technology	515	2.9K
GPQA (Rein et al., 2023)	PhD-Level Science Questions	348	2.9K
SciEval (Sun et al., 2024)	Biology, Chemistry, and Physics problems from various sources	227	0.7K
s1-prob	Stanford statistics qualifying exams	182	4.0K
LiveCodeBench (Jain et al., 2024)	Code problems from coding websites (LeetCode, AtCoder, and CodeForces)	151	3.5K
s1-teasers	Math brain-teasers crawled from the Internet	23	4.1K
All 59K questions	Composite of the above datasets with reasoning traces and solutions	59029	3.6K



Domain	#questions	Total token count	Keywords
Geometry	109	560.2K	Area, Triangle, Distance
Number theory	98	522.5K	Sequences, Divisibility
Combinatorics	75	384.7K	Permutations, Counting
Real functions	43	234.8K	Trigonometry, Calculus
Biology	41	120.9K	Organic reactions
Complex functions	32	170.2K	Complex roots
Quantum theory	32	127.9K	Particles, Wave functions
Field theory	28	150.1K	Polynomials, Roots
Calculus of variations	28	155.5K	Optimization, Control
Difference equations	24	132.5K	Recurrence, Recursion
Electromagnetic theory	23	95.8K	Optics, Waves, Diffraction
Group theory	22	100.0K	Groups, Automorphisms
Linear algebra	22	128.3K	Matrices, Determinants
Probability theory	20	114.6K	Random walk, Expectation
Algebraic systems	19	109.9K	Functional equations
Mechanics	19	103.6K	Forces, Motion, Energy
Thermodynamics	19	74.2K	Heat engines, Entropy
Differential equations	18	89.6K	Substitution, Existence
Computer science	18	34.2K	Complexity theory, Algorithms
Numerical analysis	18	76.5K	Error analysis, Stability
Calculus	17	96.3K	Convergence, Summation
Algebraic structures	17	90.4K	Inequalities, Sets
Astronomy	16	37.7K	Stellar populations, Orbits
Remaining 27 domains	242	982.2K	Domains with ≤ 16 questions
All domains (51)	1000	4.7M	**s1K**

s1K

품질(Quality) 기준을 적용하여, 잘못된 형식의 데이터나 의미가 명확하지 않은 문제들을 제거

난이도(Difficulty) 기준을 적용하여, 문제 해결을 위해 깊은 추론이 필요한 문제들을 선별

다양성(Diversity) 기준을 적용하여, 특정 유형의 문제에 치우치지 않도록 다양한 주제의 문제들을 균형 있게 포함하도록 구성

s1K

세 가지 기준의 중요성

1. Only Quality: 1K-random

Gemini 모델을 활용해 높은 품질의 추론 과정을 생성, 무작위로 1000개의 샘플을 선택함 = 난이도와 다양성 필터링 적용 X

2. Only Diversity: 1K-diverse

도메인 별로 균등하게 샘플링 = 난이도 필터링 적용 X

3. Only Difficulty: 1K-longest

가장 긴 추론과정 1000개 선택

4. Maximize Quantity: 59K-full

모든 샘플을 학습. But 학습에 394시간 필요. s1K는 7시간 학습

Model	AIME 2024	MATH 500	GPQA Diamond
1K-random	36.7 [-26.7%, -3.3%]	90.6 [-4.8%, 0.0%]	52.0 [-12.6%, 2.5%]
1K-diverse	26.7 [-40.0%, -10.0%]	91.2 [-4.0%, 0.2%]	54.6 [-10.1%, 5.1%]
1K-longest	33.3 [-36.7%, 0.0%]	90.4 [-5.0%, -0.2%]	59.6 [-5.1%, 10.1%]
59K-full	53.3 [-13.3%, 20.0%]	92.8 [-2.6%, 2.2%]	58.1 [-6.6%, 8.6%]
s1K	50.0	93.0	57.6

Budget Forcing

Test-time scaling을 최적화 하는 기법

사용자가 지정한 token 추론하도록 하는 기법

추론 과정이 너무 빨리 끝나면 -> 해당 내용을 한 번 더 검토하도록 수행
“Wait”

추론 과정이 너무 오래 걸리면 -> 중간에 중단
“Final Answer:”

기법 평가 지표

Control (제어력)

모델이 정해진 테스트 시간 내에 (예: 특정 범위의 추론 토큰 수) 얼마나 정확하게 동작하는지를 나타냅니다.
100%에 가까울수록 모델이 설정한 계산 예산 내에서 정밀하게 제어된다는 의미입니다.

Scaling (스케일링 능력)

테스트 시점에서 추가 연산(토큰 수 증가)에 따라 성능이 얼마나 개선되는지를 나타내는 지표입니다.
이는 추론 과정에서 더 많은 계산을 투입할 때 성능 향상의 기울기를 측정하며, 기울기가 클수록 추가 계산이 성능 향상에 효과적임을 의미합니다.

Performance (최종 성능)

주어진 테스트 시간 내에서 모델이 달성할 수 있는 최대 정확도를 의미합니다.
즉, 테스트 시점에서 할당된 연산 자원 하에서 모델이 내놓는 가장 좋은 성능 수준을 평가합니다.

기법 비교 테스트

- BF (Budget Forcing)

- TCC (Token-Conditional Control)
  특정 토큰 수에 도달할 때까지 추론 강제

- SCC (Step-Conditional Control)
  사고 단계를 기준으로 연산량 조절

- CCC (Class-Conditional Control)
  유형에 따라 사고 시간을 동적으로 조절

Method	Control	Scaling	Performance	$\|\mathcal{A}\|$
BF	**100%**	15	**56.7**	5
TCC	40%	-24	40.0	5
TCC + BF	**100%**	13	40.0	5
SCC	60%	3	36.7	5
SCC + BF	**100%**	6	36.7	5
CCC	50%	**25**	36.7	2
RS	**100%**	-35	40.0	5

- RS (Rejection Sampling)
  특정 기준을 충족하는 출력이 나올 때까지 반복적으로 샘플링하는 방식
  But 역스케일링 문제(스케일링을 많이 할수록 성능이 오히려 낮아지는 현상)가 발생할 수 있음

다른 기법들의 문제점

TCC – 모델이 토큰을 정확하게 세지 못함.

SCC – 총 토큰 수가 비슷하게 생성됨.

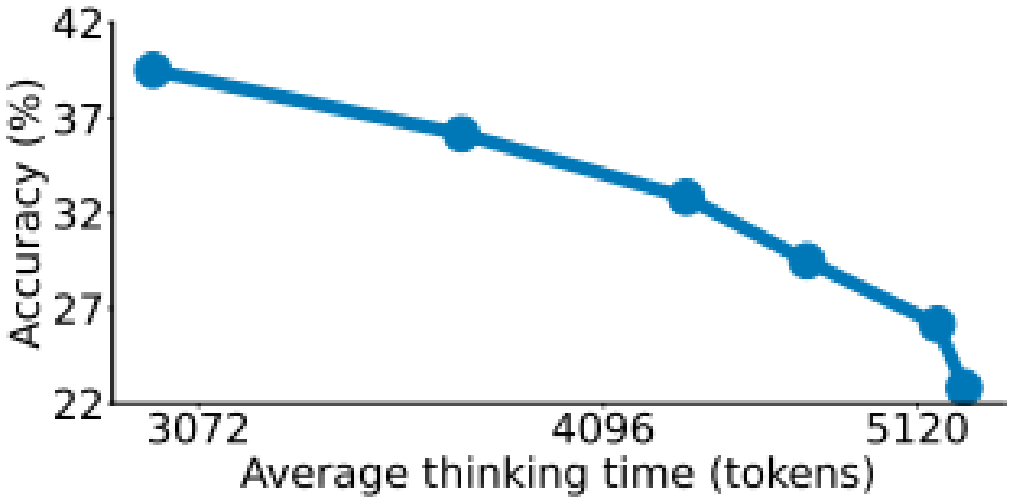
CCC – ‘think longer’라는 명령만으로 연산량이 증가하고 성능이 향상됨. 비교적 효과적

RS : 특정 길이에 맞는 생성물이 나올 때 까지 반복적으로 샘플링 하는 방법
예를 들어, 출력 길이가 4000 토큰 이하일 때까지 샘플을 계속 생성하여 기준을 충족하는 결과를 선택
역스케일링 현상 발견

역스케일링

O 4000 토큰 이하에서는 모델이 정답에 빠르게 도달.
O 8000 토큰 이하에서는 모델이 백트래킹을 많이 수행하며 정답이 틀리는 경우 증가

-> 긴 샘플일수록 모델이 실수를 하고, 이를 수정하기 위해 지나치게 되돌아가거나 자문하는 과정이 많아짐



[가설]
짧은 출력일수록 모델이 처음부터 정답 방향으로 가는 경우가 많음
긴 출력일수록 모델이 실수를 하고, 이를 고치려고 백트래킹하면서 잘못된 방향으로 가는 경향이 있음
따라서, 길이가 길어질수록 오히려 성능이 떨어지는 "역스케일링 " 현상이 발생.

Budget Forcing Extrapolation

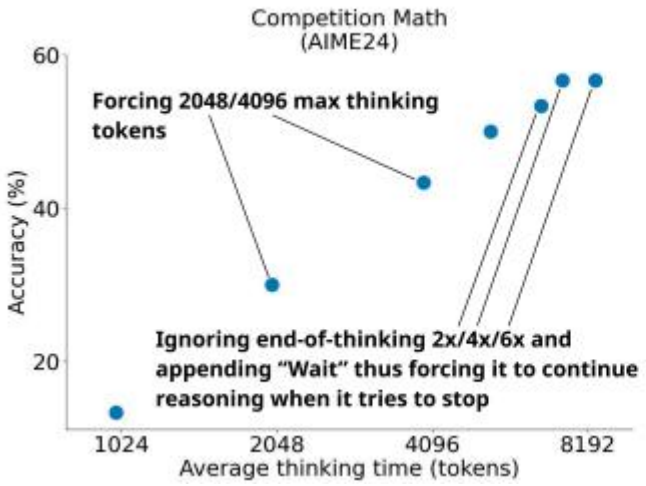
BG의 문자열로 여러가지를 테스트

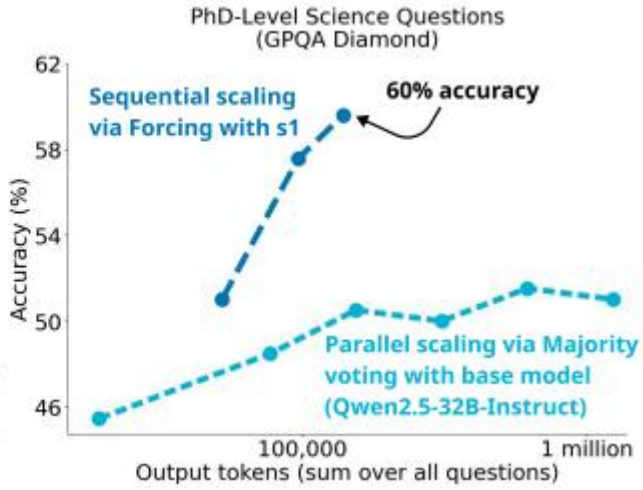
-> Wait가 가장 결과가 좋았음

Model	AIME 2024	MATH 500	GPQA Diamond
No extrapolation	50.0	**93.0**	57.6
2x without string	50.0	90.2	55.1
2x "Alternatively"	50.0	92.2	**59.6**
2x "Hmm"	50.0	**93.0**	**59.6**
2x "Wait"	**53.3**	**93.0**	**59.6**

+ BF의 한계
AIME 성능을 57%까지 향상할 수 있었으나 한계가 존재함.

1. 성능이 결국 한계에 도달해 정체가 발생
2. 모델의 Context Window 크기에 의해 제약이 발생 -> 한번에 처리할 수 있는 정보량이 한정되어 있어 일정 이상 확장이 어려움

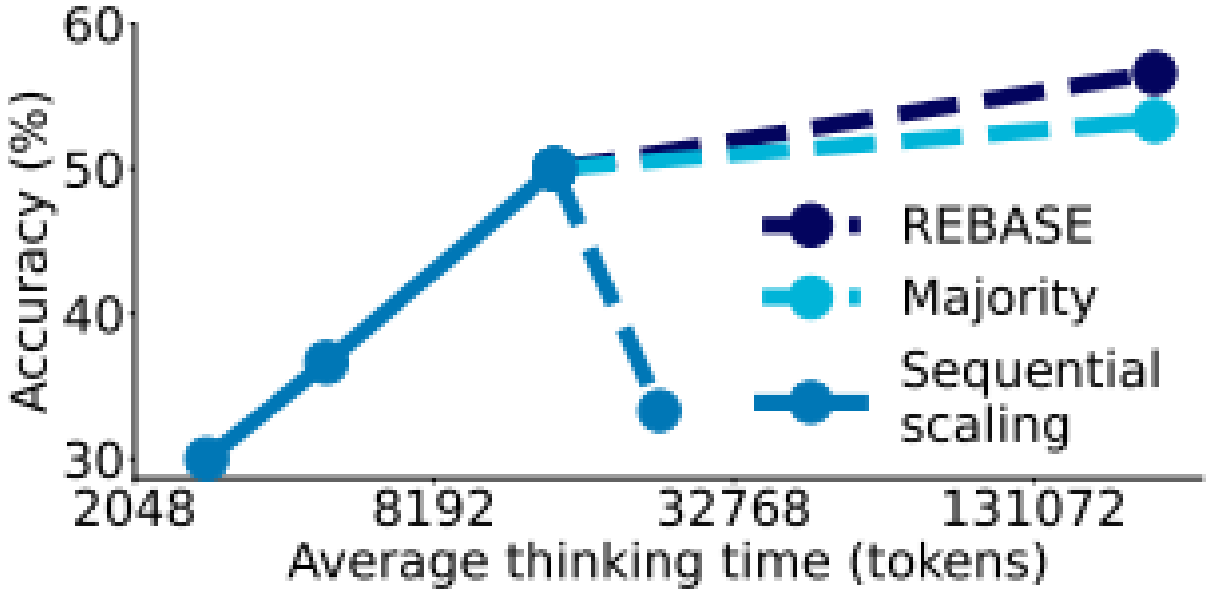




Budget Forcing 개선 방안

- “Wait” 뿐만 아니라, 다른 문자열을 순환하면서 사용하는 방법 연구 가능
- 반복적인 루프를 방지하기 위한 빈도 패널티 또는 높은 temperature 설정 결합 가능
- 강화 학습과 결합한 BF에 대한 연구가 필요

추가 접근



순차적 확장의 토큰을 늘이면
오히려 정확도가 급격히 감소함.
일정 수준 이상의 순차적 확장은
오버피팅 또는 백트래킹 문제를 초래할 수 있음

“병렬 확장”이 해결책이 될 수 있음

Q&A